

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>HTML5 und CSS3</title>
    <meta name="author" content="Thomas Maier">
    <meta name="date" content="2022-03-15">
    <meta name="website" content="www.css4.at">
  </head>
  <body>
    <h1>Inhalt</h1>
    <ul>
      <li>Didaktische Konzeption</li>
      <li>Zielformulierungen</li>
      <li>Lernhandouts</li>
      <li>Übungsblätter</li>
      <li>Pädagogischer Beipackzettel</li>
    </ul>
  </body>
</html>
```

# HTML & CSS

2024



## **HTML & CSS** **von Thomas Maier**

Zeppelinstraße 12A/7, 8055 Graz  
www.css4.at | html@css4.at

Copyright: CC Lizenz BY NC SA 2020, Version 2.1

Diese Arbeit wurde mit LibreOffice auf einem Linux Betriebssystem erstellt und entspricht damit einem Grundsatz der Creative Commons. Alle Inhalte sind als Open Educational Resources gekennzeichnet. Bitte prüfen Sie auf <https://oer.css4.at> ob eine neuere Version dieses Scriptes vorhanden ist!

# Inhaltsverzeichnis

<b>Allgemeine didaktische Konzeption.....</b>	<b>6</b>
Unterrichtsgrundsätze.....	6
Allgemeine didaktische Grundsätze.....	6
Unterrichtsprinzipien.....	7
Eingangsvoraussetzungen.....	7
Flankierende Methoden.....	8
Raumausstattung.....	9
EDV Infrastruktur.....	10
Blendet Learning.....	12
Sozialformen.....	13
Exemplarisches Stundenbild.....	13
Intensionen der Schüler_innen.....	14
Kompetenzbegriff und Zielsetzungen.....	14
Grobziele und Kompetenzzuwachs.....	15
<b>Handouts und Assignments.....</b>	<b>16</b>
Lernhandout 1.1 HTML Grundgerüst.....	17
Lernhandout 2.1 Basis Tags.....	19
Übungsblatt 2.1 Basis Tags.....	21
Lernhandout 2.2 Basis Elemente (img, a href).....	23
Übungsblatt 2.2 Basis a href, img.....	25
Lernhandout 2.3 Best Practice.....	27
Lernhandout 3.1 Tabellen.....	29
Übungsblatt 3.1 Tabellen.....	31
Lernhandout 3.2 HTML validieren.....	33
Lernhandout 4.1 style, span, color.....	35
Lernhandout 4.2 Rahmen.....	37
Lernhandout 4.3 Höhe und Breite.....	39
Übungsblatt 4.3 Farben, Rahmen und Größen.....	41
Lernhandout 5.1 CSS im Head.....	43
Lernhandout 5.2 Schriftarten.....	45
Lernhandout 5.3 Schriftgrößen.....	47
Lernhandout 5.4 Text Dekoration.....	49
Übungsblatt 5.4 Schriften und Textdekoration.....	51
Lernhandout 6.1 Ausrichtung.....	53
Lernhandout 6.2 Abstände.....	55
Übungsblatt 6.2 Ausrichtung und Abstände.....	57
Lernhandout 6.3 Best Practice.....	59
Lernhandout 7.1 Hintergrund.....	61
Lernhandout 7.2 Hintergrund.....	63

Lernhandout 7.3 Farbverlauf.....	65
Übungsblatt 7.3 Hintergrund.....	67
Lernhandout 8.1 Positionierung.....	69
Lernhandout 8.2 display-Eigenschaft.....	71
Übungsblatt 8.2 Positionierung und Display.....	73
Lernhandout 8.3 Strukturierung.....	75
Lernhandout 8.4 Float.....	77
Lernhandout 9.1 Pseudoklassen (Buttons).....	79
Lernhandout 9.2 Pseudoklassen.....	81
Übungsblatt 9.2 Pseudoklassen.....	83
Lernhandout 9.3 iFrames.....	85
Übungsblatt 9.3 iFrames.....	87
Lernhandout 10.2 Pseudoelemente.....	89
Übungsblatt 10.2 Pseudoelemente.....	91
Lernhandout 10.3 Audio.....	93
Übungsblatt 10.3 Audio.....	95
Lernhandout 10.4 Video.....	97
Übungsblatt 10.4 Video.....	99
Lernhandout 11.1 Medienabfrage.....	101
Übungsblatt 11.1 Medienabfragen.....	103
Lernhandout 11.2 Objekte.....	105
Übungsblatt 11.2 Objekte.....	107
Lernhandout 11.3 Viewport.....	109
Übungsblatt 11.3 Viewport.....	111
Lernhandout 11.4 Box Eigenschaften.....	113
Lernhandout 12.1 Formulare.....	115
Lernhandout 12.2 Textarea.....	117
Lernhandout 12.3 Input.....	119
Übungsblatt 12.3 Input.....	121
Lernhandout 12.4 Input Typen.....	123
Übungsblatt 12.4 Input Typen.....	125
Lernhandout 12.5 Form Button.....	127
Übungsblatt 12.5 Formular Buttons.....	129
Lernhandout 12.6 Dialogeingaben.....	131
Übungsblatt 12.6 Dialoge.....	133
Lernhandout 12.7 Auswahleingaben.....	135
Übungsblatt 12.7 Auswahleingaben.....	137
Lernhandout 12.8 Auswahllisten.....	139
Übungsblatt 12.8 Auswahllisten.....	141
Lernhandout 13.1 Animationen.....	143
Lernhandout 13.2 Animationen.....	145
Lernhandout 13.3 Animationen.....	147
Übungsblatt 13.3 Animationen.....	149
Lernhandout 13.4 Transformieren.....	151
Übungsblatt 13.4 Transformieren.....	153
Lernhandout 13.5 Transformieren.....	155
Lernhandout 13.6 Transitions.....	157
Übungsblatt 13.6 Transitions.....	159
Lernhandout 14.1 CSS Variablen.....	161
Lernhandout 14.2 calc() Funktion.....	163
Lernhandout 14.3 counter() Funktion.....	165

Übungsblatt 14.3 CSS Funktionen.....	167
Lernhandout 14.4 Attributselektoren.....	169
Lernhandout 15.1 Image Map.....	171
Übungsblatt 15.1 Image Map.....	173

**Pädagogischer Beipackzettel.....175**

Warum überhaupt noch HTML?.....	175
Gedanken zur Leistungsbewertung.....	175
Warum Blendet Learning?.....	176
Sozialformen in der IKT.....	176
Der konsequente Lehrplanbezug.....	177
Die kritische Sozialformierung.....	177
Jenseits von Gut und Sehr Gut.....	177
Mit Imperativen [programmieren] lernen.....	181

**Quellenverzeichnis.....184**

Internetquellen und online Tools.....	184
Literaturverzeichnis.....	184

# Allgemeine didaktische Konzeption

## Unterrichtsgrundsätze

---

Im Unterricht soll ein Klima der Wertschätzung sowie des Respekts und der Gleichbehandlung vorherrschen.

Das Klassenzimmer soll ein Ort der offenen Kommunikation sein. Der Unterricht ein Forum für Fragen und Diskussionen. Fragen aller Art sind erwünscht. Getreu: "Es gibt keine dummen Fragen – die einzige dumme Frage ist jene, die nicht gestellt wird".

Im Unterricht sollte Zeit für individuelles Fördern gefunden werden. Leistungsschwächere Schüler\_innen sollen gefördert werden. Leistungsstärkere Schüler\_innen gefordert. Dabei sollte dem "Schüler-Schüler-Coaching" Vorrang gegeben werden.

Der Fokus liegt auf den Leistungen der Schüler\_innen. Der besonderen Würdigung von Qualitäten soll einer fehlerzentrierten Wahrnehmung Vorzug gegeben werden. Fehler machen ist erlaubt – jedoch sollen diese kommuniziert werden.

Die Leistungen der Schüler\_innen sind sowohl auf der Prozessebene als auch auf der Produktebene zu würdigen. Es wird nicht nur das aktive Lernen gefordert, sondern auch das passive Lernen akzeptiert.

## Allgemeine didaktische Grundsätze

---

- Unterrichtsbeispiele mit Praxis- und Lebensbezug.
- Methodenvielfalt und kooperatives offenes Lernen
- Förderung von Selbstständigkeit, Selbstverantwortung, Einzel- und Teamarbeit, sozialem Lernen.
- Einsatz von Blended Learning (IT-Bezug), Standardsprache und Fachterminologie
- Individuelles Fördern bei einer Maximierung der reinen Lernzeit.
- Handlungsorientierte Unterrichtsplanung und Einhaltung der Progressionslogik
- Produkt und Prozessorientierte Leistungsbewertung

## Unterrichtsprinzipien

---

Es gelten die allgemeinen Unterrichtsprinzipien laut dem Bundesministerium für Bildung. Den fettgedruckten Prinzipien wurde in den Detailplanungen Rechnung getragen. Andere sollten unbedingt im Rahmen des pädagogischen/situationsangemessenen Gesprächs vermittelt werden. Vergessen Sie niemals diese äußerst wichtigen Prinzipien – sie sind weit mehr als einfaches fächerübergreifendes Lernen/Lehren.

- Erziehung zur Gleichstellung von Frauen und Männern
- Gesundheitserziehung
- **Interkulturelles Lernen**
- **Leseerziehung**
- **Medienbildung**
- **Politische Bildung**
- Sexualerziehung
- **Umweltbildung**
- Verkehrserziehung
- Ethisches Handeln
- **Wirtschaftserziehung und Verbraucher\_innenbildung**

## Eingangsvoraussetzungen

---

Das Erlernen von HTML und CSS kann nicht sofort ohne weiteres starten. Es sind Kompetenzen wie das Englische auf Niveau A2 sowie das Schreiben mit 10-Fingern klar von Vorteil. Viel wichtiger ist, grundsätzliche Erkenntnisse der Typographie, Design und Layout, Gestaltungslehre und einem (wohl am wichtigsten) Ordnungsbewusstsein als Eingangskompetenzen anzuführen. Da nun also, HTML und CSS als eine Vertiefung der konzeptionellen Erkenntnisse der Textverarbeitung (in primärer Hinsicht) angesehen werden kann, ist es erst nach der Kenntnisvermittlung einer allgemeinen Textverarbeitung als diskussionswürdig darzustellen.

Es stellen sich die kognitiven Probleme im einen Feld genauso dar, wie sie es im wechselseitig anderem tun. So will die Darstellung von Text sowohl im naheliegenden Einfachen die Entwicklungsmöglichkeiten für die\_ den Lernende\_n dieselben Erfahrungsmomente nach sich ziehen. Exemplarisch: Die Unterscheidung zwischen einer Zeilenschaltung und einer Absatzschaltung in MS Word, liegt im Drücken der Shift-Taste. Die Kennzeichnung einer Zeilenschaltung in HTML bedarf eines `<br>` Tags – und noch extremer: die Absatzschaltung wird durch einen Start-Tag `<p>` und End-Tag `</p>` realisiert.

So bleiben also, jene Textverarbeitungslehrer\_innen weiterhin die einführende Instanz und sollten aufgrund ihrer Kompetenzen niemals als vernachlässigt betrachtet werden. Sie sind es, die einen Text die Lesbarkeit verleihen – neben der äußerst wichtigen Arbeit von Germanisten (im Deutsch-Unterricht) und im besonderen (Kunstlehrer\_innen) die den

Spaß an der textuellen Veränderung anstreben um damit den Nimbus der kreativen textuellen Gestaltung zu nähren.

Das Vorkenntnisse in der allgemeinen Informatik als Grundvoraussetzung betrachtet werden soll, liegt doch klar auf der Hand. Ebenso sollte der\_die Schüler\_in ein Office Paket beherrschen. Einblicke in die Datenbankmodellierung (z. B. über Access) sind von Vorteil. Der\_die Schüler\_in muss das eigene Betriebssystem kennen.

## Flankierende Methoden

---

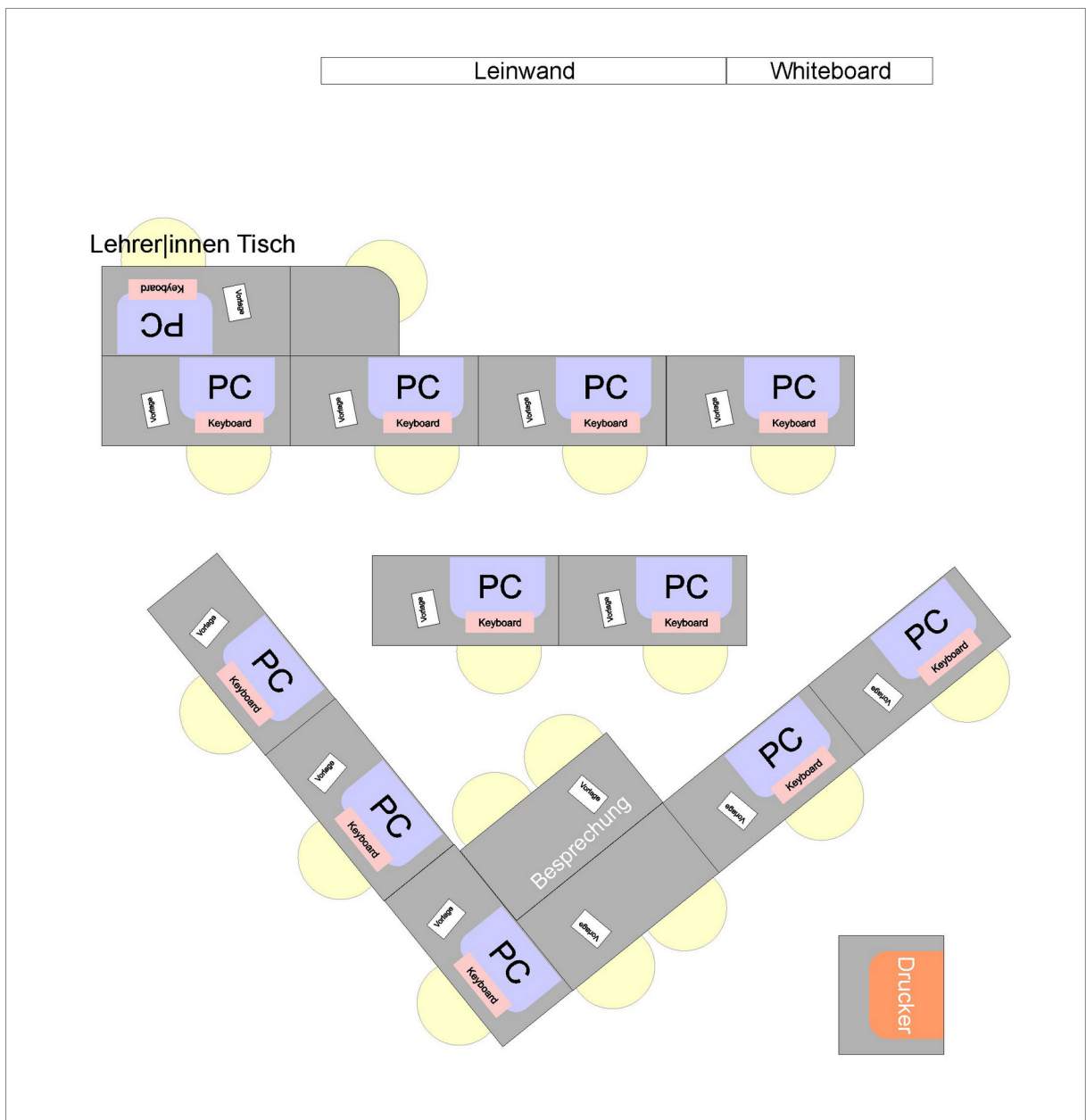
Neben den klassischen Methoden Frontalunterricht, Erklären, Lehrervortrag und Vorzeigen-Nachmachen (bzw. Vorzeigen-Mitmachen), sind folgende Unterrichtsmethoden ebenfalls denkbar:

- Aufgabenorientiertes Lernen
- Fächerübergreifendes und –verbindendes Lernen
- Handlungsorientiertes Lernen
- Einzelarbeit
- Konstruktionsaufgaben
- Projektlernen
- projektorientierter Unterricht
- Freie Arbeit
- Entdeckendes Lernen
- Mind-Mapping
- Brainwriting
- uvm.

# Raumausstattung

Folgende Medien und Ausstattungen sind für den Unterricht empfehlenswert:

- Großes Whiteboard
- Beamer (Projektor)
- Besprechungstische
- ausreichend Bürostühle
- jeweils einen Bürotisch pro PC (Arbeitsplatz – PC sind rechts angeordnet, damit die Linke Hälfte des Tisches Platz für Vorlagen bietet).
- ein Drucker



Vorschlag für eine Raumgestaltung

## EDV Infrastruktur

---




HTML & CSS ist äußerst genügsam, was die Leistung der Hard- und Software betrifft. Im Grunde reicht ein gängiger PC mit entsprechender Standardperipherie (z. B. Tastatur, Maus, Bildschirm usw.). Das Betriebssystem spielt nur eine untergeordnete Rolle, da HTML und CSS plattformunabhängig ist. Ob nun Windows, OSX oder Linux – egal! Mit Linux ist die Arbeit m. E. am schönsten.

Dennoch sind folgende Software-Spezifikationen empfehlenswert:

- Notepad ++, Kate (Texteditor)
- Microsoft Visual Code, Atom (IDE)
- Adobe Photoshop oder Gimp (Bildbearbeitung) {+}
- sämtliche gängige Browser (Chrome, Firefox, Edge usw.) {+}
- Audacity (Audioeditor)
- Microsoft Word oder OpenOffice (Textverarbeitung)
- CMS Editoren (z. B. Adobe Muse, Joomla)
- weitere Tools: Color Picker, ZIP Software, Screenshot-Tools, PDF Viewer
- kollaborative Anwendungen: Office 365, OneNote, Google, TeamViewer
- didaktische Tools: Veyon, Moodle
- FTP Software (z. B. FileZilla)
- Virtualisierungssoftware

Eine Webanbindung des Schulservers (z. B. über IIS, Apache odgl.) wäre sicher ein Pluspunkt, damit die Schüler\_innen ihre Webseiten sofort online stellen können. Die Planung eines Webservers ist aber ein anderes Thema, das den Rahmen dieser Arbeit sprengen würde. Die Schüler\_innen können aber ihre Lernprodukte auf gratis Webspace Anbietern, wie z. B. mur.at oder Lima-City.de publizieren (wenn nötig und wenn sie es wollen).

## Beispiele für eine Basisinstallation

		
Microsoft Windows	Apple OSX	Linux
<ul style="list-style-type: none"> <li>• XAMPP (WAMPP)</li> <li>• Notepad ++</li> <li>• Adobe Dreamweaver</li> <li>• ATOM</li> <li>• Edge, Chrome, Firefox</li> <li>• MS Office</li> <li>• MS Access</li> <li>• Color Picker</li> <li>• 7Zip</li> <li>• Snipping Tool</li> <li>• Adobe PDF Acrobat</li> <li>• FileZilla</li> <li>• OneNote</li> <li>• TeamViewer</li> <li>• Veyon (iTalc)</li> <li>• Gimp bzw. Adobe Photoshop</li> <li>• Corel Draw, Adobe Illustrator, Inkscape</li> <li>• Audacity</li> <li>• Oracle VirtualBox</li> <li>• MS SQL Server</li> </ul>	<ul style="list-style-type: none"> <li>• XAMPP</li> <li>• TextWrangler</li> <li>• Adobe Dreamweaver</li> <li>• ATOM</li> <li>• Safari, Chrome, Firefox</li> <li>• Apple iWork</li> <li>• MS Access</li> <li>• Color Picker</li> <li>• 7Zip</li> <li>• Skitch</li> <li>• Adobe PDF Acrobat</li> <li>• FileZilla</li> <li>• Evernote</li> <li>• TeamViewer</li> <li>• Veyon (iTalc)</li> <li>• Gimp bzw. Adobe Photoshop</li> <li>• Corel Draw, Adobe Illustrator, Inkscape</li> <li>• Audacity</li> <li>• Oracle VirtualBox</li> <li>• SQL Server</li> </ul>	<ul style="list-style-type: none"> <li>• XAMPP</li> <li>• Sublime</li> <li>• Bluefish</li> <li>• ATOM</li> <li>• Chrome, Firefox</li> <li>• LibreOffice</li> <li>• Kexi</li> <li>• KColorChooser</li> <li>• 7Zip</li> <li>• Shutter</li> <li>• Adobe Reader</li> <li>• FileZilla</li> <li>• Evernote</li> <li>• TeamViewer</li> <li>• Veyon (iTalc)</li> <li>• Gimp</li> <li>• Inkscape</li> <li>• Audacity</li> <li>• Oracle VirtualBox</li> <li>• SQL Server</li> </ul>

Die Hardware ist definitiv genügsamer:

- Bei Laptops sollte ein externes Keyboard bereitgestellt werden. Ebenso eine Maus.
- Ein zweiter Monitor ist ein Nice-to-Have, aber nicht zwingend notwendig.
- Eine Breitband-Internet-Verbindung.
- Ein funktionierendes Netzwerk mit Serverseitigen Profilen und Netzlaufwerken.
- USB Anschlüsse für die Verwendung von USB-Speicher-Sticks.
- Ein Netzwerk-Drucker
- Mobile Devices (Smartphone und Tablets) um die Scripte zu testen

# Blendet Learning



Abbildung: Die Bannernavigation von www.css4.at

Auf der Lernplattform [www.css4.at](http://www.css4.at) befindet sich links oben ein Suchfeld. Über dieses kann man nach notwendigen Dateien, Suchbegriffen und Refcodes suchen.

Auf der Startseite befindet sich der Lernbereich als dunkelblauer Kasten. Dort findet man unter Web Development einen Link zur Lernplattform: HTML und CSS.

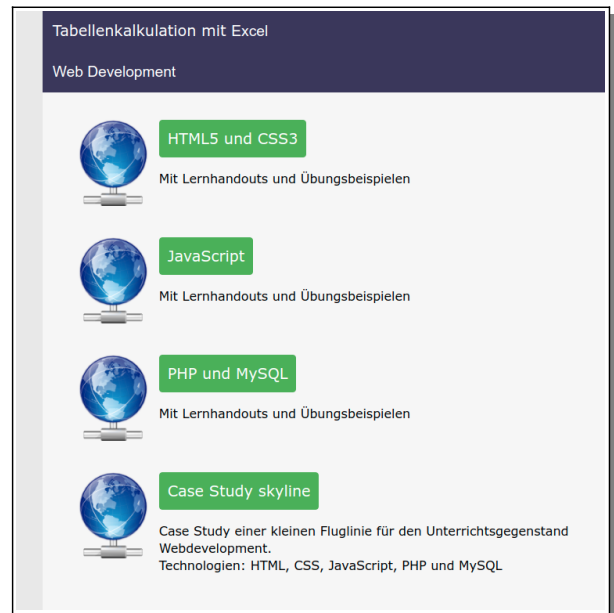


Abbildung: Im Lernbereich

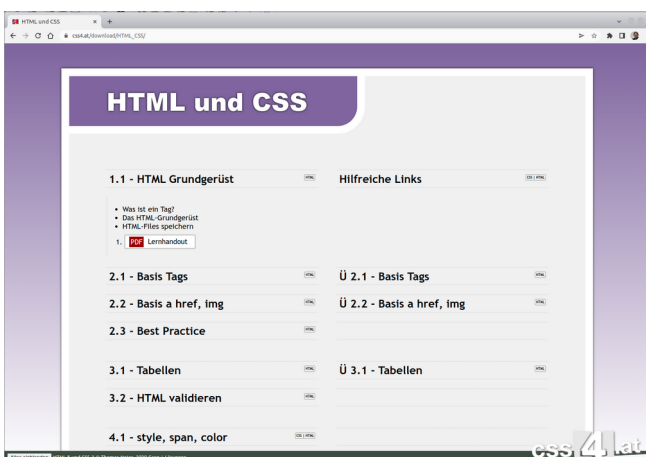


Abbildung: Lernplattform für HTML und CSS

Die Lernplattform für HTML und CSS erlaubt zu jeder Einheit die Downloads der Lern- und Übungshandouts und der benötigten Dateien für die Übungsbeispiele.

Zusätzlich findet man auf der Lernplattform noch hilfreiche Links zum Unterricht. In der linken Spalte findet man die Lernhandouts und korrespondierend rechts die dazu passenden Übungshandouts.

Zu allen Übungsbeispielen gibt es selbstverständlich eine Lösung, die in den meisten Fällen nur als Referenz oder Anschauungsmaterial dient und nicht als absolute Lösung auf Komma und Beistrich herhalten soll. In der Webseitengestaltung gibt es immer mehr als nur einen Weg!

Der Zugang zu den Lösungen erfolgt passwortgeschützt über <https://solved.css4.at/> - dort gibt es auch ein Antragsformular für den Zugriff auf den Lösungsbereich. Die Verwendung von Lern- und Übungshandouts, sowie der Lösungen ist selbstverständlich für Lehrer\_innen von öffentlichen Schulen oder von Organisationen mit öffentlichem Charakter kostenlos und frei verwendbar.

Sollten Sie das Unterrichtsmaterial von [www.css4.at](http://www.css4.at) kommerziell verwenden (z. B. für bezahlte Kurse in der Erwachsenenbildung) dann bitte ich Sie recht höflich mir zuvor eine Mail an [webmail@css4.at](mailto:webmail@css4.at) mit dem didaktischen Umfang ihres Unterrichts (didaktische Konzeption) und dem damit verbunden finanziellen Volumen zuzusenden. Wir finden sicher einen Modus für die Verwendung des Unterrichtsmaterials.

## Sozialformen

Das Unterrichtsmaterial ist auf Einzelarbeit ausgelegt. Dennoch ist Partnerarbeit bei Bildschirmknappheit denkbar. Gruppenarbeit kommt kategorisch nicht vor. Das Plenum wird nur im Rahmen des Frontalunterrichts aktiviert.

Methoden, die andere Sozialformen bedingen, unterliegen der didaktischen Kreativität des\_der Lehrer\_in. Gegen Ende des Unterrichts kann gerne eine Projektarbeit angesetzt werden, die auch in der Gruppe gelöst werden kann.

## Exemplarisches Stundenbild

↓	Dauer	Wann? Zeit bis	Was? Lerninhalte	Wie? Methoden	Wodurch? Medien	Warum? Methode	
Einstieg	2 min	2'	Begrüßung	päd. Gespräch	---	---	
	5 min	7'	Vorbesprechung	päd. Gespräch	---	Erhebung von Interessen	
	6 min	13'	Wiederholung	Mind-Mapping	A3 Bogen oder Whiteboard	Wiederholung der letzten UE.	
Erarbeitung	10 min	23'	Einheit x	Vorzeigen – Nachmachen	Bildschirm	Vermittlung der Skills	
	30 min	53'	Übungsbeispiele Einheit x	freies Arbeiten	Bildschirm	Kombiniert mit individuellen Coaching	
	5 min	58'	PAUSE				
	10 min	68'	Einheit y	Vorzeigen Nachmachen	Bildschirm	Vermittlung der Skills	
	30 min	98'	Übungsbeispiele Einheit y	freies Arbeiten	Bildschirm	Kombiniert mit individuellen Coaching	
	15 min	113'	Individuelles Coaching und freies Arbeiten	Coaching	Bildschirm	Leistungsbeobachtung, Weiterführende Fragen beantworten	
Erfolgssicherung	5 min	118'	Ausblick auf die nächste UE.	päd. Gespräch	---	Die nachgelagerten Themen ansprechen	
	2 min	120'	Verabschiedung	päd. Gespräch	---	Verabschiedung und Herstellung der Arbeitsplatz-Ordnung. (z. B. Herunterfahren des PC)	

## Intensionen der Schüler\_innen

---

Was genau in den Köpfen der Schüler\_innen vorgeht können wir nicht wissen. Würden wir alle Intensionen, Gedanken und Affekte der Schüler\_innen kennen, dann würde auch unsere Didaktik gänzlich anders aussehen – aber so wie es ist, ist es auch die große Herausforderung pädagogischer Arbeit. Hier eine Aufstellung möglicher Intensionen:

Der\_die Schüler\_in will ...

- grundlegende Kenntnisse des Webdevelopments erwerben.
- vertiefende Kenntnisse des Webdevelopments erwerben.
- Webseiten bzw. Apps nach eigenen Vorstellungen entwerfen.
- das Internet mitgestalten.
- eine positive Note bzw. eine gezielte Note für das Unterrichtsfach bekommen.
- die Ausbildung abschließen.
- das erworbene Wissen aus dem Webdevelopment fächerübergreifend in anderen Gegenständen einsetzen.
- mit dem Wissen und den Fähigkeiten Geld verdienen.
- soziale Kontakte erleben.

## Kompetenzbegriff und Zielsetzungen

---

Die Konzeption der Lern- und Lehrunterlagen dieser Arbeit unterliegen dem strengen Gebot der Progressionslogik. Alle Themenbereiche sind aufbauend gegliedert und meist von ihrem Vorgängerthema abhängig. Wichtige Elemente des didaktischen Aufbaus zur Erlernung von HTML & CSS wiederholen sich in den Übungsbeispielen um diese zu festigen.

Die Unterrichtsplanung entspricht nicht gängigen Vorstellungen von Bildungsstandards, ist aber sehr wohl kompetenzorientiert aufgebaut. Es gibt keine Deskriptoren die in eine 4.0 Skala zur Leistungsbewertung übertragen werden können. Dafür gibt es klassische Zielformulierungen (Grob- und Feinziele), die um die Taxonomien "Verstehen", "Anwenden", "Analysieren" und "Entwickeln" erweitert wurden. Die Zielarten werden in "Affektiv, Kognitiv, Psychomotorisch, Selbstkompetenz und Inhaltsebene" differenziert.

Um die Konzeption offen zu halten, gibt es keinen definitiven Lehrplanbezug. Die Lern- und Lehrinhalten sind für die Sekundarstufe II (ISCED Level 3) bzw. der tertiären Bildungstufe optimiert.

## Grobziele und Kompetenzzuwachs

Die Studierenden ...	Zielart	Taxonomie
... erlernen die Verwendung von HTML und CSS.	<b>I</b>	I
... nutzen das Wissen zur Gestaltung von Webseiten.	<b>I</b>	II
... erhalten das Rüstwerkzeug um individuelle Vorstellung vom Webdesign (jenseits von Vorlagen und Templates) umzusetzen.	<b>SIK A</b>	II
... erhalten einen Einblick in die darunterliegende Ebene der IT-konformen Darstellung und betrachten den kritischen letzten Eindruck einer Darstellung.	<b>KI</b>	III
... nutzen die erlernten Fähigkeiten um visuelle Vorstellungen und Gedanken in schriftlich/auditiver Form darzustellen.	<b>KI</b>	IV
... entwickeln ein höheres Professionsbewusstsein.	<b>ASK</b>	IV
... erwerben die Fähigkeit selbstständig ihr Wissen zu erweitern.	<b>SIK</b>	III
... sind befähigt ein mittleres und statisches Web-Projekt zu erledigen.	<b>SI</b>	IV
... steigern ihre kognitiven Leistungen im Bereich des linearen und logischen Denken.	<b>KS</b>	IV
... entwickeln Problemlösungsstrategien.	<b>KS</b>	IV
... steigern ihr Bewusstsein für Ordnung und Genauigkeit.	<b>AS</b>	IV

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

# Handouts und Assignments

Die Handouts sind in Einheiten gegliedert. Jede Einheit beschreibt einen Themenkomplex aus dem jeweiligen Bezugssystem. Grundsätzlich unterliegen die Einheiten einer Progressionslogik, können aber auch voneinander unabhängig unterrichtet werden (je nach pädagogischen Bedarf). Alle Handouts wurden nach dem Prinzip des "One-Page-Management" erstellt und können auch als Kopiervorlage genutzt werden. Die Handouts wurden mit Screenshots bebildert. Die Anleitungen sind auf die Übungsbeispiele hin konzipiert.

Die Symbole auf den Handouts sollen den Schüler\_innen eine Ordnung im Lernprozess vermitteln.



↩ **Übungsbeispiel**



↩ **Information und Hervorhebung**



↩ **Nachschlagen und Bibliothek**

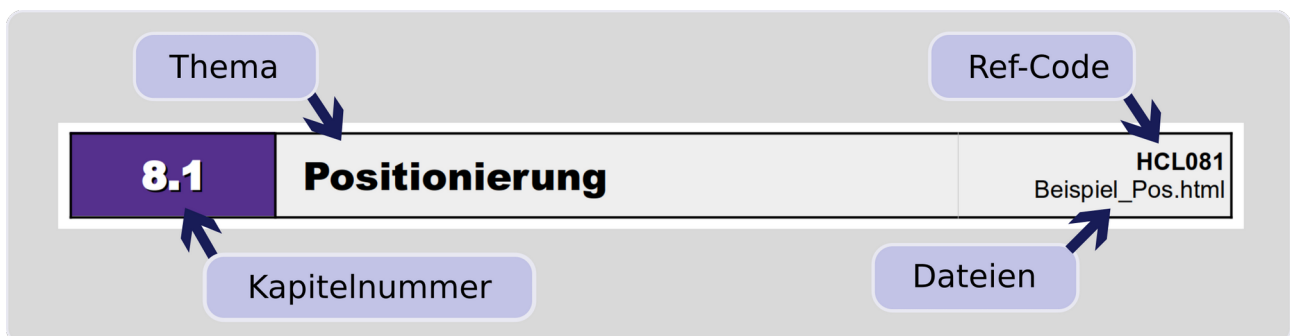


↩ **Tools und Werkzeuge**



↩ **Notizen und Anmerkungen**

Dieses One-Page-Management wurde der Wirtschaft entnommen. Es ist eine Methode um Führungspersonlichkeiten einen raschen Überblick über die aktuellen Fakten zu geben. Rasch und Einfach – was nun also für Manager von großen Betrieben stimmig ist, sollte für den Schüler bzw. die Schülerin genauso gelten. Ein neues Thema – ein neues Blatt Papier.



Über die Kapitelnummer werden Themenbereiche gegliedert. Übungsblätter haben zusätzlich noch ein "Ü" bei der Kapitelnummer. Der Ref-Code identifiziert ein Handout eindeutig und kann im Suchfeld von [www.css4.at](http://www.css4.at) abgerufen werden. Dateien, die für das Thema notwendig sind, findet man als rechts, unter dem Ref-Code. Die Dateien können von [www.css4.at](http://www.css4.at) herunter geladen werden! In dieser Arbeit folgt immer zuerst die didaktische Konzeption (Feinzielen und Anmerkungen) und im Anschluss daran das Schüler\_innen Handout bzw. Übungsblatt.

# Lernhandout 1.1 HTML Grundgerüst

Referenzcode: **HCL011**

Technologien: HTML

Dateinamen: L1\_1\_HTML\_Start.odt, L1\_1\_HTML\_Start.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verstehen die Funktion von Start und End-Tags	K	I
... erkennen die Struktur des HTML Grundkonstrukts mit seinen Elementen.	KI	I
... speichern HTML Seiten mit richtigen Namen und Dateierweiterung.	KI	II
... erstellen eine Vorlage mit dem Grundkonstrukt für spätere Arbeiten.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Workflow: Die Studierenden ...

- ... schreiben ein HTML Grundkonstrukt. Dabei ergänzen sie die META Tags mit dem eigenen Namen und dem heutigen Datum.
- ... sie speichern die Vorlage als HTML File ab und nutzen sie später noch für weitere Arbeiten.

## Fremdquellen

- keine

## Anmerkungen

- Grundsätzlich hat jedes Element in HTML einen Starttag und einen Endtag. Es gibt aber Ausnahmen. Diese heißen **Standalone-Tags** und haben keine Childs und deshalb auch keinen Endtag. Beispiele sind: Die Metatags, `<br>` (Zeilenschaltung), `<hr>` (Horizontale Linie).
- Der UTF-8 Zeichensatz kann Umlaute darstellen. Andere Zeichenkodierungen (z. B. US-Amerikanische) kennen keine Umlaute.
- Die Meta-Tags beschreiben die Webseite und helfen den Suchmaschinen diese bessere zu indexieren. Die Metatags mit dem Autorennamen und dem Datum sollten (allein der Übersicht wegen) immer angeführt werden.
- Das gilt auch für das `<title>` Element. Abermillionen Webseiten sind **untitled** und das ist ziemlich unschön. Der Inhalt des `<title>` Tags wird im Browserfenster oben angezeigt.
- Ob eine `index.html` automatisch angezeigt wird, hängt von den Webserver Einstellung ab. Es können auch `default.html`, `start.html` udgl. als Standard definiert sein!

Ein HTML-Tag beginnt mit einem Start-Tag und endet mit einem End-Tag.  
Dazwischen ist der zu formatierende Bereich (z. B. Text).

Das Grundkonstrukt jeder HTML-Seite beginnt mit einem `<html>` Tag. Darin befindet sich der `<head>` und der `<body>`. Im `<head>` stehen Metadaten wie z. B. der Titel. Im `<body>` ist der Inhalt der Website, der dann im Browser angezeigt wird.

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Titel der Webseite</title>
    <meta name="author" content="John Doe">
    <meta name="date" content="2022-04-20">
  </head>
  <body>
    Hier ist der sichtbare Teil der Website!
  </body>
</html>
```



`<!doctype html>`

↳ Definiert den Code als HTML Seite

`<html lang="de">`

↳ HTML-Starttag mit einem Hinweis auf die Sprache. z. B. de für Deutsch

`<meta charset="utf-8">`

↳ UTF8 Zeichenkodierung mit deutschen Sonderzeichen (ä, ö, ß usw.)

`<title>Titel der Webseite</title>`

↳ Der Titel der Webseite

`<meta name="date" content="2022-04-20">`

↳ Ein Metatag beschreibt das Dokument. z. B. das Erstelldatum



**Speichern unter ...**

HTML Dateien haben die Extension (Dateierweiterung) `*.htm` bzw. `*.html`  
Die `index.html` wird angezeigt, wenn man im Browser die URL eines Ordners eingibt – z. B: `www.johndoe.com/projekte`



**Übung A: Das HTML Grundkonstrukt**

- Starte einen Webeditor (Notepad ++, Dreamweaver, Atom odgl.)
- Erstelle ein HTML Grundgerüst
- Trage deinen Namen als Autor ein
- Speichere es als **vorlage.html** ab



Wir arbeiten in Zukunft immer mit der `vorlage.html` – speichere sie also so ab, dass du immer Zugriff darauf hast (USB-Stick, Cloud, Netzlaufwerk).

# Lernhandout 2.1 Basis Tags

Referenzcode: **HCL021**

Technologien: HTML

Dateinamen: L2\_1\_Basis.odt, L2\_1\_Basis.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verwenden Tags richtig (öffnen und schließen von Tags).	I	II
... definieren Überschriften <h1> bis <h6>	I	II
... kennen Standalone Tags wie z. B. <hr>	K	I
... kennen den Unterschied zwischen   und <p> Elementen	K	I
... führen div. HTML Zeichenformatierungen durch.	I	II
... können Tags verschachteln und kennen das Prinzip von Eltern- und Kindelementen.	K	II
... erstellen HTML Listen und Aufzählungen.	I	II
... wissen, was Attribute sind und nutzen diese für Listen und Aufzählungen.	I	II
... validieren ihren HTML Code online über w3.org	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- HTML Validator auf <http://validator.w3.org> – Bei Validate by Direct Input kann der HTML Code per Copy & Paste eingefügt werden!

## Anmerkungen

- In der Anfangsphase wird man noch zwischen dem Editor und der Browserdarstellung wechseln um die Auswirkungen sofort zu sehen. → F5
- Für Tieferstellungen <sub> kann man eine chemische Formel vorzeigen oder entwickeln lassen. z. B: H<sub>2</sub>SO<sub>4</sub> → `<p>H<sub>2</sub>SO<sub>4</sub></p>`
- Für Höherstellungen <sup> kann man den Pythagoras vorzeigen oder entwickeln lassen. z. B: a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup>  
`<p>a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup></p>`
- Das Type-Attribut im <ul> wird bei einer Validierung als **DEPRECATED** (veraltet) angezeigt und sollte (laut HTML 5) mit CSS gestaltet werden. CSS wird erst später thematisiert – die meisten Browser sollten es trotzdem darstellen.

Es folgen die geläufigsten HTML-Elemente. Nicht vergessen: Jeder Tag muss geschlossen werden! Ausnahmen bilden die Standalone-Tags (z. B. `<br>` oder `<hr>`).

<pre>&lt;h1&gt; Überschrift der 1. Ordnung &lt;/h1&gt; &lt;h6&gt; Überschrift der 6. Ordnung &lt;/h6&gt;</pre>	<pre>&lt;br&gt; ← eine Zeilenschaltung &lt;hr&gt; ← horizontale Linie</pre>
<pre>&lt;p&gt; Absatz &lt;/p&gt;</pre>	<p>Tags können auch verschachtelt werden.</p> <pre>&lt;p&gt;Das hier ist &lt;i&gt;besonders Wichtig&lt;/i&gt; und &lt;b&gt;muss hervorgehoben&lt;/b&gt; werden!&lt;/p&gt;</pre> <p>Im Browser wird folgendes dargestellt: Das hier ist <i>besonders Wichtig</i> und <b>muss hervorgehoben</b> werden!</p>
<pre>&lt;i&gt; kursive Schrift &lt;/i&gt; &lt;b&gt; Fettschrift &lt;/b&gt; &lt;sub&gt; tiefgestellt &lt;/sub&gt; &lt;sup&gt; hochgestellt &lt;/sup&gt; &lt;s&gt; durchgestrichen &lt;/s&gt;</pre>	

<p><b>Nicht nummerierte Liste</b> (unordered list)</p> <p>Beispiel:</p> <ul style="list-style-type: none"> <li>○ HTML lernen</li> <li>○ HTML anwenden</li> <li>○ CSS lernen</li> </ul>	<p><b>Nummerierte Liste</b> (ordered list)</p> <p>Beispiel:</p> <ol style="list-style-type: none"> <li>1 Handout lesen</li> <li>2 Im Internet suchen</li> <li>3 Fragen stellen</li> </ol>
<pre>&lt;ul&gt;   &lt;li&gt;HTML lernen&lt;/li&gt;   &lt;li&gt;HTML anwenden&lt;/li&gt;   &lt;li&gt;CSS lernen&lt;/li&gt; &lt;/ul&gt;</pre>	<pre>&lt;ol&gt;   &lt;li&gt;Handout lesen&lt;/li&gt;   &lt;li&gt;Im Internet suchen&lt;/li&gt;   &lt;li&gt;Fragen stellen&lt;/li&gt; &lt;/ol&gt;</pre>
<p><code>&lt;ul&gt;</code> ← ist das Elternelement <code>&lt;li&gt;</code> ← ist das Kindelement</p> <p><code>&lt;li&gt;</code> ← bedeutet List-Item</p> <p><code>&lt;ul type="circle"&gt;</code> Aufzählung mit nicht ausgefüllten Kreisen</p> <p><code>&lt;ul type="square"&gt;</code> Aufzählung mit Rechtecken</p> <p><b>Achtung:</b> Das <code>Type-Attribut</code> im <code>&lt;ul&gt;</code> Tag wird unter HTML5 nicht mehr unterstützt. Die meisten Browser zeigen es aber trotzdem an.</p>	<p><code>&lt;ol type="I"&gt;</code> Nummerierte Liste mit Römischen Zeichen.</p> <p>Weitere Werte für das <code>type-Attribut</code>:</p> <ul style="list-style-type: none"> <li>1 ... Arabische Zahlen</li> <li>A ... Großbuchstaben</li> <li>a ... Kleinbuchstaben</li> <li>i ... kleine Römische Zeichen</li> </ul> <p><code>&lt;ol start="3"&gt;</code> Startet die nummerierte Liste mit 3.</p> <p><code>&lt;ol reversed&gt;</code> Die Reihenfolge wird umgedreht!</p>



Bei `type=""` im `<ul>` oder `<ol>` Tag handelt es sich um ein Attribut. Attribute bestimmen im HTML-Element zusätzliche Eigenschaften/Informationen. Sie stehen immer im Starttag und werden in der Regel als `Name = "value"` verwendet. (`Name` ist die Bezeichnung des Attributs, `value` ist der Wert)



**Web-Tipp: Der HTML Validator vom W3C**

<http://validator.w3.org> ← den eigenen HTML Code auf Fehler überprüfen

# Übungsblatt 2.1 Basis Tags

Referenzcode: **HCU021**

Technologien: HTML

Dateinamen: U2\_1\_Basis.odt, U2\_1\_Basis.pdf, LW\_Salzburg\_2013.txt

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... verwandeln ein TXT File in ein HTML Dokument	I	II
... arbeiten mit dem HTML Grundkonstrukt	I	II
... führen div. Elementzuweisungen und Zeichenformatierungen durch	I	II
... erstellen Listen und Aufzählungen	I	II
... begeistern sich für strukturiertes Arbeiten.	A	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Workflow: Die Studierenden ...

- ... verwandeln einen Plain-Text in ein HTML Dokument. Dazu fügen sie um den Text ein HTML-Konstrukt und verteilen dann die entsprechenden Elemente.

## Anmerkungen

- Die Datei ‚LW\_Salzburg\_2013.txt‘ sollte bereitgestellt werden. Dazu eignen sich Lernplattformen wie Moodle, Cloud-Dienste, der Schul-Server uvm. Die Datei kann auch über die Suche von [www.css4.at](http://www.css4.at) gefunden werden!
- Lehrer\_innen aus Deutschland mögen bitte verzeihen, dass die Übungsbeispiele einen starken Österreich Bezug haben.
- Im Rahmen des Unterrichtsprinzip „politische Bildung“ kann auf die grundsätzlichen Ausrichtungen der unterschiedlichen Parteien näher eingegangen werden. Das Beispiel wurde bewusst so gewählt, das eine hohe Partei-Diversität vorkommt.
- Es sollte auch von Anfang an schon viel Wert auf die Strukturierung des Codes gelegt werden. So rückt man ein Kindelement (z. B. <li>) um zwei Leerzeichen ein!
- Wichtig ist auch ein Verständnis für Whitespace (Leerzeichen, Zeilenschaltungen) in HTML. In dieser Phase spielt es noch keine Rolle wie viele Leerzeichen tatsächlich im Code sind – es wird immer nur eines angezeigt!  
Gleiches gilt auch für Zeilenschaltungen.
- Die Übung eignet sich gut für Vorzeigen und Nachmachen. Besonders begabte Schüler\_innen nutzen Suchen-und-Ersetzen. z. B. Salzburg → <b>Salzburg</b>



### Übung A: TXT in HTML

Öffne die Datei ‚LW\_Salzburg\_2013.txt‘ mit einem Texteditor und speichere diese als ‚LTW\_Salzburg\_2013.html‘ ab.

- Füge um den Text das HTML-Grundkonstrukt ein.

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Landtagswahl Salzburg 2013</title>
    <meta name="autor" content="Vorname Nachname">
    <meta name="date" content="jjjj-mm-tt">
  </head>
  <body>
    {Der Inhalt von LW_Salzburg_2013.txt}
  </body>
</html>
```

- Weise den Überschriften die passenden Tags zu. <h1> bis <h3>
- Verteile <p> Tags im Text um Absätze zu bestimmen.
- Die Aufzählung mit den Parteien soll als "nicht nummerierte Liste" (unordered list) dargestellt werden. <ul> und <li>

#### Folgende Parteien traten zur Wahl an:

- BZÖ - Bündnis Zukunft Österreich
- FPÖ - Freiheitliche Partei
- Grünen - Die Alternativen
- KPÖ - Kommunistische Partei Österreich
- ÖVP - Österreichische Volkspartei
- PIRAT - Piratenpartei
- SPÖ - Sozialdemokraten
- TS - Team Stronach

- Die Liste mit den Verteilungen der Sitze soll als "nummerierte Liste" (ordered list) dargestellt werden! <ol> und <li>

#### Verteilung der Sitze

Nach der Wahl wurden die Sitze im Landtag wie folgt verteilt:

1. ÖVP - 11 Sitze (Schwarz)
2. SPÖ - 9 Sitze (Rot)
3. Grüne - 7 Sitze (Grün)
4. FPÖ - 6 Sitze (Blau)
5. TS - 2 Sitze (Gelb)

- Alle Vor- und Nachnamen in dem Text sollen kursiv dargestellt werden. <i>
- Das Wort "Salzburg" soll immer fett geschrieben werden. <b>

# Lernhandout 2.2 Basis Elemente (img, a href)

Referenzcode: **HCL022**

Technologien: HTML

Dateinamen: L2\_2\_Basis.odt, L2\_2\_Basis.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... binden Hyperlinks ein.	I	II
... verstehen und nutzen das href Attribut	I	III
... verstehen und nutzen das target Attribut	I	III
... binden Bilder in den HTML Code ein.	I	II
... verstehen und nutzen die Attribute des <img> Elements.	I	III
... möchten das Internet barrierefreier mitgestalten (alt Attribut des <img>).	A	II
... kennen Bildtypen und differenzieren durch ihre Eigenschaften.	K	III

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Bilder für "Vorzeigen und Nachmachen" müssen selbst organisiert werden. Eine gute Quelle dafür wäre z. B. [www.pixabay.com](http://www.pixabay.com)
- Hier wäre es gut auch die Urheberrechte (von Bildern) zu thematisieren. Was ist erlaubt? Was ist verboten? Welche Lizenzen gibt es (z. B. CommonCreative) usw.
- Die Schüler\_innen sollte sich angewöhnen bei Bildern **immer** ein alt Attribut anzuführen. Dieses ist ein wichtiger Beitrag für das Barrierefreie Web (Web-Reader)! Die Attribute title, height und width sind nicht zwingend notwendig.
- Hier die Eigenschaften der wichtigsten Bildformate:
  - \*.gif → 256 Farben, geringer Speicherbedarf, transparenter Hintergrund (eine Farbe transparent schalten), animierbar.
  - \*.png → hohe Qualität mit transparenten Hintergrund
  - \*.jpg → sehr hohe Qualität, stark komprimierbar – dadurch eine geringe Dateigröße, keine Transparents
  - \*.svg → für Vektorgraphiken, frei skalierbar ohne Qualitätsverlust.

## Hyperlinks

## HTML

```
<a href="" target="" > ... </a>
```

```
<a href="https://www.google.at" target="_blank">Google Suche</a>
```



Hyperlinks sind Querverweise zu anderen Webseiten oder Dateien.

**href=""**

Innerhalb der Anführungszeichen wird der Verweis eingetragen.

Ein absoluter Link ist ein Hyperlink zu einer fixen Adresse

z. B. "<https://www.google.at>"

Der relative Link verweist abhängig vom Speicherort

z. B.: "[../html/grundbegriff.html]( ../html/grundbegriff.html)"

"mailto: john@google.at" ← mit E-Mail-Client öffnen

"skype: johndoe90" ← für Skype

"tel:+4301511452" ← für Telefonnummern

**target=""**

Legt das Zielfenster für die Verweise fest.

`_self` öffnet im aktuellen Fenster (Standardwert)

`_blank` öffnet in neuem Fenster.

`_parent` Elternfenster

`_top` oberstes Fenster

## Bilder

## HTML

```
<img src="" alt="" title="" height="" width="">
```

```

```



Bilder werden mit dem `<img>` Tag eingefügt. Es handelt sich dabei um ein Standalone-Tag ohne Elementinhalt und ohne End-Tag!

**src=""**

Verweis auf den Speicherort des Bildes.

(relativ oder absolut)

**alt=""**

Beschriftung des Bildes.

**title=""**

Zeigt einen Tooltip, wenn man mit der Maus darüber fährt.

**height=""**

Höhe des Bildes (in Pixel) ohne einer Angabe wird die Originalhöhe des Bildes verwendet.

**width=""**

Breite des Bildes (in Pixel) ohne einer Angabe wird die Originalbreite des Bildes verwendet.



Ein Hyperlink kann auch auf ein Bild gelegt werden:

```
<a href="https://www.google.at" >
  
</a>
```



Die wichtigsten Bildtypen im Web sind: `*.gif`, `*.png`, `*.jpg`, `*.svg`

# Übungsblatt 2.2 Basis a href, img

Referenzcode **HCU022**

Technologien HTML

Dateinamen U2\_2\_Basis.odt, U2\_2\_Basis.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen ein Inhaltsverzeichnis mit HTML.	I	II
... verschachteln und kombinieren von Listen und Aufzählungen.	K	II
<b>Übung B</b>		
... erstellen eine Top-Ten Aufstellung mit HTML.	I	II
... reflektieren auf ihre Lieblingsseiten (bevorzugte) im Internet.	A	III
<b>Übung C</b>		
... binden Bilder in den HTML Code ein.	I	II
... kombinieren Bilder mit Hyperlinks.	I	II
... nutzen Bilder um zu emotionalisieren (in Verbindung mit Hyperlinks als Buttons).	A	II
... kennen mind. vier Suchmaschinen.	K	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Sollte das Inhaltsverzeichnis auf dem Bild nicht lesbar sein.

Hier die Kopiervorlage:

### I. Einleitung

Einige Worte zur Begrüßung

### II. Projektbeschreibung

Ein kurzes Abstract über das Projekt

### III. Projektplanung

- Ziele setzen
- Soll-Ist-Vergleich
- Ziele überprüfen

### IV. Realisierung

Das Projekt wird ausgeführt

### V. Checkliste

- Kopieren
- Verteilen
- Erklären
- Abhacken

### VI. Risikomanagement



### Übung A: Ein Inhaltsverzeichnis

- Erstelle das HTML-Grundgerüst mit deinem Namen und dem aktuellen Datum. Der Titel ist Inhaltsverzeichnis.
- Schreibe ein Inhaltsverzeichnis! (wie neben dargestellt).
- Die Listen müssen verschachtelt sein!
- Du benötigst folgende Tags: `<h1>`, `<o1>`, `<ul>`, `<li>`, `<b>`, `<s>`, `<br>`
- Als HTML-Dokument speichern!**

### Inhaltsverzeichnis

- I. **Einleitung**  
Einige Worte zur Begrüßung
- II. **Projektbeschreibung**  
Ein kurzes Abstract über das Projekt
- III. **Projektplanung**
  - a. Ziele setzen
  - b. Soll-Ist-Vergleich
  - c. Ziele überprüfen
- IV. **Realisierung**  
Das Projekt wird ausgeführt
- V. **Checkliste**
  - Kopieren
  - Verteilen
  - Erklären
  - Abhacken
- VI. **Risikomanagement**



### Übung B: Top-Ten Websites

- Erstelle das HTML-Grundgerüst mit deinem Namen und dem aktuellen Datum. Der Titel ist Top-Ten.
- Schreibe eine "Top-Ten Liste" von beliebten Webseiten (wie neben dargestellt).
- Verwende hier Hyperlinks zu den Webseiten!
- Vollende die Liste bis Platz 1 mit deinen Lieblingsseiten im Netz!
- Du benötigst folgende Tags: `<h1>`, `<o1>`, `<li>`, `<a href>`, `<s>`, `<br>`
- Als HTML-Dokument speichern!**

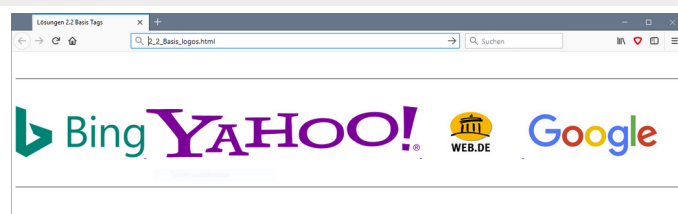
### Top-Ten Websites

- 10. [Selfhtml.org](#)
- 9. [HTML Code validieren](#)
- 8. [Google Suche](#)
- 7. [Facebook](#)
- 6. ...



### Übung C: Suchmaschinen

- Erstelle das HTML-Grundgerüst mit deinem Namen und dem aktuellen Datum. Der Titel ist Suchmaschinen.
- Suche im Internet nach den Logos von mind. vier Suchmaschinen.
- Die Logos sollen nebeneinander dargestellt werden (siehe unten).
- Mit einem Klick auf das Logo soll sich die jeweilige Suchmaschine in einem neuen Browserfenster öffnen.
- Du benötigst folgende Tags: `<img>`, `<a href>`
- Als HTML-Dokument speichern!**



# Lernhandout 2.3 Best Practice

Referenzcode **HCL023**

Technologien HTML

Dateinamen L2\_3\_BePr.odt, L2\_3\_BePr.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... strukturieren den HTML Code lesbarer.	I	II
... haben ein gesteigertes Ordnungsbewusstsein.	A	II
... erkennen die Struktur von Kind und Elternelementen.	K	I
... begreifen, dass es zu Problemen kommen kann, wenn man HTML-eigene Zeichen für den darstellbaren Text verwendet.	K	I
... nutzen die HTML Zeichenreferenz.	I	II
... verwenden HTML Kommentare um Anmerkungen im Code zu platzieren.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- HTML Zeichenreferenz von selfhtml.org  
<https://wiki.selfhtml.org/wiki/Zeichenreferenz>

## Anmerkungen

- Das `&nbsp;` ist ein geschütztes Leerzeichen. Wenn man z. B. mehr als ein Leerzeichen benötigt, dann verwendet man `&nbsp;`. Außerdem sollte es verwendet werden, wenn man ein leeres Element hat. So z. B. bei einer leeren Absatzschaltung `<p>&nbsp;</p>`.

Dem Browser ist es relativ egal wie man den Code strukturiert. Zeilen- und Absatzschaltungen werden im Browser nicht dargestellt – dafür verwendet man `<br>` oder `<p>`. Man kann theoretisch auch eine Website in nur einer Zeile schreiben:

```
<body><div>Aufzählung<ul><li>Punkt 1</li><li>Punkt 2</li></ul></div></body>
```

Um nun aber den Code für den Menschen lesbarer zu machen, rückt man die Child-Elemente (Kindelemente) mit Leerzeichen ein:

<code>&lt;body&gt;</code>	← Parent (Elternelement)
<code>&lt;div&gt;Aufzählung</code>	← First Child (Kindelement) von <code>&lt;body&gt;</code>
<code>&lt;ul&gt;</code>	← Child von <code>&lt;div&gt;</code> , Parent von <code>&lt;li&gt;</code> , Sibling von <code>&lt;p&gt;</code>
<code>&lt;li&gt;Punkt 1&lt;/li&gt;</code>	← Child von <code>&lt;ul&gt;</code>
<code>&lt;li&gt;Punkt 2&lt;/li&gt;</code>	
<code>&lt;/ul&gt;</code>	
<code>&lt;p&gt;Etwas Text&lt;/p&gt;</code>	← Second Child von <code>&lt;div&gt;</code> , Sibling (Geschwister) von <code>&lt;ul&gt;</code>
<code>&lt;/div&gt;</code>	
<code>&lt;/body&gt;</code>	



Der `<div>` Tag ist ein Container Element! (engl. division ↔ Bereich).

## Besondere Zeichen

Wenn man im Text Zeichen verwenden möchte, die in HTML eine Bedeutung haben (`<`, `>`, `&`, `"`), dann verwendet man HTML-eigene Zeichen. Beispiele dafür sind:



<code>&lt;</code>	..... öffnende Spitze Klammer (lower than)	..... <code>&amp;lt;</code>
<code>&gt;</code>	..... schließende Spitze Klammer (greater than)	..... <code>&amp;gt;</code>
<code>&amp;</code>	..... Kaufmännisches Und (Ampersand)	..... <code>&amp;amp;</code>
<code>"</code>	..... Anführungszeichen oben	..... <code>&amp;quot;</code>
	..... geschütztes Leerzeichen (non-breaking space)	..... <code>&amp;nbsp;</code>



Ein umfangreiche Zeichenreferenz findet man bei [selfhtml.org](https://wiki.selfhtml.org/wiki/Zeichenreferenz)  
<https://wiki.selfhtml.org/wiki/Zeichenreferenz>

## HTML Kommentare

Der Browser ignoriert Kommentare und stellt diese auch nicht dar. Sie dienen dem Autor um innerhalb des Codes Anmerkungen zu platzieren oder Codeteile auszukommentieren. Eingeleitet wird ein Kommentar mit `<!--` dann folgt der Kommentartext. Abgeschlossen wird es mit `-->`

```
<!-- einzeliger Kommentar -->
<p>viel Text</p>
<!-- und das ist ein mehrzeiliger Kommentar zu dem Text mit <p>...</p>
      Letzte Zeile des Kommentars -->
```

# Lernhandout 3.1 Tabellen

Referenzcode: **HCL031**

Technologien: HTML

Dateinamen: L3\_1\_Tabellen.odt, L3\_1\_Tabellen.pdf, Testtabelle.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... kennen den Aufbau einer Tabelle (Zeilen, Spalten, Zellen).	<b>K</b>	I
... kennen den Aufbau einer HTML-Tabelle.	<b>K</b>	II
... scripten HTML Tabellen.	<b>I</b>	II
... nutzen Tabellenköpfe zur besseren Gliederung.	<b>I</b>	II
... verbinden Spalten zu einer Tabellenzelle (colspan).	<b>I</b>	II
... verbinden Zeilen zu einer Tabellenzeile (rowspan).	<b>I</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das `border` Attribut wird benutzt, um die Tabelle sichtbar zu machen. Es ist in HTML5 obsolet (deprecated) – sollte aber problemlos von den meisten Browsern dargestellt werden.
- Tabellen wurden früher (xHTML) für das Layout verwendet. Das ist seit HTML5 ein No-Go. Besonders beliebt war eine einzige Zelle mit einer Höhe und Breite von 100%. Weil diese Zelle das `vertical-align: middle` und `text-align: center` versteht, konnte so der Content genau im Zentrum des offenen Browserfenster dargestellt werden.
- Die Arbeit mit Tabellen ist ein wichtiges Thema. Nicht nur, weil sich dadurch Inhalte besser strukturieren lassen - sie soll auch ein besseres Verständnis für Tabellenkalkulationen (z. B. Excel) vermitteln. Was sich schlussendlich auch in einem besseren Verständnis für MySQL und andere Datenbanken äußert.
- Die Tags `<thead>` und `<tbody>` sind nicht zwingend notwendig - sind aber überaus hilfreich bei einer Formatierung mit CSS. Darüber gibt es auch noch für den Tabellenfuß das `<tfoot>` Element.

Man verwendet zur übersichtlichen Darstellung von Daten Tabellen. Diese sind in Zeilen (waagrecht) und Spalten (senkrecht) gegliedert. Am Schnittpunkt einer Zeile und einer Spalte ist eine Zelle. HTML-Tabellen werden nur verwendet, um tabellarische Daten darzustellen, nicht, um das Layout einer Website zu gestalten.

```
<table>
  <thead>
    <tr> <th> </th> <th> </th> <th> </th> </tr>
  </thead>
  <tbody>
    <tr> <td> </td> <td> </td> <td> </td> </tr>
    <tr> <td> </td> <td> </td> <td> </td> </tr>
  </tbody>
</table>
```



- <table>** leitet eine Tabelle ein
- <thead>** Der Tabellenkopf ist für eine bessere Gliederung angedacht. Der Tag wird für die Beschriftung einer Tabelle verwendet und ist für die Darstellung nicht zwingend notwendig.
- <tbody>** Der Tabellenkörper ist ebenfalls für eine bessere Gliederung der Inhalte und kann genauso wie der **<thead>** weggelassen werden.
- <tr>** leitet eine Tabellenzeile ein (tr = table row = Tabellenzeile)
- <th>** eine Zelle im Tabellenkopf (th = table header)
- <td>** eine normale Datenzelle (td = table data)



Damit unsere Tabellen sichtbar werden, schreiben wir in den **<table>** Tag das Attribut **border="2"**. **Achtung: Diese Vorgehensweise ist in HTML5 obsolet da Rahmen mit CSS definiert werden müssen!**

Die Zellen lassen sich selbstverständlich auch verbinden. Dies wird mit den Attributen **colspan** bzw. **rowspan** im **<td>** Tag umgesetzt.

- <td colspan="3">** **colspan** (Spalte überspannen) erlaubt eine Tabellenzelle nach rechts über mehrere Spalten auszudehnen (hier über 3 Zellen)
- <td rowspan="2">** **rowspan** (Zeile überspannen) verbindet Tabellenzellen über mehrere Zeilen (hier über 2 Zeilen)

```
<table border="2">
  <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr>
  <tr> <td> </td> <td colspan="3"> </td> </tr>
  <tr> <td rowspan="2"> </td> <td> </td> <td> </td> <td> </td> </tr>
  <tr> <td> </td> <td> </td> <td> </td> </tr>
</table>
```

# Übungsblatt 3.1 Tabellen

Referenzcode: **HCU031**

Technologien: HTML

Dateinamen: U3\_1\_Tabellen.odt, U3\_1\_Tabellen.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen einen Stundenplan mit HTML.	I	II
... verwenden das <code>colspan</code> Attribut.	I	II
<b>Übung B</b>		
... nutzen Tabellen zur Gegenüberstellung von Informationen.	I	II
... binden Bilder in eine Tabelle ein.	I	II
... nutzen das Internet zur Erhebung von Informationen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Anmerkung zu Übung A Stundenplan: Es ist ein sehr umfangreiches Übungsbeispiel. Natürlich ist dieses Beispiel eines, welches die Eigeninitiative des\_der Schüler\_in wecken sollte. So gedacht, könnte man einen beliebigen Wochenplan (z. B: Reservierung eines Tennisplatzes udgl.) als Übungsbeispiel anwenden. Der Stundenplan jedoch wird noch in späteren Übungsbeispielen verwendet. Man kann das Beispiel bis zur 3 Schulstunde begrenzen (Hauptsache das `colspan` kommt zur Anwendung).
- Anmerkung zu Übung B Ländervergleich: Die Bilder der Landesflaggen und das Wikipedia-Logo sollten aus dem Internet heruntergeladen werden. Empfehlenswert wäre dabei, Bilder in der gleichen Auflösung (Größe) zu verwenden - denn sonst müssten die Schüler\_innen mit dem obsoleten Attribut `height` bzw. `width` arbeiten.
- Hier nochmals die Zeilenbeschriftungen für den Ländervergleich:  
Land, Amtssprache, Hauptstadt, Fläche, Einwohner, Staatsform, Währung, Internet-TLD, Telefon-Vorwahl.



### Übung A: Stundenplan

- Erstelle mit dem `<table>` Element deinen Stundenplan.
- Im Tabellenkopf sind folgende Beschriftungen anzuführen: **Stunde | von | bis | Montag | Dienstag | Mittwoch | Donnerstag | Freitag | Samstag**
- Ändere die Unterrichtszeiten (wenn nötig) und verbinde die Zellen für die Pausen mit `colspan`
- **Trage deine Unterrichtsfächer in die freien Zellen ein.**

Stunde	von	bis	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
1.	07 <sup>45</sup>	08 <sup>35</sup>						
2.	08 <sup>35</sup>	09 <sup>25</sup>						
	09 <sup>25</sup>	09 <sup>40</sup>	Pause (15 Minuten)					
3.	09 <sup>40</sup>	10 <sup>30</sup>						
4.	10 <sup>30</sup>	11 <sup>20</sup>						
	11 <sup>20</sup>	11 <sup>30</sup>	Pause (10 Minuten)					
5.	11 <sup>30</sup>	12 <sup>20</sup>						
6.	12 <sup>20</sup>	13 <sup>10</sup>						
	13 <sup>10</sup>	13 <sup>15</sup>	Pause (5 Minuten)					
7.	13 <sup>15</sup>	14 <sup>05</sup>						
8.	14 <sup>05</sup>	14 <sup>55</sup>						
	14 <sup>55</sup>	15 <sup>05</sup>	Pause (10 Minuten)					
9.	15 <sup>05</sup>	15 <sup>55</sup>						
10.	15 <sup>55</sup>	16 <sup>45</sup>						
11.	16 <sup>45</sup>	17 <sup>35</sup>						



### Übung B: Ländervergleich

- Erstelle mit dem `<table>` Element einen Ländervergleich (Schweiz, Österreich, Deutschland) wie unten dargestellt.
- Durch einen Klick auf die Landesflagge soll sich die jeweilige Wikipedia-Seite zum passenden Land öffnen.
- Füge noch ein weiteres Land deiner Wahl dazu!**

			
Land	Schweiz	Österreich	Deutschland
Amtssprache	Deutsch	Deutsch	Deutsch
Hauptstadt	Bern	Wien	Berlin
Fläche	41 285 km <sup>2</sup>	83 878 km <sup>2</sup>	357 385 km <sup>2</sup>
Einwohner	8,4 Mio	8,7 Mio	82,5 Mio
Staatsform	föderale Republik	Bundesrepublik	Bundesrepublik
Währung	Schweizer Franken	Euro	Euro
Internet-TLD	.ch, .swiss	.at	.de
Telefon-Vorwahl +41		+43	+49

# Lernhandout 3.2 HTML validieren

Referenzcode: **HCL032**

Technologien: HTML

Dateinamen: L3\_2\_Validieren.odt, L3\_2\_Validieren.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... kennen Unterschiede zwischen den Browsern.	I	I
... kennen das W3C Konsortium und seine Bedeutung für das Web.	I	I
... validieren (überprüfen) ihren HTML Code.	I	II
... kennen weitere Dienste des w3c (z. B. Link Checker).	I	I
... haben ein gesteigertes Bewusstsein für Genauigkeit.	A	II
... erstellen Screenshots.	I	II
... verstehen die Fehlermeldungen des Validierungsergebnisses.	I	III
... korrigieren Fehler des HTML Codes.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- HTML Validator des W3C - <https://validator.w3.org>

## Anmerkungen

- Hier ist ein didaktisch interessantes Feld beheimatet. Durch die Validierung bekommen die Schüler\_innen ein sofortiges Feedback auf ihre Schreibe mit einer Erklärung zum Fehler bzw. der Warnung. Nur so kann das Lernkonzept von "Try-and-Fail" umgesetzt werden, da die meisten Browser fehlerhaften Code parsen und darstellen (z. B: fehlenden End-Tag). Dadurch wird der Fokus vom Lernprodukt auf den Lernprozess verlegt.
- Es sollte auch gezeigt werden, dass jeder Browser den Quellcode anzeigen kann. Dabei kann auch schon ein kurzer Blick auf die Entwicklertools des Browser geworfen werden. Tastenkombination: STRG + U
- Die meisten IDE's und Texteditoren validieren den HTML Code ebenso. Dennoch, das Validationsergebnis von W3C genießt das größte Ansehen und Vertrauen.

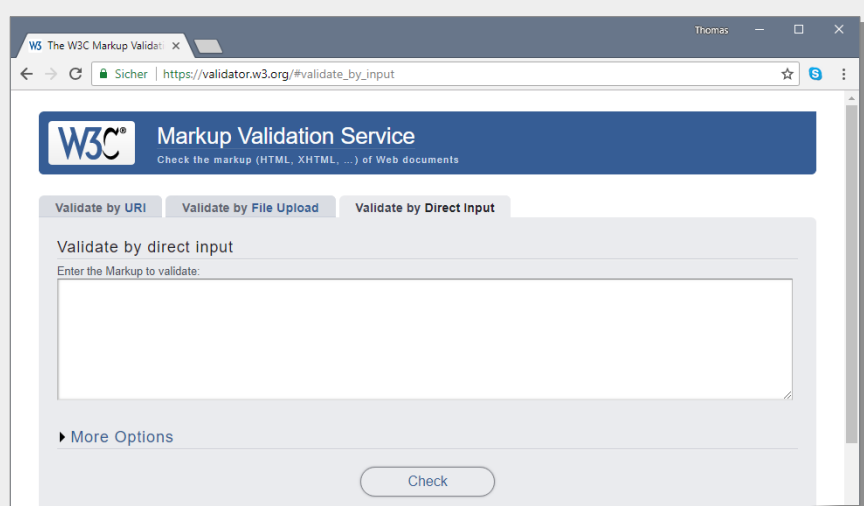
HTML Code wird in der Regel von einem Browser interpretiert. Jeder Browser hat seine Eigenheiten – Firefox ist nicht Chrome – Chrome ist nicht Edge usw. Manche Elemente oder Attribute gibt es nur für bestimmte Browser. Wir wollen aber für alle schreiben. Wie können wir nun sicher sein, dass unser Code richtig ist? Dafür gibt es das W3C. Das W3C Konsortium entwickelt Standards und Richtlinien für das Web. Die meisten Browser-Entwickler halten sich an diese Richtlinien.

Wie können wir nun aber sicher sein, dass unser Code richtig ist?

Dafür hat das W3C auf seiner Website ein **Markup Validation Service** (Prüfungsdienst, Validator) eingerichtet.

Auf [www.w3.org](http://www.w3.org) findet man auch weitere Prüfdienste wie z. B. für CSS oder einen Link Checker.

Der **HTML-Validator** bietet drei Möglichkeiten an um den HTML Code zu überprüfen.



<https://validator.w3.org>



**Validate by URI**.....Man gibt die Adresse zu einem Online Dokument an.

**Validate by File Upload**.....Hier ladet man das HTML Dokument hoch.

**Validate by Direct Input**.....Man kopiert den HTML Code ins Textfeld.

Check

Nach einem Klick auf Check wird der Code überprüft und man bekommt einen umfangreichen Bericht.

1. **Error** End tag `h2` seen, but there were open elements.

[From line 10, column 27, to line 10, column 31](#)

ein Stelle `</h2>` → `</bod`



**Error** bedeutet Fehler und muss unbedingt korrigiert werden!

Document checking completed.

Source

```
1. <!doctype html>↵
2. <html lang="de">↵
3.   <head>↵
4.     <meta charset="utf-8">↵
5.     <title>Übungen 4.3</title>↵
6.     <meta name="author" content="Thomas Maier">↵
7.     <meta name="date" content="2018-04-26">↵
8.   </head>↵
9.   <body>↵
10.    <h1>Führerschein Stelle</h2>↵
11.  </body>↵
12. </html>
```



### Übung A: HTML Code validieren

- Überprüfe den HTML Code von der **Übung 2.2 A: Ein Inhaltsverzeichnis**.
- Tipp: Jeder Browser kann den Quelltext anzeigen.
- Erstelle einen Screenshot vom Prüfergebnis.

# Lernhandout 4.1 style, span, color

Referenzcode **HCL041**

Technologien HTML, CSS

Dateinamen L4\_1\_Color.odt, L4\_1\_Color.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... arbeiten zum ersten Mal mit CSS.	I	II
... nutzen das style Attribut für Inline-Styles.	I	II
... kennen den Aufbau von CSS Eigenschaften.	K	I
... verändern Schriftfarben mit CSS.	I	II
... verändern die Hintergrundfarbe eines HTML Elements mit CSS.	I	II
... arbeiten mit dem <span> Tag.	I	II
... kennen die Möglichkeiten zur Farbzweisung. (Farbname, RGB, HEX)	I	II
... generieren eigene Farben durch RGB bzw. HEX.	I	IV
... verstehen weitere Farbmodelle (z. B: CMYK für den Druck)	K	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene


**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Ein beliebiges "Color-Picker-Tool" hilft hier die Farben zu ermitteln. Manchmal werden HEX Werte auch als HTML Farben angegeben.
- RGB steht für ROT, GRÜN und BLAU. Dieses Farbmodell wird von gängigen Monitoren verwendet. Jeder Wert reicht von 0 bis 255 - also kann RGB ca. 16,7 Mio Farben anzeigen.


Bislang hatten unsere HTML Seiten nur das Standard Erscheinungsbild des Browsers. Um nun einem Element ein besonderes Design zu verleihen, verwendet man das `style` Attribut. Das `style` Attribut gilt für nahezu alle HTML Tags und hat folgende Schreibweise:

## CSS Styles


<b>HTML</b>	<code>&lt;tagname style="property: value;"&gt;</code>
	<code>property</code> ist der CSS-Befehl (Eigenschaft), <code>value</code> ist der Wert. Wichtig ist das Semikolon (;) am Ende jeder CSS Anweisung. Im folgenden Beispiel wird die Schriftfarbe des Textes eines <code>&lt;p&gt;</code> Tags auf die Schriftfarbe rot gesetzt.
<b>CSS</b>	<code>color: [Wert];</code>
<pre>&lt;p style="color: red;"&gt;Die Schrift ist jetzt rot&lt;/p&gt;</pre>	

Man kann auch mehrere CSS Anweisungen (Stile) mit dem `style` Attribut zuweisen.

<b>CSS</b>	<code>background-color: [Wert];</code> ← Hintergrundfarbe (siehe Farben unten)
<pre>&lt;p style="color: white; background-color: blue;"&gt;   Der Schnee ist weiß und der Himmel ist blau.&lt;/p&gt;</pre>	

<b>HTML</b>	<code>&lt;span&gt; ... &lt;/span&gt;</code>
	Wenn man nur ein Wort oder eine Textstelle auszeichnen möchte, benutzt man das <code>&lt;span&gt;</code> Element (span = engl. überspannen). Der <code>&lt;span&gt;</code> Tag alleine bewirkt nichts – erst in Kombination mit CSS erfüllt er eine Aufgabe!
<pre>&lt;h1&gt;Die Lehre von &lt;span style="color: green;"&gt;Energie&lt;/span&gt; und   &lt;span style="color: red;"&gt;Wärme&lt;/span&gt;&lt;/h1&gt;</pre>	

## Farben

<b>CSS</b>	<code>color: [Wert];</code>
	Um den <b>WERT</b> (value) für Farben festzulegen, gibt es 3 wichtige Möglichkeiten:
<b>1. Farbname</b>	Die Namen werden in Englisch ausgeschrieben <code>color: DeepSkyBlue;</code>
<b>2. RGB</b>	RGB (Rot, Grün, Blau) ist das Standard Farbmodell für Bildschirme. Die einzelnen Werte reichen von 0 bis 255. <code>color: rgb(0, 191, 255);</code>
<b>3. HEX</b>	Hier werden die RGB Werte Hexadezimal dargestellt. <code>color: #00bfff;</code>



Weitere Farbmodelle sind HSL, HWB und CMYK. CMYK (Cyan, Magenta, Yellow und Schwarz) ist ein Farbmodell für den Druck.

# Lernhandout 4.2 Rahmen

Referenzcode **HCL042**

Technologien CSS

Dateinamen L4\_2\_Rahmen.odt, L4\_2\_Rahmen.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... definieren Rahmen durch Dicke, Typ und Farbe.	I	II
... bestimmen einzelne Rahmenbereiche (z. B: oben durch border-top).	I	II
... runden Rahmenecken mit <b>border-radius</b> ab.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die CSS Eigenschaft **border** ist ein '**shorthand-property**' und verbindet **border-width**, **border-style**, **border-color** zu einer Zuweisung.  
Das bedeutet: `border: 1px dotted green;` kann auch so dargestellt werden:  
`border-width: 1px;`  
`border-style: dotted;`  
`border-color: green;`

Mit der `border` Eigenschaft (CSS) kann man den Rahmen um ein Element bestimmen. `border` hat drei Werte: **Rahmendicke**, **Rahmentyp** und **Rahmenfarbe**! Zwischen den Werten wird ein Leerzeichen gesetzt. Am Ende ein Semikolon (;).

## CSS

```
border: [dicke] [typ] [farbe];
```



**[dicke]** Die Rahmendicke kann in Pixel angegeben werden (z. B: 5px) bzw. `thin` (dünn) `medium` (mittel) oder `thick` (dick). Die Rahmendicke alleine reicht noch nicht für eine Darstellung – es muss der Rahmentyp angegeben werden!

**[typ]** Folgende Rahmentypen kennt CSS:

<code>solid</code>	.....	fester Rahmen, ohne besondere Erscheinung
<code>dotted</code>	.....	gepunkteter Rahmen
<code>groove</code>	}	simuliert einen 3D Effekt!
<code>ridge</code>		
<code>inset</code>		
<code>outset</code>		
<code>double</code>	.....	doppelter Rahmen
<code>dashed</code>	.....	gestrichelter Rahmen

Das Beispiel erzeugt einen doppelten Rahmen mit einer Rahmendicke von 8 Pixel um das `<h1>` Element.

```
<h1 style="border: 8px double;">Kapitel 16 ...</h1>
```



**[farbe]** Nun kann man die Eigenschaft noch um einen Farbwert (siehe 4.1) ergänzen. (`#00f` ist der Hex-Wert für Blau).

```
<h1 style="border: 10px dotted #00f;">Deutsche Automarken</h1>
```



Man kann auch einzelne Rahmen bestimmen:

`border-top` für den oberen Rahmen. `border-bottom` für den unteren Rahmen. `border-left` für den linken und `border-right` für den rechten Rahmen.

```
<p style="border-bottom: 5px solid red;">Unser Jahrbuch</p>
```

## Abgerundete Ecken

## CSS

```
border-radius: [li-oben] [re-oben] [re-unten] [li-unten];
```



Der Wert wird in Pixel angegeben. Gibt man nur einen Wert an, dann werden alle vier Recken gleichmäßig abgerundet. Die Ecken werden im Uhrzeigersinn definiert.

```
<h2 style="border: 10px solid rgb(255,126,0);
border-radius: 25px 0px 25px 0px;
background-color: blue;
color: white;">... in einem fernem Land <br>
... vor gar nicht allzu langer Zeit!</h2>
```

# Lernhandout 4.3 Höhe und Breite

Referenzcode **HCL043**

Technologien CSS

Dateinamen L4\_3\_Breite.odt, L4\_3\_Breite.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... nutzen die CSS Eigenschaften height (Höhe) und width (Breite).	<b>I</b>	II
... haben ein Wissen über absolute Längenmaße.	<b>K</b>	I
... haben ein Wissen über relative Längenmaße.	<b>K</b>	I
... unterscheiden zwischen absoluten und relativen Längenmaßen.	<b>K</b>	II
... arbeiten mit den passenden Längenmaßen nach pragmatischen Gesichtspunkten.	<b>K</b>	III
... begrenzen Höhe und Breite durch minimal oder maximal Werte (min-height, max-height).	<b>I</b>	II
... steigern ihr Layout-Bewusstsein (z. B. Goldener Schnitt, Proportionenlehre).	<b>A</b>	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die absoluten Längenmaße sind abhängig von der tatsächlichen Bildschirmauflösung, d. H. von der DPI-Dichte (Dots per Inc). Deshalb ist  $cm = 37,8$  Pixel nur ein Richtwert.
- Um die relativen Längenmaße vorzuzeigen, muss man die Browser-Größen verändern. Meist reicht es das Browserfenster aus dem Vollbild zu nehmen um es an den Seiten zu resizen. Die Entwickler-Tools der Browser (z. B. Firefox) bieten auch eine Möglichkeit Bildschirmgrößen zu testen.
- em orientiert sich an der Breite des Buchstaben M.

Die CSS Eigenschaften `height` (Höhe) und `width` (Breite) bestimmen die Größenangaben eines HTML Elements.

## CSS



`height: [Wert]; width: [Wert];`

**[Wert]** Für die Größenangaben gibt es in CSS eine breite Palette an Möglichkeiten. Wir unterscheiden zwischen absoluten und relativen Längenmaßen.

**Absolute Längenmaße** sind fixe Größe und ändern sich in der Regel nicht!

<b>px</b>	Pixel	Bildpunkte des Screens (Bildschirm, Smartphone udgl.)
<b>cm</b>	Zentimeter	Entspricht ca. 37,8 Pixel
<b>mm</b>	Millimeter	Entspricht ca. 3,78 Pixel, bzw. 0,1 cm
<b>in</b>	Zoll	Ein Zoll sind 2,54 cm oder 96 Pixel
<b>pt</b>	Punkt	Werden für Schriftgrößen verwendet. 1 pt entspricht ca. 1,33 Pixel

```
<p style="background-color: blue; width: 4cm;">&nbsp;&nbsp;&nbsp;</p>
<p style="background-color: red; width: 400px;">&nbsp;&nbsp;&nbsp;</p>
```



**Relative Längenmaße** beziehen sich auf die Größe eines Elements (meist dem Elternelement). Wird die Größe verändert – z. B. Browserfenster verkleinern/vergrößern, dann verändert sich auch die Größe des Kind Elements – und zwar relativ.

<b>%</b>	Prozent %	z. B. <code>width: 50%;</code>
<b>em</b>	em	Vertikale Größe einer Schrift. Praktisch für Schriftarten.
<b>rem</b>	Wurzel-Em	Relativ zum Wurzelement (<html>).
<b>vw</b>	Viewport-Breite	entspricht dem 100. Teil des Anzeigebereichs.
<b>vh</b>	Viewport-Höhe	entspricht dem 100. Teil der Höhe des Anzeigebereichs. <code>height: 25vh;</code> hat ¼ der Höhe des Screens.

```
<div style="border: 5px solid; height: 50vh;">&nbsp;&nbsp;&nbsp;</div>
```

Das Beispiel oben zeigt das Container-Element `<div>` mit der halben Höhe des Browser-Fensters an. Verändert sich die Browsergröße (nicht jeder Browser wird in Vollbild gestartet), dann verändert sich auch die Höhe des `<div>` Elements.

Relative Größen haben das Problem, dass sie entweder zu groß oder zu klein sind wenn sich die Bezugsgröße verändert. Dafür gibt es die Eigenschaften:

## CSS



`max-height: [Wert]; max-width: [Wert];`

Sie beschreiben die maximale Breite oder Höhe. z. B. eine relative Breite von 50% endet bei 300 Pixel, wenn man `max-width: 300px;` definiert hat.

`min-width` und `min-height` geben jene Größen an, die auf jeden Fall dargestellt werden sollen. Dieses Minimum darf nicht unterschritten werden!

```
<div style="background-color: green;
width: 50%;
min-width: 200px;">&nbsp;&nbsp;&nbsp;</div>
```

# Übungsblatt 4.3 Farben, Rahmen und Größen

Referenzcode: **HCU043**

Technologien: HTML, CSS

Dateinamen: U4\_3\_Farben.odt, U4\_3\_Farben.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... reproduzieren eine graphische Darstellung laut Vorlage.	I	II
... verschachteln <code>&lt;span&gt;</code> Elemente zur Lösung der Aufgabe.	I	III
<b>Übung B</b>		
... erstellen eine Tabelle mit farblichen Hintergründen.	I	II
... finden RGB und HEX Werte selbstständig heraus.	I	II
<b>Übung C</b>		
... erstellen einfache HTML Buttons.	I	II
... steigern ihr Kreativbewusstsein durch freie Arbeit.	A	II
... verwenden Rahmen und Farben.	I	II
<b>Übung D</b>		
... erstellen eine Verkehrsampel ausschließlich mit HTML und CSS.	I	II
... verwenden relative Größenangaben.	I	II
... erzeugen Kreise durch Eckenabrundungen.	I	II
... reproduzieren eine graphische Darstellung laut Vorlage.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Übung A: Mit `&nbsp;` können Zwischenräume gestaltet werden!
- Übung C: Die Buttons können mit einem `<span>` Element oder einem `<a href>` erzeugt werden. Wenn man `<a href>` verwendet, kann man auch einen Hyperlink hinzufügen. Buttons werden später für Ankerpunkte und besonders für JavaScript benötigt.



**Übung A: Loveparade**

- Erstelle nur mit HTML und CSS (`style=""` Attribut) einen Loveparade-Schriftzug wie unten dargestellt.
- Bilddateien sind **nicht** erwünscht!
- Verwende nur ein einziges `<h1>` Element.
- Jeder Buchstabe hat eine eigene Hintergrundfarbe.
- TIPP: Verwende `<span>` Tags!**



**Übung B: Farbtabelle**

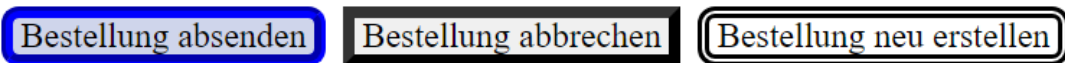
- Erstelle nur mit HTML und CSS eine Farbtabelle.
- Ermittle die RGB und HEX Werte der fehlenden Farben und trage Sie in die Tabelle ein!**  
(Türkis, Blaugrün, Himmelblau, Magenta)

Farbname	R	G	B	HEX
Gold	255	215	0	#FFD700
Gelbgrün	154	205	50	#9aCd32
Türkis				
Blaugrün				
Himmelblau				
Magenta				



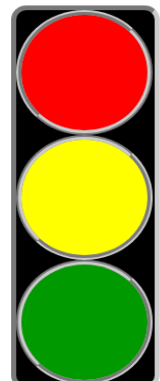
**Übung C: Einfache Buttons**

- Designe drei unterschiedliche Button!
- FREIES Design  
(alles ist erlaubt, solange es nur in HTML und CSS ist, keine Bilder)
- Die Buttons haben noch keine Funktion – noch nicht!**



**Übung D: Eine Ampel**

- Erstelle nur mit HTML und CSS eine Verkehrsampel.
- Die Größe der Ampel wird durch die Breite des Browsers bestimmt.  
(z. B. wird der Browser in der Breite größer, dann soll die Ampel auch größer werden – sie skaliert).
- TIPP: Verwende `border-radius` und relative Größenangaben!**



# Lernhandout 5.1 CSS im Head

Referenzcode: **HCL051**

Technologien: HTML, CSS

Dateinamen: L5\_1\_CSSHead.odt, L5\_1\_CSSHead.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verlagern die CSS Anweisungen in den <code>&lt;head&gt;</code> des HTML Codes.	I	II
... wissen, was ein Selektor ist.	I	I
... kennen Typselektoren um HTML Elemente anzusprechen.	I	I
... verwenden Typselektoren mit der korrekten Syntax.	I	II
... kennen ID-Selektoren um ein bestimmtes Element anzusprechen.	I	I
... verwenden ID-Selektoren mit der korrekten Syntax.	I	II
... kennen Klassenselektoren um mehrere gleiche Elemente anzusprechen.	I	I
... verwenden Klassenselektoren mit der korrekten Syntax.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die CSS Styleanweisungen im `<head>` haben innerhalb des gesamten Dokuments eine Gültigkeit. Alle anderen Inhalte (z. B. `<title>`, Metatags) im `<head>` bleiben selbstverständlich erhalten.
- **WICHTIG:** Die CSS-Selektoren sind Case-Sensitive! Das bedeutet, es wird zwischen Groß- und Kleinschreibung unterschieden.  
Ein `id="meinElement"` Attribut wird mit `#meinelement {}` nicht selektiert! Richtig ist: `#meinElement {}`
- In den folgenden Einheiten/Handouts werden die Beispiele nach selben Muster aufgebaut!  
Zuerst die CSS-Anweisungen im `<head>`  
eine Wellenlinie ~~~~~  
Dann die HTML Elemente im `<body>`  
Wo es der Platz auf der Seite erlaubt, wird darauf hingewiesen. Ansonsten erkennt man die CSS Anweisungen an der Syntax `# {}` oder `. {}` usw.

Bisher haben wir unsere CSS Anweisungen direkt in das HTML Element geschrieben. Praktischer ist es jedoch, die CSS Anweisungen innerhalb des `<head>` Tags zu schreiben. Dafür öffnet man mit `<style>` und schließt dann mit `</style>`. Für die Zuweisung der CSS Stile gibt es drei Möglichkeiten (drei Selektoren: Typ, ID und Klasse).

```
<head>
  <style>
    h1 {border-bottom: thin solid black;}
    #hauptinhalt {background-color:#CFD5EB; color:#00C;}
    .wichtig {color:red;}
  </style>
</head>
```

### 1. Typselektor: Ein HTML Element überschreiben

Man definiert für einen Tag ein neues Aussehen. Jedes mal wenn dieser Tag dann im HTML-Dokument verwendet wird, erhält er das Aussehen aus dem `<style>` im `<head>`.

CSS	TAG {CSS-Anweisungen;}
-----	------------------------

```
CSS im <head><style> ... </style></head>:
h2 {color:red;}
~~~~~
HTML im <body> ... </body>:
<h2>Empfehlungen</h2>
```

### 2. ID-Selektor: Eine ID festlegen für ein einziges Element

Eine ID (Identifikation) gilt nur für ein einziges HTML Element. Im gesamten Code darf die ID nur einmal vorkommen. Im CSS schreibt man eine Raute (#) vor dem Namen. Ins HTML Element schreibt man das Attribut `id=" "`

CSS	#ID {CSS-Anweisungen;}
-----	------------------------

```
CSS im <head><style> ... </style></head>:
#haupt {background-color:blue;}
~~~~~
HTML im <body> ... </body>:
<p id="haupt">Empfehlungen</p>
```

### 3. Klassenselektor: Eine Klasse definieren für mehrere Elemente

Mit einer Klasse kann man beliebig viele Elemente ansprechen. Eine Klasse kann auch auf unterschiedliche Tags angewandt werden. Im CSS schreibt man einen Punkt vor den Klassennamen. Ins HTML Element schreibt man das Attribut `class=" "`

CSS	.CLASS {CSS-Anweisungen;}
-----	---------------------------

```
CSS im <head><style> ... </style></head>:
.sr1 {color:red;}
~~~~~
HTML im <body> ... </body>:
<h1 class="sr1">Vorlagen</h1>
<span class="sr1">WICHTIG</span>
```

# Lernhandout 5.2 Schriftarten

Referenzcode **HCL052**

Technologien CSS

Dateinamen L5\_2\_Schriften.odt, L5\_2\_Schriften.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verstehen wie Browser auf Schriftarten zugreifen.	I	I
... wissen was Schriftfamilien und generische Schriftarten sind.	I	I
... verwenden die korrekte CSS Syntax um eine Schriftfamilie zu definieren.	I	II
... kennen die Kategorien von Schriften.	I	I
... steigern ihre Kreativität durch Kenntnisse der Typographie.	A	II
... binden eigene Schriftarten ein.	I	II
... beachten die Urheberrechte von Schriftarten.	K	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- Google Fonts - <https://fonts.google.com>

## Anmerkungen

- Wenn nun generische Schriftarten (z. B. monospace) angegeben werden, wählt der Browser selbstständig eine passende Font bzw. die vom Benutzer ausgewählte Standardschriftart.
- Urheberrechte sind ein wichtiges Thema und sollten mehrmals zur Sprache gebracht werden. Auch die CC (Common Creatives) sollte thematisiert werden.
- Moderne Browser brauchen im `@font-face` keine format-Anweisung.
- Die Google Fonts können auch via Fremdquelle eingebettet werden

Um eine Schriftart für ein Element zu bestimmen, verwenden wir `font-family`. Der Browser greift auf die installierten Schriftarten des Empfänger-System zu. Da es sein kann, dass der Endbenutzer nicht jede Schriftart installiert hat, geben wir mehrere Schriftarten als Wert mit. Zusätzlich noch eine Generische Schriftart (allgemeine Schriftfamilie). Die Schriftnamen werden durch einen Beistrich getrennt. Wie bei allen anderen CSS Anweisungen endet `font-family` ebenfalls mit einem Semikolon (;). Besteht eine Schriftart aus mehreren Wörtern, dann wird sie unter Anführungszeichen (") geschrieben.

## CSS

```
font-family: [Schriftname] ["Schrift 2"] [Generische-Schrift];
```



[Generische-Schrift] Folgende generische Schriftfamilien können ausgewählt werden:

**serif** eine Schriftart mit Serifen (z. B. Times New Roman, Palatino)  
**sans-serif** ohne Serifen (z. B. **Univers**, Calibri)  
**cursive** Schreibschriftarten (z. B. Mistral, Vivaldi)  
**fantasy** ungewöhnliche Schriften (z. B. Impact Label, Rosewood)  
**monospace** alle Zeichen haben die gleiche Breite (z. B. Courier)

```
h1 {font-family:Tahoma, Geneva, sans-serif;}
```

Will man nun eine eigene Schriftart verwenden, dann benutzen wir die CSS Anweisung `@font-face`. Innerhalb der `@font-face` Anweisung, wird der Name der Schriftfamilie, eine gültige URL zur Schriftart (wo ist diese abgespeichert?) und das Format der Schriftart definiert.

```
@font-face {font-family: 'neue-schrift';
            src: url('pfad/zu/neueschrift.ttf') format('truetype'); }
```

Der Name der Schriftfamilie muss nicht immer `'neue-schrift'` lauten. Es kann ein beliebiger Name gewählt werden – jedoch bitte kein Name einer schon bestehenden anderen Schriftart. Danach kann die Schriftart mit `font-family` zugewiesen werden!

```
<p style="font-family: neue-schrift, Arial, sans-serif">
... Symbole erleichtern das Verständnis ...</p>
```



Achte auf die Dateigrößen einer Schriftart. Schriftarten von mehr als 300 KiB verlangsamen die Ladegeschwindigkeit. Nach der erstmaligen Abholung wird die Schriftart im Browsercache gespeichert.



Schriftarten sind durch Urheberrechte geschützt. Ohne eine Genehmigung (erworben/gekauft, CC, gratis Schriftarten) kann die Nutzung im Internet teuer werden. Manche Schriftarten sind technisch geschützt und werden im Browser nicht angezeigt.



Google bietet eine Vielzahl von freien und offenen Schriftarten auf <https://fonts.google.com> an.

Diese Schriftarten lassen sich praktisch und einfach in eine Webseite einbinden!

# Lernhandout 5.3 Schriftgrößen

Referenzcode: **HCL053**

Technologien: CSS

Dateinamen: L5\_3\_Schriftgr.odt, L5\_3\_Schriftgr.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verändern die Schriftgrößen mit CSS.	I	II
... unterscheiden zwischen absoluten und relativen Schlüsselwörtern.	K	II
... verwenden auch die allgemeinen CSS-Größenangaben.	I	II
... entscheiden welche Angaben für Druck bzw. für Bildschirmdarstellung passend sind.	I	III
... verändern die Strichstärke eines Textes.	I	II
... verwenden CSS um einen Text in Kapitälchen darzustellen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Absolute Schriftgrößenangaben (z. B. pt, px) eignen sich für den Druck. Relative Schriftgrößenangaben (z. B. em, %) für die Bildschirmdarstellung.
- Der Browser entscheidet bei Schlüsselwörter selbstständig über die Darstellungsgröße. Das hat den Vorteil, weil man nie wirklich weiß auf welchem Screen das HTML Dokument dargestellt wird. So wird eine Schrift mit absoluten Angaben in Pixel (px) auf einem Screen mit hoher DPI sehr klein.
- Im Bereich `font-variant` wird als Beispieltext: **Was it the cat i saw** verwendet. Dieses schöne Palindrom ist aus Alice im Wunderland. Es ist zwar äußerst ästhetisch, hat aber wenige Großbuchstaben um die Kapitälchen besser zu illustrieren.

## CSS



**font-size:** [Wert]

Mit **font-size** bestimmt man die Darstellungsgröße der Schrift. Als Werte können alle Größe (siehe 4.3 Höhe und Breite) angewendet werden. Zusätzlich gibt es noch Schlüsselwörter:

**Schlüsselwörter mit absoluten Angaben**

<b>xx-small</b> .....	<i>winzig</i>	<b>large</b> .....	<i>groß</i>
<b>x-small</b> .....	<i>sehr klein</i>	<b>x-large</b> .....	<i>sehr groß</i>
<b>small</b> .....	<i>klein</i>	<b>xx-large</b> .....	<i>riesig</i>
<b>medium</b> .....	<i>mittel</i>		

**Schlüsselwörter mit relativen Angaben**

<b>smaller</b> .....	<i>Kleiner als das Elternelement</i>
<b>larger</b> .....	<i>Größer als das Elternelement</i>
<b>inherit</b> .....	<i>Die gleiche Größe wie im Elternelement</i>



Über die Schlüsselwörter entscheidet der Browser selbstständig in welcher Größe die Schrift dargestellt werden soll.

```
h2 {font-size: 45px;}
#haupttext {font-size: medium;}
.tipp {font-size: larger;}
span {font-size: 1.2em;}
```

## CSS



**font-weight:** [Wert];

Mit **font-weight** wird die Strichstärke (Dicke) des Textes festgelegt. Folgende Werte sind möglich:

<b>lighter</b> .....	<i>dünnere als im Elternelement</i>
<b>normal</b> .....	<i>normale Strichstärke</i>
<b>bold</b> .....	<i>fett</i>
<b>bolder</b> .....	<i>fetter als im Elternelement</i>
<b>100, 200, 300 ... 900</b> .....	<i>extra-dünn (100) bis extrafett (900)</i>
<b>inherit</b> .....	<i>Strichstärke des Elternelements</i>

```
.fetter {font-weight: bolder;}
#impressum {font-weight: 400;}
```

## CSS



**font-variant:** [Wert];

Mit **font-variant: small-caps**; kann man einen Text mit Kapitälchen darstellen. Kapitälchen sind Großbuchstaben in der Höhe von Kleinbuchstaben.

*Beispiel ohne Kapitälchen:* Was it the cat i saw?

**Beispiel mit Kapitälchen:** WAS IT THE CAT I SAW?

```
<p>Was it the cat i saw?</p>
<p style="font-variant: small-caps;">Was it the cat i saw?</p>
```

# Lernhandout 5.4 Text Dekoration

Referenzcode **HCL054**

Technologien CSS

Dateinamen L5\_4\_Text.odt, L5\_4\_Text.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... über-, durch- oder unterstreichen Text mit CSS.	<b>I</b>	<b>II</b>
... kennen die Attribute der CSS Eigenschaft text-decoration.	<b>I</b>	<b>I</b>
... verändern das Aussehen von Hyperlinks.	<b>I</b>	<b>II</b>
... achten bei Unterstreichungen auf die Unterlänge von Buchstaben.	<b>K</b>	<b>II</b>
... definieren Textschatten mit CSS.	<b>I</b>	<b>II</b>
... entwickeln ein Designverständnis für die Verwendung von Textdekorationen. (z. B. sparsamer Umgang mit Schatten).	<b>KA</b>	<b>II</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Sehr viele Browser haben das Problem mit der Unterlänge und Unterstreichungen behoben indem einfach Buchstaben mit Unterlänge nicht unterstrichen dargestellt werden. Dennoch sollten die Schüler\_innen auf das Thema "Unterstreichungen" sensibilisiert werden.
- Der Text-Shadow ist vielleicht zu Anfang etwas abstrakt. Es gibt aber zahlreiche Text-Shadow-Generatoren im Internet die einen CSS-Code liefern.

Mit der `text-decoration` Eigenschaft kann man einen Text über-, unter- oder durchstreichen. Zusätzlich können Stiel und Farbe der Linien definiert werden! Für Color-Angaben gelten die allgemeinen Regeln (siehe 4.1 `style`, `span`, `color`).

## CSS



```
text-decoration: [line] [style] [color];
```

## [line]

`underline`.....unterstrichen

`overline`.....überstrichen

`line-through`.....durchgestrichen

`none`.....keine Textdekoration

## [style]

`solid`.....durchgezogene Linie

`double`.....doppelte Linie

`dotted`.....gepunktete Linie

`dashed`.....gestrichelte Linie

`wavy`.....Wellenlinie

```
.falsch {text-decoration: underline wavy red;}
```

```
<h1 style="text-decoration: underline;">Fehlerquellen</h1>
<p>Standard wird oft mit einem t geschrieben
<span class="falsch">(Standart)</span></p>
```

Hyperlinks (`a href`) werden vom Browser meist blau mit einer durchgezogenen Linie dargestellt. Will man die Unterstreichung von den Links entfernen, dann kann das mit `text-decoration: none`; umgesetzt werden.

```
a {text-decoration: none;}
```



**Design Tipp:** Achte darauf, ob die Unterstreichung die Unterlänge des Textes nicht übermalt. Buchstaben mit Unterlänge sind: g, j, y usw.

## Textschatten

Mit der `text-shadow` Eigenschaft fügt man dem Text einen Schatten hinzu.

## CSS



```
text-shadow: [H] [V] [blur] [color];
```

[H] Die Horizontale Position des Schattens.  
(Negative Werte sind erlaubt)

[V] Die vertikale Position des Schattens. (Negative Werte sind erlaubt)

[blur] Radius der Unschärfe (Weichzeicheneffekt).

[color] Farbangabe

```
h1 {text-shadow: 5px 5px 10px blue;}
h2 {color: white; text-shadow: 2px 2px 4px #000000;}
```

```
<h1>Moderne Medizin</h1>
<h2>Die chinesische Medizin im Überblick</h2>
```

# Übungsblatt 5.4 Schriften und Textdekoration

Referenzcode: **HCU054**

Technologien: HTML, CSS

Dateinamen: U5\_4\_Schrift.odt, U5\_4\_Schrift.pdf, Austria.txt

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... reorganisieren die CSS Anweisungen in den <head>.	<b>I</b>	II
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... ändern die Schriftart von Buttons.	<b>I</b>	II
... suchen im Internet nach freien Schriftarten und beachten dabei das Urheberrecht.	<b>IK</b>	III
... binden Schriftarten mit CSS ein.	<b>I</b>	II
<b>Übung C</b>		
... recherchieren die Nutzung von Google-Fonts und stellen dabei Überlegungen zur praktischen Umsetzung an.	<b>I</b>	III
... dokumentieren das Rechercheergebnis.	<b>I</b>	II
<b>Übung D</b>		
... wandeln einen Plain-Text in ein HTML Dokument.	<b>I</b>	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die nochmalige Bearbeitung von Buttons bzw. Schaltflächen sollen zu einem gesteigerten kreativen Selbstbewusstsein führen. Buttons, Schaltflächen und Navigation sind essentielle Teile einer jeden Webanwendung - wenn man schon frühzeitig beginnt einen eigenen Stil zu entwickeln, so hat man auch die besten Chancen sich von anderen Webseiten abzuheben bzw. abzugrenzen! In der Praxis spielt dann wieder (contraire) die Usability eine Rolle!



### Übung A: Einfache Buttons im Head

- Öffne deine Arbeit/Lösung von **4.3 C: Einfache Buttons**
- Schreibe die Arbeit so um, dass alle CSS Anweisungen im Head sind!
- Solltest du 4.3 C noch nicht gelöst haben, dann erstelle drei Buttons mit CSS Anweisungen im Head.**



### Übung B: Buttons im Head

- Öffne deine Lösung von **5.4 A: Einfach Buttons im Head.** (siehe oben)
- Erstelle einen weiteren Button – dieser soll eine außergewöhnliche Schriftart haben. (also keine Standard Schriftart).
  - o Suche dafür im Internet nach einer **freien** Schriftart die du downloaden kannst und die dir gefällt. Beachte die Urheberrechte!
  - o Binde die neue Schriftart mit CSS ein. Tipp: `@font-face`
  - o **Der neue Button soll als Klasse verfügbar sein und vier mal dargestellt werden.** Tipp: `class=" "`



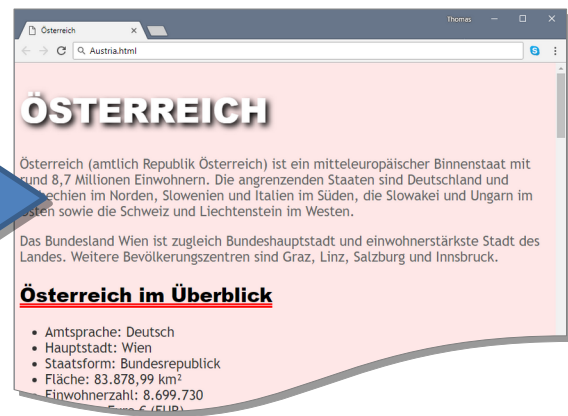
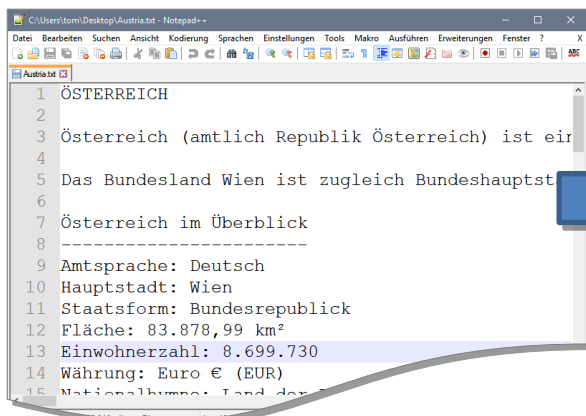
### Übung C: Google Fonts

- Finde heraus, wie man die Schriftarten von font.google.com nutzen kann!
- Dokumentiere es schriftlich. (z. B. mit Word, PowerPoint, HTML ...)**



### Übung D: Österreich

- Öffne die Austria.txt
- Kopiere den Inhalt der **Austria.txt** in den `<body>` eines neuen HTML Dokuments.
- Formatiere den Text bzw. das Dokument mit CSS (Freie Übung).
- Speichere die fertige Arbeit als `Austria.html`**



# Lernhandout 6.1 Ausrichtung

Referenzcode **HCL061**

Technologien CSS

Dateinamen L6\_1\_Ausrichtung.odt, L6\_1\_Ausrichtung.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... bestimmen die Ausrichtung von Textteilen.	I	II
... verwenden die text-align Eigenschaft für Links- und Rechtsbündig, Zentriert und Blocksatz.	I	II
... erzeugen Erstzeileneinzüge mit text-indent.	I	II
... definieren Spalten für Bereiche mit viel Text.	I	II
... nutzen weitere CSS Eigenschaften in Verbindung mit Spalten. (column-gap, column-rule, column-rule).	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Es könnte sein, dass die CSS-Eigenschaften column-gap und column-rule noch nicht in alle Browser implementiert wurde. Zur Erscheinung von diesem Script hier ist es aber eher unwahrscheinlich - kommt natürlich auf die Browser-Politik an, auf die wir nur mäßig Einfluss haben!
- Columns sind definitiv nützlich - aber nur für viel Text. Es soll nicht die hohe Kunst zwei oder drei Div's nebeneinander zu platzieren ersetzen. Insbesondere noch, wenn die Media-Query's eine Rolle spielen werden! Aber lassen wir den Schüler\_innen den glauben, sie hätten noch alles unter Kontrolle!

Natürlich muss der Text nicht immer Linksbündig sein. Mit der `text-align` Eigenschaft (CSS) kann der Text (ebenso andere inline-Element, z.B. `<img>`) auch Zentriert, Rechtsbündig oder als Blocksatz dargestellt werden.

-  Linksbündig `left`
-  Zentriert `center`
-  Rechtsbündig `right`
-  Blocksatz `justify`

**CSS**

`text-align: [Wert];`

Für **[Wert]** können folgende Werte eingesetzt werden:

- left** Linksbündig
- center** Zentriert
- right** Rechtsbündig
- justify** Blocksatz (mit `start` und `end` kann man noch das Verhalten der letzten Zeile bestimmen).

```
<h1 style="text-align:center;
border-bottom:3px solid;">Mitteilung</h1>
<p style="text-align:right; font-size:smaller">by John Doe</p>
```

Einen **Erstzeileneinzug** erzeugt man mit `text-indent`. Als Wert `[value]` sind positive oder negative absolute Längenangaben, oder prozentual % relativ zur Breite möglich.

**CSS**

`text-indent: [Wert];`

```
<p style="text-indent: 5%;" >Auf der Registerkarte ...</p>
```

**Spalten**

Besonders viel Text lässt sich mit Spalten übersichtlicher gestalten – besonders wenn es zu einer Darstellung der Webseite auf breiten Monitoren kommt.

**CSS**

`columns: [count] [width];`

- [count]** Wie viele Spalten sollen angezeigt werden?  
Eine positive Zahl oder `auto`. z. B. 2 für 2 Spalten.
- [width]** Eine Längenangabe für die Mindestspaltenbreite.  
z. B. 6em, 500px,  
Zwischen den zwei Werten wird ein Leerzeichen gesetzt.

```
.absatz {columns:3 7em; text-align:justify;}
```



Weitere CSS Eigenschaften in Verbindung mit Spalten (`columns`):

- `column-gap` bestimmt den Abstand zwischen den Spalten.
- `column-rule` um eine Linie zwischen den Spalten zu bestimmen.

**CSS**

`column-rule: [width] [style] [color];`

- [width]** Liniestärke, **[style]** alle border-style-Werte,  
**[color]** alle Farben.

```
.absatz {columns: 2; column-gap: 1.8em; column-rule: 5px dotted red;}
```

# Lernhandout 6.2 Abstände

Referenzcode **HCL062**

Technologien HTML, CSS

Dateinamen L6\_2\_Abstand.otd, L6\_2\_Abstand.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... kennen und verstehen das "klassische" Boxmodell.	I	I
... definieren den Außenabstand einer Box mit der CSS Eigenschaft margin.	I	II
... definieren den Innenabstand einer Box mit der CSS Eigenschaft padding.	I	II
... setzen die Abstände an einer bestimmten Länge oder Breite (top, right, bottom, left).	I	II
... erkennen die Auswirkungen eines Innenabstands (padding) auf die Größe der Box.	I	II
... entfernen die Abstände des <body> Elements.	I	II

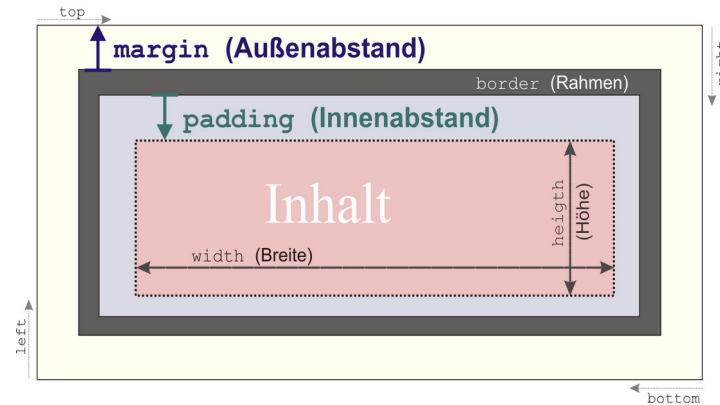
**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Nehmen wir an, wir definieren mit CSS eine Box:  
`#meineBox {width: 300px, height: 300px, border: 1px solid red;}`  
Die Box ist also 300 Pixel hoch und 300 Pixel breit.  
Nun bekommt die Box noch ein Padding von 50px dazu.  
Sie wird nun in der Größe von 400 Pixel Breite x Höhe dargestellt.  
Das Problem lässt sich mit der CSS Eigenschaft: `box-sizing` lösen.  
Mit `box-sizing: border-box;` bekommt die Box wieder die ursprüngliche Größe von 300 x 300 Pixel.
- Das `<body>` Element hat in den meisten Browsern defaultmäßig einen kleinen Abstand. Mit `body {margin: 0;}` wird dieser aufgehoben!

Im "klassischen" Boxmodell wird eine Box (rechteckiges HTML Element z. B. `<div>` oder `<h1>`) durch seine Höhe und Breite, durch den Innen- und Außenabstand sowie durch den Rahmen bestimmt. Mit der CSS Eigenschaft `margin` bestimmt man den Außenabstand – mit `padding` den Innenabstand. Beide beziehen sich auf den Rahmen (auch wenn kein Rahmen definiert wurde!)



## CSS



```
padding: [top] [right] [bottom] [right];
```

[top] Innenabstand oben. (auch `padding-top`).

[right] Innenabstand rechts (auch `padding-right`).

[bottom] Innenabstand unten (auch `padding-bottom`).

[left] Innenabstand links (auch `padding-left`).

Als Werte sind alle numerischen Längenmaße erlaubt (z. B. cm, px, em). Negative Werte sind nicht erlaubt. Gibt man nur einen Wert an, dann gilt dieser für alle vier Seiten.

```
h1 {padding-left: 3em;}
td {padding: 10px 5px 5px 10px;}
```

Der Außenabstand (`margin`) funktioniert gleich wie `padding`. Man kann den Außenabstand auch einzeln ansprechen (`margin-top`, `margin-right` usw.). Zusätzlich erlaubt die `margin` Eigenschaft noch den Wert `auto` und negative Längenmaße.

## CSS

```
margin: [top] [right] [bottom] [right];
```

```
.wichtig {margin: 10px 4px 10px 5px; border: 2px solid;}
h2 {margin: -10px;}
```



Um ein Box-Element zentriert dazustellen muss man den `margin` Wert auf `auto` setzen und eine Breite angeben!

```
.dieBox {margin:auto; width:600px;
background:#FC0; height:3em; border:2px solid; }
~~~~~
<h1 class="dieBox">Wir verwenden Cookies!</h1>
<div class="dieBox" style="text-align: right;">Ist das Okey?</div>
```

# Übungsblatt 6.2 Ausrichtung und Abstände

Referenzcode: **HCU062**

Technologien: HTML, CSS

Dateinamen: U6\_2\_Abstand.odt, U6\_2\_Abstand.pdf, Wordelemente.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... führen Absatz- und Zeichenformatierungen durch.	<b>I</b>	II
... zentrieren das Tabellenelement über die Bildschirmbreite (margin: auto;).	<b>I</b>	II
... definieren Abstände, Ausrichtungen und Rahmen.	<b>I</b>	II
... arbeiten strukturiert nach Vorgaben.	<b>K</b>	II
<b>Übung B</b>		
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II
<b>Übung C</b>		
... arbeiten und orientieren sich in einem fremden HTML Dokument.	<b>KI</b>	III
... führen Absatz- und Zeichenformatierungen durch.	<b>I</b>	II
... definieren Abstände und Ausrichtungen.	<b>I</b>	II
... arbeiten strukturiert nach Vorgaben.	<b>K</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die Startdatei Wordelemente.html muss zum Download bereitgestellt werden! Sie kann über die Suche von [www.css4.at](http://www.css4.at) gefunden werden!



### Übung A: Ländervergleich

- Öffne deine HTML-Arbeit **3.1 B: Ländervergleich** (Tabellenübung).
- Die Ländernamen sollen hervorgehoben sein (Font-weight, Font-size).
- Die Zellen zu den Ländern sollen zentriert sein – die Beschriftungszellen (Land, Amtssprache, Hauptstadt usw.) bleiben linksbündig.
- Zwischen den Zeilen soll ein "Rahmenstrich unten" sein.
- Die Beschriftungszellen sollen einen grauen Hintergrund haben.
- Ändere die Schriftart und Schriftgröße der Tabelle (z. B. Arial, Verdana).
- Füge einen Innenabstand für Zellen hinzu.
- Zentriere die gesamte Tabelle über die Browserbreite! Tipp: `margin`
- Die Tabelle soll einen dicken groovie Rahmen haben.**

Land	Schweiz	Österreich	Deutschland
Amtssprache	Deutsch	Deutsch	Deutsch
Hauptstadt	Bern	Wien	Berlin
Fläche	41 285 km <sup>2</sup>	83 878 km <sup>2</sup>	357 385 km <sup>2</sup>
Einwohner	8,4 Mio	8,7 Mio	82,5 Mio
Staatsform	föderale Republik	Bundesrepublik	Bundesrepublik
Währung	Schweizer Franken	Euro	Euro
Internet-TLD	.ch, .swiss	.at	
Telefon-Vorwahl	+41		



### Übung B: Stundenplan

- Öffne deinen Stundenplan (siehe **3.1 A: Stundenplan**).
- Verschönere ihn mit CSS – freie Kreativübung.**



### Übung C: Text mit Spalten

- Öffne die Datei `wordelemente.html`
- Lege um den gesamten Text ein `<div>` Element mit der `id="Hauptdiv"`.
- Das Hauptdiv hat eine Breite von 75%, schwarzen Hintergrund und weiße Schrift. Einen Innenabstand (`padding`) von 2em.
- !!! Das Hauptdiv soll ganz Rechts am Browserrand ausgerichtet werden. !!!**
- Der Seitenrand vom `body` soll entfernt werden.
- Alle Absätze sind linksbündig ausgerichtet und haben einen Erstzeileneinzug von 3em.
- Der zweite Absatz (Lorem ipsum dolor sit ...) soll mit drei Spalten und als Blocksatz dargestellt werden.**

# Lernhandout 6.3 Best Practice

Referenzcode **HCL063**

Technologien HTML, CSS

Dateinamen L6\_3\_BePr.odt, L6\_3\_BePr.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... binden externe Stylesheets in ein HTML Dokument ein.	I	II
... organisieren Dateien um Ordnung zu schaffen.	S	II
... assoziieren ein einfaches Verständnis für Corporate Design (viele HTML Seiten mit demselben Aussehen).	K	III
... unterscheiden CSS Stylesheets nach Ausgabemedien (screen, print).	K	I
... definieren für mehrere Selektoren die gleichen CSS Eigenschaften.	I	II
... weisen einem HTML Element mehrere CSS Klassen zu.	I	II
... kennen den CSS Universalselektor.	I	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Wenn das media-Attribut `print` bekommt, dann wird dieses bei einem Druckauftrag angewendet. Hier sind dann absolute Größenangaben (z. B. in cm) äußerst nützlich.
- CSS Kommentare werden mit `/* Kommentar */` definiert.
- Der Universalselektor überschreibt alle CSS Anweisungen. Äußerst praktisch in Verbindung mit `font-family` um einem Dokument eine Schriftart zuzuweisen.

Bisher haben wir unsere CSS Anweisungen entweder im `<head>` oder im HTML Element selbst geschrieben. Was aber, wenn man nicht nur eine Webseite hat, sondern viele? Natürlich will man, dass jede Seite das gleiche Design erhält. Dafür kann man CSS-Dateien (externe Stylesheets) einbinden. Externe Stylesheets haben die Endung `.css` und werden mit dem `<link>` Element im `<head>` eingebunden.

## HTML



```
<link rel="..." href="..." media="...">
```

`rel="..."` Ist der Beziehungstyp. Wir verwenden `stylesheet` – das sagt dem Browser, dass es sich um CSS Eigenschaften handelt.

`href="..."` die Adresse, wo die `.css` Datei zu finden ist. (relativ oder absolut)  
`href="https://www.css4.at/hauptdesign.css"`  
`href=" ../Design/Style.css"`

`media="..."` Ausgabemedium (z.B. `screen` = Bildschirm, `print` = Druck)



In der externen `.css` Datei, schreibt man gleich wie wir es von der `<head>` Lösung kennen. `<style> ... </style>` ist nicht nötig. Nötig ist aber, eine Regel für unseren Zeichensatz (UTF-8). Eine Regel wird mit dem At-Zeichen (@) eingeleitet.

```
<!doctype html>
<html lang="de">
<head>
  <link rel="stylesheet" href="stylesheet.css" media="screen">
  ... </head>
```

```
~~~~~ In der externen CSS Datei ~~~~~
@charset "UTF-8";
/* CSS Document */
body {margin: 0px;}
```

Man kann auch mehreren Selektoren die gleichen Eigenschaften zuweisen. Diese werden dann durch einen Beistrich getrennt. In folgendem Beispiel wird der Klasse `.tipp` und der Klasse `.hinweis` und dem `<p>` Tag eine blaue Schriftart zugewiesen.

```
.tipp, .hinweis, p {color: blue;}
.wichtig {background: #CCC;}
.fett {font-weight: bolder;}

~~~~~
<h1 class="tipp">Denkspiele</h1>
<h2 class="hinweis">Online oder Offline</h2>
<p>Es gibt so viele Möglichkeiten</p>
```



Ein HTML Element kann mehrere Klassen haben. Hier im Beispiel hat das `<h1>` Element eine Klassenzugehörigkeit zu `.wichtig` und zu `.fett`.

```
<h1 id="haupt" class="wichtig fett">Dokumente</h1>
```



Um alle Elemente anzusprechen, kann man den **Universalselektor** verwenden: Der Universalselektor wird mit einem Stern eingeleitet:  
 z. B.: `* {color: green; font-family: monospace;}`

# Lernhandout 7.1 Hintergrund

Referenzcode **HCL071**

Technologien CSS

Dateinamen L7\_1\_Hintergrund.odt, L7\_1\_Hintergrund.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... nutzen CSS um in ein HTML Element ein Hintergrundbild einzubinden.	I	II
... verstehen die Idee von Ebenen (z. B. Photoshop, zwei Bilder übereinander).	I	I
... skalieren Hintergrundbilder.	I	II
... kacheln (wiederholen) Hintergrundbilder.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Für ein "Vorzeigen-Nachmachen" müssen Bilder bereitgestellt werden.
- Wenn man zwei Hintergrundbilder übereinander legen möchte, dann sollte man nur Bildformate mit transparentem Hintergrund (wie z. B. PNG) verwenden. Bei zwei JPG Graphiken wird das untere überdeckt.
- Mit  
`#div {background-image: url("bild1.png"), url("bild2.png");`  
lassen sich zwei Bilder übereinander legen!  
**bild1.png** ist im Vordergrund und **bild2.png** im Hintergrund!
- Zu beachten ist, dass z. B. ein `<div>` auch Größenangaben benötigt, damit das Hintergrundbild dargestellt wird. z. B.: `width: 300px, height: 300px`

Wir können schon die Hintergrundfarbe eines Elements bestimmen. Jetzt wollen wir Bilder als Hintergründe verwenden. Dafür gibt es die **background** Eigenschaft.

### Hintergrundgrafik einbinden

#### CSS



**background-image:** `url("[pfad]");`

[pfad] Es wird der Pfad (absolut oder relativ) zum Bild angegeben. Bei einer externen .css Datei ist der Speicherort der .css Datei ausschlaggebend.

```
body {background-image: url("../pix/hintergrund.jpg");}
```

### Hintergrundbild skalieren (Größe verändern)

#### CSS



**background-size:** `[value];`

[value] Längenangaben wie z. B. px, % sind möglich. Dabei wird zuerst die Breite und dann die Höhe definiert.

**background-size:** `200px 400px;`

Wird nur ein Wert angegeben, dann wird die Höhe unter Beibehaltung des Seitenverhältnis skaliert.

Folgende Schlüsselwörter sind möglich:

**auto** Originalgröße des Bildes, keine Skalierung

**contain** Beibehaltung des Seitenverhältnis, die größere Seite wird angepasst.

**cover** Beibehaltung des Seitenverhältnis, die kleinere Seite wird angepasst. Damit wird der Anzeigebereich vollständig gefüllt.

```
#haupt {background-size: cover;}
.bilder {background-size: 300px;}
```

### Hintergrund wiederholen

#### CSS



**background-repeat:** `[value];`

[value] Wenn das Bild kleiner als das Element ist, kann man es wiederholen (kacheln). Dabei wird der gesamte zur Verfügung stehende Platz ausgefüllt. Folgende Werte sind möglich:

**repeat** Wiederholung des Hintergrundbildes

**repeat-x** Das Bild wird horizontal wiederholt.

**repeat-y** Das Bild wird vertikal wiederholt

**no-repeat** keine Wiederholung

**space** Das Hintergrundbild wird wiederholt ohne dass ein Bild beschnitten wird.

**round** Genauso wie **space**, hier wird aber das Bild skaliert.

```
#haupt {background-repeat: repeat-x;}
.bilder {background-repeat: space;}
```

# Lernhandout 7.2 Hintergrund

Referenzcode **HCL072**

Technologien CSS

Dateinamen L7\_2\_Hintergrund.odt, L7\_2\_Hintergrund.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... positionieren Hintergrundbilder in einem HTML Element.	I	II
... verändern den Ausgangspunkt für den Hintergrund.	I	II
... befestigen Hintergrundbilder (verhalten beim Scrolling).	I	II
... definieren Hintergrundeigenschaften für bestimmte Bereiche eines HTML Elements.	I	II
... finden im Internet lizenzfreie Hintergrundbilder.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- Background-Generator: <http://bg.siteorigin.com/>

## Anmerkungen

- Der Background-Generator <http://bg.siteorigin.com/> liefert schöne Hintergrundbilder zum Kacheln. Gekachelte Hintergrundbilder haben den Vorteil eines geringen Speicherbedarfs.

### Position der Hintergrundgrafik

#### CSS



**background-position:** [value-x] [value-y];

**[value-x]** Bestimmt die Position entlang der x-Richtung (horizontal). Erlaubt sind negative und positive Längenmaße und die Werte: **left center right**.

**[value-y]** Bestimmt die Position entlang der y-Richtung (vertikal). Erlaubt sind negative und positive Längenmaße und die Werte: **top center bottom**.

Der Ausgangswert ist 0 0 – Das ist in der linken oberen Ecke des Elements.

```
#haupt {background-position: 50px 100px;}
```



Mit **background-origin** wird ein Ausgangspunkt festgelegt.  
z. B: **background-origin: border-box;** richtet den Hintergrund auf die Außenkante des Rahmens.

### Hintergrundbild befestigen

#### CSS



**background-attachment:** [value];

Damit definiert man das Verhalten, wenn ein Element oder sein Inhalt bewegt wird. Folgende Werte [value] stehen zur Verfügung:

<b>fixed</b>	<b>Bewegung des Elements:</b>	ein anderer Bereich des Hintergrund wird sichtbar.
	<b>Bewegung des Inhalts:</b>	der Hintergrund bleibt unverändert.
<b>scroll</b>	<b>Bewegung des Elements:</b>	Hintergrund bleibt unverändert und scrollt mit dem Element mit.
	<b>Bewegung des Inhalts:</b>	der Hintergrund bleibt unverändert.
<b>local</b>	<b>Bewegung des Elements:</b>	Hintergrund bleibt unverändert.
	<b>Bewegung des Inhalts:</b>	der Hintergrund bewegt sich mit.

```
body {background-attachment: fixed;}
```

### Hintergrundeigenschaften für bestimmte Bereiche

#### CSS



**background-clip:** [value];

Die Hintergrundeigenschaften (z. B. Bild oder Farbe) werden hier für einen bestimmten Bereich festgelegt. Dafür gibt es folgende Werte [value] :

<b>content-box</b>	Hintergrund wird nur im Inhalt angezeigt.
<b>padding-box</b>	Hintergrund im Inhalt und im Innenabstand
<b>border-box</b>	Hintergrund im Inhalt, Innenabstand und im Rahmen.

Der Ausgangswert ist 0 0 – Das ist in der linken oberen Ecke des Elements.

```
#haupt {background-clip: content-box;}
```

# Lernhandout 7.3 Farbverlauf

Referenzcode **HCL073**

Technologien CSS

Dateinamen L7\_3\_Farbverlauf.odt, L7\_3\_Farbverlauf.pdf,  
Beispieluebungen.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... wissen was eine CSS Funktion ist und kennen die Syntax.	I	I
... erstellen einfache Farbverläufe (mit zwei Farben) für HTML Elemente.	I	II
... erstellen komplexe Farbverläufe (mit mehrere Farben).	I	II
... verändern die Erscheinung des <body> Tag.	I	II
... erstellen nur mit einem CSS Farbverlauf eine Österreich Flagge und verwenden dafür Längenangaben.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die Datei Beispieluebungen.html unterstützt die Visualisierung dieser Einheit.
- `linear-gradient()` ist eine CSS Funktion. Zu beachten ist, dass kein Leerzeichen zwischen `linear-gradient` und der runden Klammer gesetzt wird!
- Interessant ist auch ein Farbverlauf im `<body>` Tag. Das `<body>` Element erstreckt sich nicht über den ganzen Viewport sondern nur über die tatsächliche Höhe des Inhaltes. Deswegen wiederholt sich ein Farbverlauf unschön, z. B.:

```
body {background-image:
```

```
    linear-gradient(to bottom left, blue, white);}
```

Dem kann man mit `background-attachment:fixed;` entgegenwirken. Damit also ein Farbverlauf über den gesamten Viewport gezeichnet wird, definiert man:

```
body {background-image:
```

```
    linear-gradient(to bottom left, blue, white);
```

```
    background-attachment:fixed;}
```

Um einen Farbverlauf im Hintergrund eines Elements zu erstellen, kann man eine Graphik bzw. Bilddatei erstellen oder man nutzt CSS mit der Funktion `linear-gradient()`. Diese Funktion lässt sich z. B. mit der CSS Eigenschaft `background-image` darstellen.

## CSS



`background-image:`

`linear-gradient([to value], [farbeVon], [farbeBis]);`

**[to value]** Gibt die Richtung an. Dafür kann man ein Winkelmaß (z. B. `75deg`) angeben oder in Verbindung mit dem Schlüsselwort `to` die Werte `top`, `right`, `bottom`, `left`. Der Standardwert (ohne Angabe) ist `to bottom = 180deg = 200grad`.

**[farbeVon]** Der Color-Start-Wert kann eine beliebige Farbangabe sein. Es ist auch der Wert `transparent` möglich. Zusätzlich kann noch ein Längenmaß mitgegeben werden.

**[farbeBis]** Für den Color-End-Wert gelten die gleichen Bedingungen wie für den Color-Start.

Die Wertangaben werden durch Beistriche getrennt.

```
#dahinter {background-image:
            linear-gradient(to top right, red, transparent);}
```



Natürlich kann man mehrere Farben zuweisen.  
Im Beispiel: `75°` rot bis orange bis gelb bis `#FFC` bis Weiß.



```
#hinten {background-image:
          linear-gradient(75deg, red, orange, yellow, #FFC, white);
          width:21cm; height:2cm; margin:auto; border: 4px groove;}
```

```
<div id="hinten">&nbsp;</div>
```



Mit CSS eine Österreich-Flagge erstellen.  
Dafür benötigen wir Längenangaben.

```
#austria {background-image:
           linear-gradient(red 100px, white 100px, white 200px, red 100px);
           width:400px; height:300px; margin:auto; border: 4px groove;}
```

```
.schrift {margin-top:125px; font-size:50px; text-align:center;}
```

```
<div id="austria"><p class="schrift">Österreich</p></div>
```

# Übungsblatt 7.3 Hintergrund

Referenzcode **HCU073**

Technologien HTML, CSS

Dateinamen U7\_3\_Hintergrund.odt, U7\_3\_Hintergrund.pdf,  
carnary\_wharf.html, london.png

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... arbeiten strukturiert nach Vorgaben und mit einem fremden HTML Dokument sowie mit bereitgestellten Bildern.	<b>I</b>	<b>II</b>
... binden Hintergrundbilder ein und weisen diesen div. CSS Eigenschaften zu.	<b>I</b>	<b>II</b>
... erstellen CSS Farbverläufe.	<b>I</b>	<b>II</b>
... verspüren ein "London-Feeling".	<b>A</b>	<b>III</b>
<b>Übung B</b>		
... definieren einen CSS Farbverlauf für einen Schriftzug.	<b>I</b>	<b>II</b>
... orientieren sich an dem Screenshot des Übungsblattes.	<b>K</b>	<b>I</b>
... recherchieren selbstständig nach CSS Anweisungen und deren Verwendung.	<b>SI</b>	<b>III</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

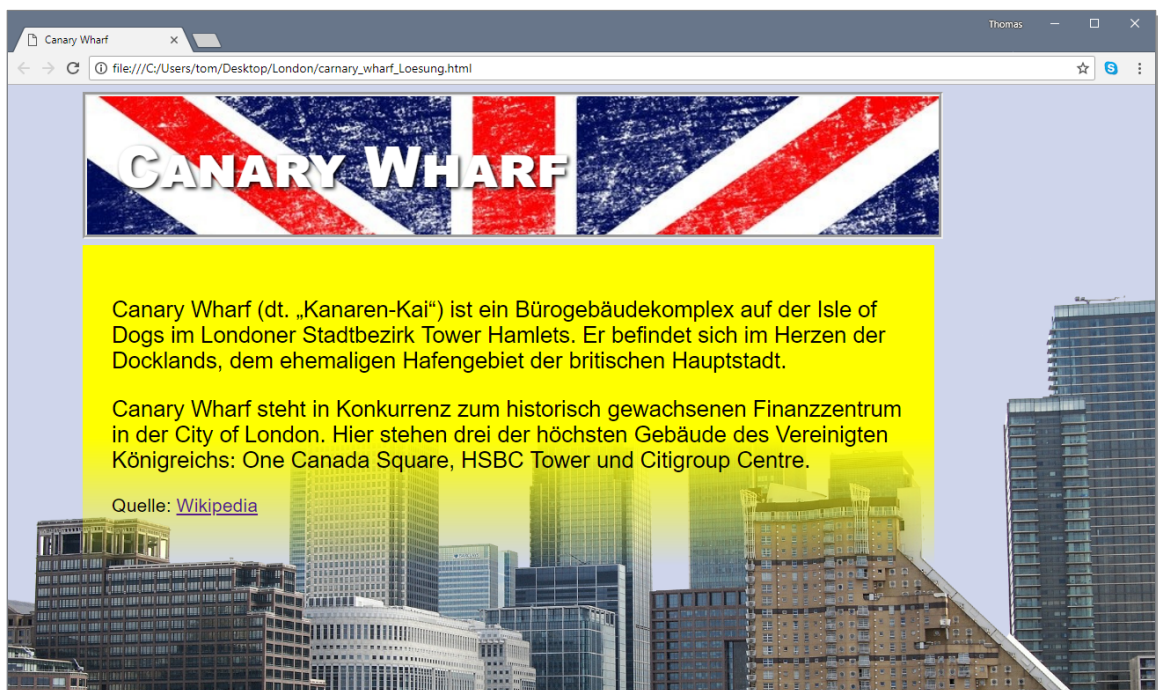
## Anmerkungen

- Zu Übung A Canary Wharf: Damit sollte ein pro europäischer Impuls gesetzt werden und wir gedenken Großbritannien als EU-Mitglied. (stand 2018, mitten im Brexit). England, wir werden dich vermissen.
- Rainbow Warrior ist der Name des Flaggschiffes von Greenpeace.



### Übung A: Canary Wharf

- Öffne die HTML Datei `canary_wharf.html` (in dieser HTML Seite sind zwei divs. Gekennzeichnet als **ERSTES DIV** und **ZWEITES DIV**).
- Im Seitenhintergrund soll das Bild `london.png` eingebettet werden. Das Hintergrundbild soll am unteren Seitenrand fixiert und auf die Seitenbreite skaliert dargestellt werden.
- Ändere die Farbe des Seitenhintergrund in ein angenehmes Blau.
- ERSTES DIV**: Suche im Internet nach einem Bild der Englischen Fahne (Union Jack) und binde es als Hintergrundgraphik im ersten Div ein.
- ZWEITES DIV**: Definiere einen Farbverlauf, von Oben nach Unten und von Gelb nach Transparent. Entferne den Rahmen um das zweite Div.
- Überprüfe ob in den Meta-Tags des Dokuments dein Name steht!**



### Übung B: Rainbow Warrior

- Definiere einen linearen Farbverlauf (in Regenbogenfarben) für den Schriftzug **RAINBOW WARRIOR**.
- Du brauchst vielleicht noch folgende CSS Eigenschaften:  
`background-clip` **und** `text-fill-color`;
- Recherchiere selbstständig nach den zusätzlichen CSS Eigenschaften und löse diese Aufgabe nur mit HTML und CSS!**

**RAINBOW WARRIOR**

# Lernhandout 8.1 Positionierung

Referenzcode: **HCL081**

Technologien: HTML, CSS

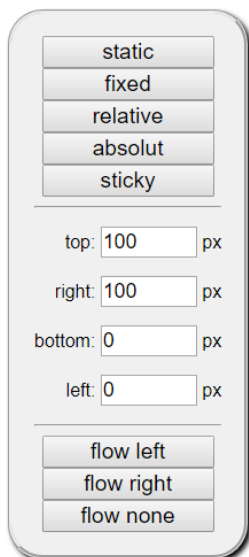
Dateinamen: L8\_1\_Position.odt, L8\_1\_Position.pdf, Beispiel\_Pos.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... positionieren Bilder und HTML Elemente im Viewport.	I	II
... kennen und nutzen die unterschiedlichen Positionierungsarten ( <b>static, fixed, relative, absolute, sticky</b> ).	I	II
... verstehen top, bottom, left und right im Zusammenhang mit der position Eigenschaft.	I	I
... erstellen fixe Fußzeilen (am Browserbottom).	I	II
... erstellen Bildbeschriftungen.	I	II
... kennen die Idee von Ebenen (z. B. wie bei Photoshop oder Gimp) und nutzen den <b>z-index</b> um diese zu bestimmen.	KI	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen



- Im HTML Dokument (Beispiel\_Pos.html) kann man die Auswirkungen von position und float ansehen. Links befindet sich eine "Konsole" und rechts im Text ein gelbes Rechteck, welches die Änderungen sofort visualisiert. Für diese Seite wurde Javascript verwendet.
- Das Beispiel (Fußzeile) kann mit sehr viel Fülltext besser visualisiert werden!
- Für das Beispiel (Bildbeschriftung) muss ein beliebiges Bild bereitgestellt werden. (zug.jpg)

HTML Elemente müssen nicht immer nur in Laufrichtung des Textes positioniert werden. Man kann jedes Element (`div`, Bilder usw) an jeder Stelle im Browserfenster (viewport) anzeigen. Dafür gibt es die CSS-Eigenschaften `top`, `bottom`, `left` und `right`. Diese können durch numerische Angaben (z. B. `top: 30px;`) bestimmt werden. Zuvor muss man dem Element eine Positionierungsart zuweisen.

## CSS



**position:** [value];

Für [value] sind folgende Werte möglich:

- static** Ist der Defaultwert. Das Element bleibt im Textfluss. `top`, `bottom`, `left` und `right` werden ignoriert.
- fixed** Richtet das Element am Viewport aus und fixiert es dort. Das Element bleibt beim Scrollen an seiner Position. (z. B. eine fixe Kopf- oder Fußzeile).
- relative** Richtet das Element von seiner eigenen Position im Textfluss aus. Es hinterlässt eine Lücke.
- absolute** Orientiert sich bei `top`, `bottom`, `left` oder `right` am Elternelement. Jedoch nur, wenn dieses Elternelement selbst mit `position: positioniert` wurde. Sonst bezieht es sich am `<html>` Element. Es behält beim Scrollen also seine Position.
- sticky** Behaltet seine Position im Textfluss, bis das obere oder untere Seitenende erreicht wurde. Dort bleibt es dann kleben!

Hier ein Beispiel für eine fixe Fußzeile:

```
#fusszeile {position: fixed;
            bottom: 0px; right: 0px;
            font-size: 1.4em; text-align: right;
            width: 100%; background: #F0F0F0;
            padding: 10px;}
```

```
<div id="fusszeile">techcom GmbH - &copy; 2023</div>
```



**Beispiel: Eine Beschriftung über einem Bild!**

```
.bilddiv {position: relative;}
.bild {height: 300px; z-index: 1;}
.beschriftung {position: absolute;
              bottom: 50px; left: 10px;
              padding: 10px; background: white; font-size: 1.5em;
              z-index: 2;}
```

```
<div class="bilddiv">
  
  <div class="beschriftung">Rail Jet</div>
</div>
```



Wenn Elemente sich überlappen, dann kann man mit `z-index` die Reihenfolge bestimmen, ähnlich wie die Ebenen im Photoshop oder Gimp.

# Lernhandout 8.2 display-Eigenschaft

Referenzcode **HCL082**

Technologien HTML, CSS

Dateinamen L8\_2\_Display.odt, L8\_2\_Display.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verändern die Art eines HTML Elements mit der display Eigenschaft.	I	II
... kennen die unterschiedlichen Werte der display Eigenschaft. (inline, block, inline-block, flex, grid, list-item, table)	I	I
... positionieren mehrere <div> Elemente nebeneinander (mit inline-block, ohne float).	I	II
... unterscheiden zwischen visibility: hidden; und display:none;	I	III
... erkennen den Element- bzw. Textfluss in einem HTML Dokument.	K	III
... zentrieren einen Text horizontal und vertikal über den gesamten Viewport mit display: table-cell;	I	II
... zentrieren einen Text horizontal und vertikal über den gesamten Viewport mit display: flex;	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Es gibt viele Möglichkeiten einen Text im Viewport-Zentrum anzuzeigen. z. B.

```
#imZentrum {display: inline-block;
margin-top: 50vh; width: 100%;
text-align: center;}
```

Mit der `display`-Eigenschaft bestimmt man die Art eines Elements näher.

## CSS



`display: [value];`

Für [value] sind folgende Werte möglich:

<b>inline</b>	Inline Boxen verlaufen in einer Zeile entlang der Schreibrichtung. Die Breite und Höhe wird allein durch den Inhalt bestimmt. <code>width</code> und <code>height</code> sind wirkungslos.
<b>block</b>	Block-Boxen haben die gleiche Breite wie das Elternelement. Die Höhe wird durch den Inhalt beeinflusst. Ein Beispiel für eine Block-Box ist der <code>&lt;p&gt;</code> Tag.
<b>inline-block</b>	Die Breite ist so schmal als möglich. Man kann die Breite mit <code>width</code> festlegen. Die Höhe ist vom Inhalt abhängig, kann aber mit <code>height</code> festgelegt werden.
<b>flex</b>	für flexible Layouts ohne fixe Größen (Flexible Box Layout Module)
<b>grid</b>	für flexible Layouts ohne fixe Größen. Im Gegensatz zu <code>flex</code> , werden hier komplexe Raster erzeugt. (Grid Layout)
<b>list-item</b>	behandelt ein Element wie einen <code>&lt;li&gt;</code> Tag.
<b>table</b>	Um ein Element als eine Tabelle darzustellen. In Folge kann ein Block auch als Tabellenzelle mit <code>table-cell</code> dargestellt werden.
<b>none</b>	Das Element wird nicht erzeugt. Es ist unsichtbar und hat keinen Einfluss auf den Elementfluss. (es hinterlässt keine Lücke im Text). <code>display: none;</code> wird oft verwendet wenn man das Medium (z. B. Print) bzw. den Viewport (z. B. für Smartphones) wechselt.

Das Beispiel zeigt, wie man mehrere `<div>` Elemente nebeneinander darstellen kann.

```
.nebenan {display: inline-block;
          width: 200px; height: 200px;
          margin: 10px; border: 3px groove blue;
          text-align: center;}
```

```
<div class="nebenan">EINS</div>
<div class="nebenan">ZWEI</div>
<div class="nebenan">DREI</div>
```



Der Unterschied zwischen `visibility: hidden;` und `display: none;` besteht darin, dass `visibility` das Element ausblendet, während `display: none;` es nicht erzeugt. Der Unterschied liegt dann im Element- bzw. Textfluss.



Zwei Beispiele wie man einen Text horizontal und vertikal über den gesamten Viewport zentriert: `<div id="haupt">Eilmeldung</div>`

```
#haupt {display: table-cell;
        background: #CCC;
        height: 100vh;
        width: 100vw;
        vertical-align: middle;
        text-align: center;}
```

```
#haupt {display: flex;
        background: #CCC;
        height: 100vh;
        align-items: center;
        justify-content: center;}
```

# Übungsblatt 8.2 Positionierung und Display

Referenzcode **HCU082**

Technologien HTML, CSS

Dateinamen U8\_2\_Display.odt, U8\_2\_Display.pdf, Wordelemente.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... positionieren HTML Elemente mit CSS.	<b>I</b>	II
... nutzen display-Eigenschaften, wie z. B. inline-block.	<b>I</b>	II
... erstellen eine fixierte Fußzeile.	<b>I</b>	II
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... binden mit HTML ein Bild (Screenshot) ein.	<b>I</b>	II
... erstellen eine Beschriftung über das Bild (überlappen).	<b>I</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



### Übung A: Textgestaltung

- Öffne die Lösung von 6.2 C: Text mit Spalten bzw. die Datei **Wordelemente.html**
- Lösche die Überschrift  
`<h1>Microsoft Word: Einfügen von Elementen</h1>`

1

- Links oben soll ein `<div>Textverarbeitung</div>` hinzugefügt werden. Gestalte es ansprechend. Das `<div>` soll nicht über gesamte Bildschirmbreite erscheinen und links über das schwarze `<div id="Hauptdiv">` hinausragen.

2

- Beim scrollen soll das `<div>` mitwandern!
- Füge eine weitere Überschrift-Ebene mit dem Text **"Einfügen von Elementen"** hinzu. Die Überschrift soll beim Scrollen am oberen Browserrand 'kleben' bleiben. Gestalte die Überschrift ansprechend!

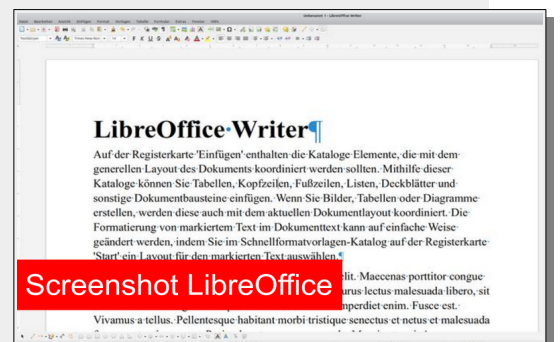
3

- Erstelle eine Fußzeile für das HTML-Dokument, die am unteren Browserrand fixiert werden und immer im Vordergrund sein soll. Der Text für die Fußzeile lautet: © by Vorname Nachname, Datum**



### Übung B: Textgestaltung mit Bild

- Öffne deine Lösung von 8.2 A: Textgestaltung.
- Besorge dir einen Screenshot von LibreOffice Writer und füge ihn als Bilddatei in den HTML Code ein.
- Beschrifte das Bild mit "Screenshot LibreOffice" Die Beschriftung soll auf das Bild gelegt werden.



# Lernhandout 8.3 Strukturierung

Referenzcode: **HCL083**

Technologien: HTML

Dateinamen: L8\_3\_Struktur.odt, L8\_3\_Struktur.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verstehen das Document Object Model (DOM).	I	I
... strukturieren HTML Seiten und erstellen damit u. a. barrierefreie Webseiten.	I	II
... definieren die Hauptnavigation mit <nav>.	I	II
... definieren den Hauptinhalt mit <main>.	I	II
... definieren einen Kopfbereich mit <header>.	I	II
... definieren den Schlussbereich für Hinweise (z. B. Impressum) mit <footer>.	I	II
... definieren semantische Gliederungen mit <section> und <article>.	I	II
... definieren Kontaktinformationen mit <adress>.	I	II
... definieren Randbemerkungen mit <aside>.	I	II
... erkennen, dass die meisten Tags zur Strukturierung nur geringfügige Auswirkung auf das Aussehen haben aber wichtig für die Semantik der HTML Seite sind.	I	III

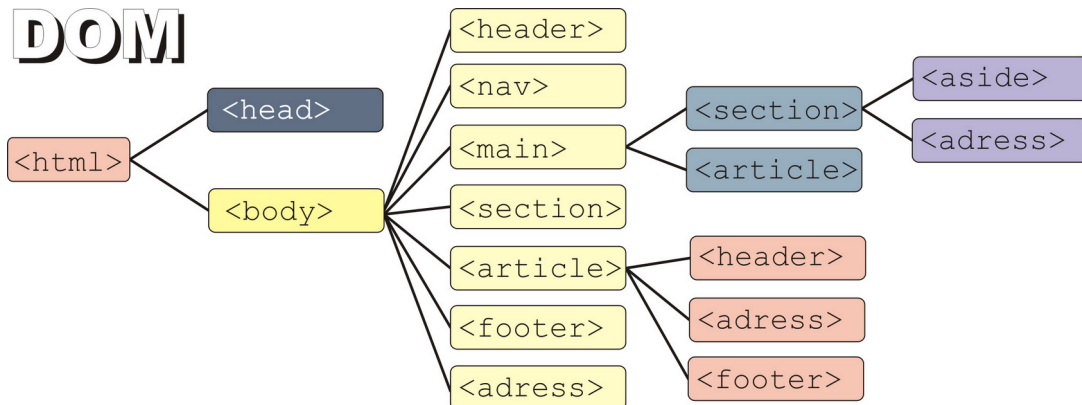
**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Alle erwähnten Elemente dienen hauptsächlich nur der Strukturierung und sind nicht zwingend notwendig. Es können genauso gut auch <div> Elemente verwendet werden!

Für die Seitenstruktur haben wir bisher `<body>`, `<div>`, `<p>` usw. verwendet. HTML5 bietet aber noch weitere Tags zur Strukturierung. Alle Tags müssen noch mit CSS formatiert werden. Die Strukturierung ermöglicht auch ein barrierefreies Web, weil Screenreader `<header>`, `<nav>` oder `<footer>` überspringen und nur die wichtigen Inhalte vorliest.

**HTML**

`<nav>` ... `</nav>`

Das `<nav>` Element bezeichnet die Hauptnavigation und bildet einen eigenen Inhaltsbereich.

**HTML**

`<main>` ... `</main>`

Das `<main>` Element enthält den Hauptinhalt der Webseite.

**HTML**

`<header>` ... `</header>` und `<footer>` ... `</footer>`

Das `<header>` Element wird für den sichtbaren Kopfbereich verwendet. Das `<footer>` Element enthält Informationen zur Website und steht am Ende. Beispiele für Infos im `<footer>` sind Hinweise zu Urheberrecht, Autor oder ein Link zum Impressum.

**HTML**

`<section>` ... `</section>` und `<article>` ... `</article>`

Das `<section>` Element umspannt eine thematische Gruppierung. Es dient zur semantischen Gliederung von Inhalten. `<article>` ist mit einem Zeitungsartikel vergleichbar und stellt einen sich geschlossenen Abschnitt dar.



Unter Semantik versteht man die Bedeutung bzw. den Inhalt eines Wortes, Satzes oder Textes.

**HTML**

`<adress>` ... `</adress>` und `<aside>` ... `</aside>`

Das `<adress>` Element enthält Kontaktinformationen (z. B. eMail-Adressen, Postadressen, Link zu Seiten mit Kontaktinformationen usw.). `<aside>` dient für Randbemerkungen. Die Darstellung muss über CSS festgelegt werden.

# Lernhandout 8.4 Float

Referenzcode **HCL084**

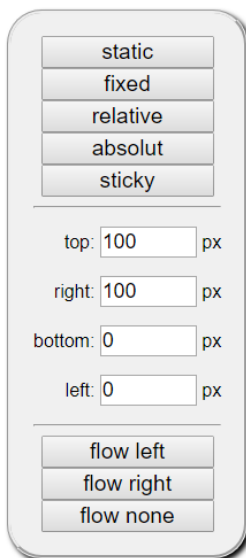
Technologien HTML, CSS

Dateinamen L8\_4\_Float.odt, L8\_4\_Float.pdf, Beispiel\_Pos.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verstehen den Text- und Elementfluss von HTML.	I	II
... verschieben Elemente nach links- bzw rechts und lassen den Text umherfließen. (CSS Eigenschaft float).	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... begreifen den collapse Effekt. (Kindelemente ragen über das Elternelement hinaus).	I	III
... wirken gegen den collapse Effekt mit clear:both;	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen



- Im HTML Dokument (**Beispiel\_Pos.html**) kann man die Auswirkungen von Position und **float** ansehen. Links befindet sich eine "Konsole" und rechts im Text ein gelbes Rechteck, welches die Änderungen sofort visualisiert. Für diese Seite wurde Javascript verwendet.

Alle HTML Elemente sind im Fluss. Um nun aber Bilder oder andere Elemente links oder rechts vom Fließtext darzustellen, gibt es die `float` Eigenschaft. Dabei wird das Objekt links oder rechts an die Innenkante des Elternobjekts verschoben.

## CSS



**float:** [Value];

Für **value** sind folgende Werte erlaubt:

- none** Standardwert, das Element wird nicht verschoben
- left** Das Element wird nach links verschoben.
- right** Das Element wird nach rechts verschoben.
- inherit** Erbt die **float**-Eigenschaft des Elternelements.

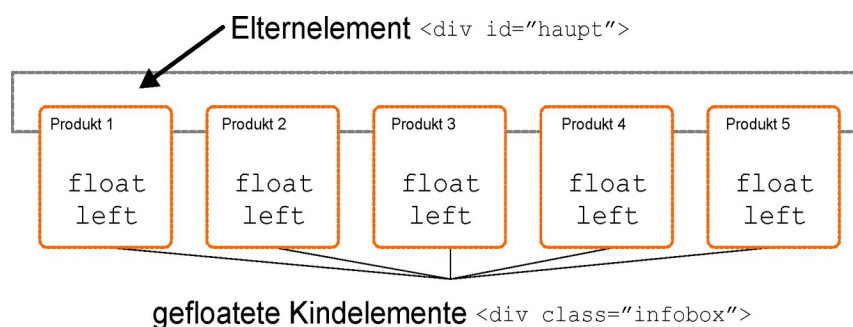
```
<p>... weiterer Lauftext ...</p>
```



Wenn man mehrere Elemente mit einer `float` Eigenschaft einbindet, dann sind die Kindelemente manchmal größer als das Elternelement und ragen über dieses hinaus. Dieser Effekt wird auch **collapse** genannt. Der Grund: Die Kindelemente werden wegen der `float` Eigenschaft aus dem Fließtext herausgenommen obwohl sie den Text verdrängen!

```
#haupt {border:2px dashed gray; min-height:50px;}
.infobox {float:left; margin:10px; height:200px; width:200px;
border:2px dotted orange;}
```

```
<div id="haupt">
  <div class="infobox">Produkt 1</div>
  <div class="infobox">Produkt 2</div>
  <div class="infobox">Produkt 3</div>
  <div class="infobox">Produkt 4</div>
  <div class="infobox">Produkt 5</div>
</div>
```



Um dem entgegenzuwirken gibt es mehrere Möglichkeiten. Eine davon ist es, am Ende ein leeres HTML Element mit der CSS Eigenschaft `clear:both;` hinzuzufügen.

```
<div id="haupt">
  ...
  <div class="infobox">Produkt 5</div>
  <div style="clear:both;">&nbsp;</div>
</div>
```

# Lernhandout 9.1 Pseudoklassen (Buttons)

Referenzcode **HCL091**

Technologien HTML, CSS

Dateinamen L9\_1\_Pseudoklassen.odt, L9\_1\_Pseudoklassen.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... kennen Syntax und Funktion von dynamischen Pseudoklassen.	I	I
... erstellen Buttons und Schaltflächen mit dynamischen Pseudoklassen.	I	II
... kennzeichnen Hyperlinks mit der Pseudoklasse :link.	I	II
... kennzeichnen besuchte Links mit der Pseudoklasse :visited.	I	II
... kennzeichnen alle Hyperlink-Zustände mit der Pseudoklasse :any-link.	I	II
... kennzeichnen ein Mouse-Over mit der Pseudoklasse :hover.	I	II
... kennzeichnen den Fokus (z. B. durch die Tabulatortaste) mit der Pseudoklasse :focus.	I	II
... kennzeichnen aktive Hyperlinks mit der Pseudoklasse :active.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

Um elegante Buttons mit HTML und CSS zu erstellen, kann man dynamische Pseudoklassen verwenden. Sie werden mit einem Doppelpunkt an die CSS Eigenschaft angeschlossen. z. B: `a:link {...}` `.imlink:hover {...}` `#derlink:visited {...}`  
Hier in Verbindung mit dem `a href` Element (Hyperlink):

## CSS

**a:link {...}**

Kennzeichnet nur unbesuchte Links. Die Verwendung ist in den `a-`, `area-` oder `link-` Elementen mit `href`-Attribut möglich.

```
a:link {text-decoration: none;}
```

## CSS

**a:visited {...}**

Kennzeichnet besuchte Links (also der User hat darauf geklickt). Achtung: Aus Datenschutzgründen unterstützen moderne Browser nur eine eingeschränkte Möglichkeit von Eigenschaften für diese Pseudoklasse!

```
a:visited {text-decoration: line-through;}
```

## CSS

**a:hover {...}**

Kommt zur Anwendung, wenn der Mauszeiger das Element berührt. Verlässt der Mauszeiger das Element, dann wird der vorherige Zustand wieder hergestellt. Die Pseudoklasse `:hover` lässt sich auf beinahe alle HTML Elemente anwenden.

```
a:hover {color:red;}
```

## CSS

**a:focus {...}**

Sobald der Fokus (z. B: durch die Tabulatortaste) auf ein HTML Element mit einer `:focus` Pseudoklasse gesetzt wird.

```
a:focus {border: 2px solid green;}
```



Weitere Pseudoklassen von Interesse im Zusammenhang mit Hyperlinks sind: `a:any-link` (Kennzeichnung von besuchten und unbesuchten Links) und `a:active` (sobald ein Hyperlink aktiviert wird).



Hier ein Beispiel für einen HTML-Button mit CSS für die Navigation:

```
.schalter:link {text-decoration: none; font-size: x-large;
padding: 10px; border: 2px groove gray;
border-radius: 10px;
background: LightGray; color: black;}

.schalter:focus, .schalter:hover, .schalter:active {background: blue;
color: white;}

~~~~~
<a href="kontakt.html" class="schalter">Kontakte</a>
<a href="about.html" class="schalter">Über uns</a>
```

# Lernhandout 9.2 Pseudoklassen

Referenzcode **HCL092**

Technologien HTML, CSS

Dateinamen L9\_2\_Pseudoklassen.odt, L9\_2\_Pseudoklassen.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... definieren aktive Verweise mit der Pseudoklasse <code>:target</code> .	I	II
... kennen Sprungmarken und Ankerpunkte.	I	I
... können Elemente über einen Link aus- bzw. einblenden.	I	II
... schreiben und verstehen ein umfangreiches Beispiel für --- <b>mehr lesen</b> --- Schaltflächen.	I	II
... kennen weitere strukturelle und dynamische Pseudoklassen.	I	I
... bekommen einen ersten Einblick über den Aufbau einer URL.	IK	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das Beispiel: Ein- und ausblenden eines Elements funktioniert folgendermaßen:
  1. Alle Elemente mit der Klasse `.details` werden ausgeblendet!
  2. Mit dem Pseudoelement `.details:target` wird genau jenes Element wieder angezeigt, das auch über die ID in der aufgerufenen URL erwähnt wird.
  3. Durch einen Klick auf --- **mehr lesen** --- (`a href="#d1"`) wird das Dokument neu geladen, aber eben mit der Sprungmarke zum `<div id="d1">` darunter. Es wird als `inline-block` dargestellt
  4. Klickt man auf `schließen`, dann wird das Dokument wieder neu geladen. Das bedeutet auch, dass alle Elemente der Klasse (`.details`) wieder ausgeblendet sind.

Mit der Pseudoklasse `:target` werden Eigenschaften eines gerade aktiven Verweises dargestellt. Den aktiven Verweis sieht man in der URL Leiste

`http://html.css4.at#L91` öffnet die Webseite und springt zum Element mit der `id="L91"`. Man spricht hier auch von Anker oder Sprungmarken.



#### Weitere Sprungmarken:

`href="#"` Springt zum Seitenanfang  
`href="#top"` Springt ebenfalls zum Seitenanfang  
`href=""` erzwingt ein Neuladen der Seite!

### CSS



#### `:target` {...}

Zur Aktivierung ist die Vergabe einer ID notwendig.  
 Die Pseudoklasse eignet sich besonders gut in Verbindung mit einer Klasse.



#### Beispiel: Ein- und ausblenden eines Elements.

Zuerst werden alle Elemente der Klasse `.details` ausgeblendet. Wegen `.details:target` wird ein Element wieder dargestellt, sobald die ID in der aufgerufenen URL steht (öffnen der Seite mit Sprungmarke)!

```
.details {display: none; background:#F0F0F0; padding:2em;}
.details:target {display:inline-block;}

<h2 id="ueber">Nachrichten</h2>
<p>Die USA plant ein Freihandelsabkommen ...
  <a href="#d1">--- mehr lesen ---</a></p>

<div id="d1" class="details">
  <h3>Weltweiter freier Handel</h3>
  <p>Die USA und die EU haben sich geeinigt.
    Die Strafzölle werden wieder abgeschafft.</p>
  <a href="#ueber" title="schließen">schließen</a>
</div>

<p>Frieden in Korea
  <a href="#d2">--- mehr lesen ---</a></p>

<div id="d2" class="details">
  <h3>Koreas Atompolitik hat positive Folgen</h3>
  <p>Der Konflikt zwischen Nord- und Südkorea
    ist jetzt entgültig zu Ende.</p>
  <a href="#ueber" title="schließen">schließen</a>
</div>
```



Speichere den Code als `nachrichten.html` ab und öffne das Dokument direkt zum Ankerpunkt `d1` mit: `nachrichten.html#d1`



Es gibt noch eine Vielzahl von strukturellen und dynamischen Pseudoklassen wie:

`:first-child` Spricht das erste Kindelement an  
`:last-child` Spricht das letzte Kindelement an  
`:empty` Spricht leere Elemente an (z. B. ohne Text)

# Übungsblatt 9.2 Pseudoklassen

Referenzcode **HCU092**

Technologien HTML, CSS

Dateinamen U9\_2\_Pseudoklassen.odt, U9\_2\_Pseudoklassen.pdf,  
abfalltrennung.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen dynamische Buttons mit Pseudoklassen.	I	II
... antizipieren zur Lösung über die graphische Darstellung.	I	III
<b>Übung B</b>		
... bearbeiten ein fremdes HTML Dokument.	I	II
... arbeiten nach Vorgaben.	I	II
... verändern die Hyperlinks im Dokument.	I	II
<b>Übung C</b>		
... erstellen eine horizontale Navigationsleiste.	I	II
... definieren Sprungmarken für einzelne Navigationspunkte.	I	II
... fügen dynamische Effekte mit Pseudoklassen hinzu.	I	II
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
<b>Übung D</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... blenden <div> Elemente ein bzw. aus.	I	III

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

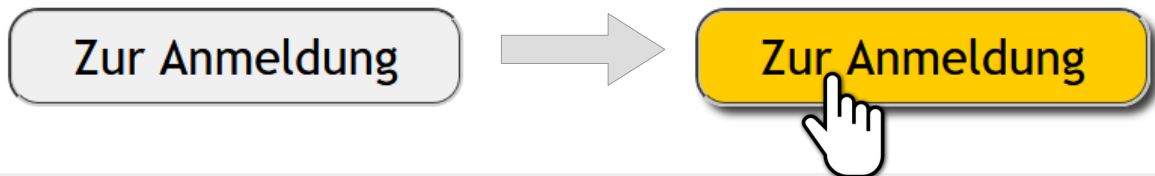
## Anmerkungen

- Die Übungen B, C und D sind ein kleiner Beitrag zum Unterrichtsprinzip Umweltschutz.



### Übung A: Buttons

- Gestalte eine Button-Klasse (wie unten dargestellt).
- Verwende die Pseudoklassen `:link`, `:visited`, `:any-link`, `:hover`, `:focus`, und `:active`.



### Übung B: Links neu gestalten

- Öffne das HTML Dokument `abfalltrennung.html`
- Alle Links (a href) sollen folgendes Aussehen haben:
- Jeder Link: keine Unterstreichung, dunkelgraue Schriftfarbe, Fettschrift
- Besuchte Links: Dunkelgrüne Schriftfarbe
- Hover: gepunktete Unterstrichen, blaue Schriftfarbe
- Fokus: einfach Unterstrichen, rote Schriftfarbe
- Aktive Links: doppelt Unterstrichen, schwarze Schriftfarbe



### Übung C: Navigationsleiste

- Öffne deine Lösung von **9.2 B: Links neu gestalten** (`abfalltrennung.html`).
- Erstelle eine horizontale Navigationsleiste.
- Die einzelnen Elemente sollen einen `:hover` und einen `:focus` Effekt haben.
- Folgende Navigationspunkte sind vorgesehen:  
**Papier | Leichtverpackung | Metall | Bio | Restmüll**
- Definiere für die Navigationspunkte Sprungmarken (bzw. Anker) zu den dazu passenden Müllsorten.
- Freie kreative Übung für die Gestaltung und Aussehen!**



### Übung D: Ein- und ausblenden

- Öffne deine Lösung von **9.2 C: Navigationsleiste** (`abfalltrennung.html`).
- Blende alle `<div class="sorte">` mit CSS aus.
- Per Klick auf einen Navigationspunkt soll das dazu passende `<div class="sorte">` wieder eingeblendet werden.

# Lernhandout 9.3 iFrames

Referenzcode: **HCL093**

Technologien: HTML

Dateinamen: L9\_3\_iframe.odt, L9\_3\_iframe.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... nutzen iFrames um ein "Objektfenster" in den HTML Code einzubauen.	I	II
... kennen und nutzen die Attribute eines iFrame.	I	II
... können YouTube Videos in ein HTML Dokument einbetten.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- YouTube, <https://www.youtube.com>

## Anmerkungen

- Zur Zeit (HTML5 und CSS3, 2024) sind iFrames eher kritisch zu betrachten. Die Verwendung von <object> Elementen sind definitiv besser, aber da Google seine Maps und Kalender als iFrame anbietet und von YouTube ebenfalls ein iFrame angeboten wird, kann man dieses wohl doch noch bedenkenlos thematisieren.
- Auf Frames und Framesets wird gänzlich verzichtet, weil diese nicht mehr Zeitgemäß sind und weil mit iFrames der gleiche Effekt erzielt werden kann.
- Im Jahr 2022 hat YouTube noch iframes verwendet, um das Einbetten von Filmen für eine HTML Seite zu ermöglichen! Es wurden Optionen zum Einbetten angeboten, die aber bei einer W3C Validierung als Fehler angezeigt wurden. Die meisten Browser haben das YouTube Video dennoch korrekt dargestellt.
- Die Verwendung von YouTube im Unterricht kann auch zu einem kritischen Reflektieren zum Unterrichtsprinzip "Mediennutzung" führen.
- Das Einbetten von fremden Code ist immer etwas heikel. So kann man nicht verhindern, das YouTube unerwünschte Werbung auf der eigenen Webseite schaltet. Dafür erspart man sich aber eine Menge Webspace Speicherbedarf.

Das `iFrame` Element wird zum Einbinden von HTML-Seiten oder anderen Inhalten (z. B. Bilder) verwendet. Dabei werden fremde Quellen eingebunden (von innerhalb oder außerhalb der Site) und auf der eigenen Webseite angezeigt.

## HTML



```
<iframe name="" src="" sandbox=""></iframe>
```

Das `<iframe>` Element hat folgende Attribute:

<b>height</b>	Höhe (CSS wird empfohlen)
<b>width</b>	Breite (CSS wird empfohlen)
<b>name</b>	Name des <code>iframes</code> (wird benötigt um später über einen Link <code>a href</code> mit dem Attribut <code>target</code> etwas einzubinden).
<b>src</b>	Quelle die beim Start gezeigt werden soll.
<b>sandbox</b>	Sicherheitseinstellungen:
	<b>allow-forms</b> Formularabsendungen sind erlaubt
	<b>allow-popups</b> erlaubt Popups
	<b>allow-scripts</b> erlaubt Scripting
	...

```
<h1>Österreichische online Tageszeitungen</h1>
<a href="https://www.kurier.at" target="meinIframe">Kurier</a>
<a href="https://www.krone.at" target="meinIframe">KronenZeitung</a>
<a href="https://www.oe24.at" target="meinIframe">oe24.at</a><br>

<iframe height="500" width="300" name="meinIframe"
src="https://derstandard.at/" sandbox="allow-forms"></iframe>
```







**Achtung:** Manch fremde Websites verhindert die Darstellung in iFrames mit technischen Hilfsmitteln.



### Einbetten eines YouTube Videos!

1. YouTube.com besuchen und ein Video auswählen
2. Auf **TEILEN** klicken
3. Auf **EINBETTEN** klicken
4. Den Quellcode mit dem `<iframe>` kopieren und in die eigene Webseite einfügen

Teilen

Einbetten WhatsApp Facebook Twitter

<https://youtu.be/vLom-87Am08>

Video einbetten

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
vLom-87Am08" title="YouTube video
player" frameborder="0"
allow="accelerometer; autoplay;
```

# Übungsblatt 9.3 iFrames

Referenzcode: **HCU093**

Technologien: HTML, CSS

Dateinamen: U9\_3\_iframe.odt, U9\_3\_iframe.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... stellen PDF's in iFrames dar.	I	II
... stellen Überlegungen zur besten Darstellung von DIN A4 Seiten in HTML an.	K	II
... testen die Darstellung in verschiedenen Browsern.	I	II
<b>Übung B</b>		
... bringen eigene Interessen und Vorlieben aus ihrer unmittelbaren Lebenswelt durch ein YouTube-Video zum Ausdruck.	A	III
... binden das YouTube Video mit einem iFrame in ein HTML Dokument ein.	I	II
<b>Übung C</b>		
... beschäftigen sich mit Google Maps.	I	III
... binden eine Google Map in ein HTML Dokument ein.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- YouTube, <https://www.youtube.com>
- Google Maps, <https://www.google.at>

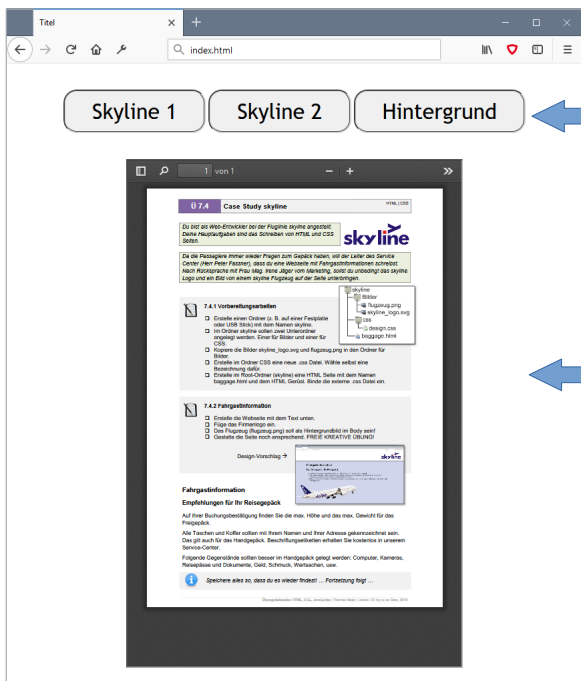
## Anmerkungen

- Zu Übung A: Auf dem Screenshot sieht man ein PDF vom Case-Studie skyline.
- Zu Übung B und C: Es könnte zu Änderungen auf den externen Websites YouTube und Google kommen.
- Eine schöne Alternative zu Google Maps bietet [openstreetmap.org](https://openstreetmap.org) an. Auch dort ist es möglich, Kartenausschnitte mittels iFrame einzubetten.



### Übung A: PDF darstellen

- Erstelle ein neues HTML Dokument
- Füge ein **iFrame** mit den ähnlichen Seitenverhältnissen eines DIN-A4 Blattes hinzu.  
TIPP: A4 hat 210 mm Breite und 297 mm Höhe,  
(z. B.: Breite: 500 px und Höhe: 707 px)
- Erstelle drei Link-Buttons.
- Verlinke die Buttons mit beliebigen PDF Dokumenten (im Din A4 Format).  
Die Darstellung soll im `iframe` passieren.



Buttons (Target ist das iframe)

Die meisten modernen Browser können PDFs darstellen. Einige skalieren die Darstellung brauchbar, einige blenden zusätzlich noch eine Navigation mit ein. Es ist also ratsam die PDF Darstellung in einem iframe unbedingt auf verschiedenen Browsern zu testen.



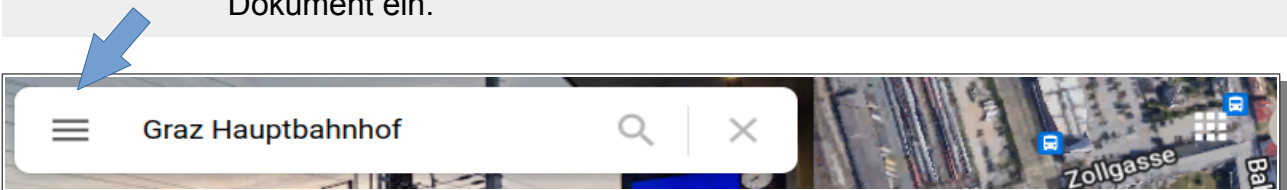
### Übung B: YouTube

- Erstelle ein neues HTML Dokument
- Suche auf YouTube.com ein cooles Video und bette es in das HTML Dokument ein.



### Übung C: Google Maps

- Erstelle ein neues HTML Dokument
- Suche auf Google Maps eine Adresse und bette die Karte in dein HTML Dokument ein.



# Lernhandout 10.2 Pseudoelemente

Referenzcode **HCL102**

Technologien CSS

Dateinamen L10\_2\_Pseudoelemente.odt, L10\_2\_Pseudoelemente.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... unterscheiden die Syntax von Pseudoklassen und Pseudoelementen.	<b>K</b>	I
... definieren die erste Zeile mit <code>::first-line</code>	<b>I</b>	II
... definieren den ersten Buchstaben mit <code>::first-letter</code>	<b>I</b>	II
... fügen vor und/oder nach einem Element einen Content mit <code>::before</code> und <code>::after</code> hinzu.	<b>I</b>	II
... definieren die Erscheinung eines markierten Textes mit <code>::selection</code> .	<b>I</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die Schreibweise mit zwei Doppelpunkten ist nicht unbedingt bindend, da die meisten Browser die Pseudoelement auch mit nur einem Doppelpunkt problemlos darstellen können.
- Mit der CSS Anweisung `user-select: none;` wird das Markieren von Text unterbunden und damit ein Copy-and-Paste verhindert. Begabte Webdevolper\_innen finden aber trotzdem einen Weg um den Text zu kopieren. Sei es nun über den Quelltext selbst oder mit Hilfe der Webdevolper-Tools des Browsers, in dem man einzellen CSS-Anweisungen einfach deaktiviert

Pseudoelemente können an einfache Selektoren (ID, Class usw.) angehängt werden um damit ein Element zu erweitern. Um Pseudoelemente von Pseudoklassen zu unterscheiden, werden sie mit zwei Doppelpunkten (::) geschrieben.

## CSS



```
::first-line {...}
```

Spricht die erste Text-Zeile eines Elementes an.

```
p::first-line {font-weight:bold;}
```

## CSS



```
::first-letter {...}
```

Spricht das erste Zeichen in einem Element an.

```
p::first-letter {font-size:4em; float:left; padding:10px;}
```

## CSS



```
::before {content: ...}
```

Fügt für ein Element zusätzliche Inhalte hinzu.

```
span::before {content: ' OK '};
```

Vor jedem <span> wird ein OK hinzugefügt.

```
p::before {content: url("icon.png");}
```

Vor jedem <p> wird ein PNG-Bild hinzugefügt.

## CSS



```
::after {content: ...}
```

Analog zu ::before. Der Inhalt wird hier nach dem Element angezeigt.

```
a::before {content: "Link: "};
a::after {content: " OK "}
p::after {content: ""; display: inline-block;
background: url("icon.png") no-repeat;
width: 50px; height: 50px;}
```

## CSS



```
::selection {...}
```

Definiert die Hintergrund- und Schriftfarbe eines markierten Textes.

Das Beispiel zeigt eine Möglichkeit, um das Markieren von Text optisch zu unterbinden:

```
div {color:black; background:gray;}
p::selection {color:inherit; background:inherit;}
~~~~~
<div><p>Sehr viel Laufertext</p> ist nicht immer gut!</div>
```

# Übungsblatt 10.2 Pseudoelemente

Referenzcode **HCU102**

Technologien HTML, CSS

Dateinamen U10\_2\_Pseudoelemente.odt, U10\_2\_Pseudoelemente.pdf,  
Wordelemente.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... arbeiten mit einem fremden HTML Dokument.	I	II
... binden Screenshots ein und floaten diese.	I	II
... verändern den ersten Buchstaben (Initial) jedes Absatzes mit CSS.	I	II
... verändern die erste Zeile jedes Absatzes mit CSS.	I	II
<b>Übung B</b>		
... suchen im Internet nach Programmsymbolen.	S	II
... schreiben für jeden Dateityp eine eigene CSS-Klasse.	I	II
... ergänzen die Dateityp-Klassen um die dazu passenden Programmsymbole.	I	II
... ergänzen die Dateityp-Klassen um die dazu passende File-Extension.	I	II
... gestalten mit einer Graphik als Vorlage.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Zur Übung A: Die drei Bilder sind eigentlich nicht so wichtig. Deshalb können irgend welche Screenshots von einem beliebigen Textverarbeitungsprogramm benutzt werden. Es ist nur eine Wiederholung des bisher gelernten. Wichtig jedoch sind das Initial und die erste Zeile jedes Absatzes.
- Zur Übung B: Die Übung soll auf ein Processing vorbereiten (wie z. B. Listen ausgeben mit PHP und MySQL). Die Tabelle soll mit beliebigen Dateinamen der verwendeten Dateitypen gefüllt werden!
- Die Schreibweise (KiB) für die Dateigröße bedeutet Kilobyte im echten Sinne. Das i steht für interpoliert - denn 1 KB sind 1000 Bytes und 1 KiB eben 1024 Bytes.

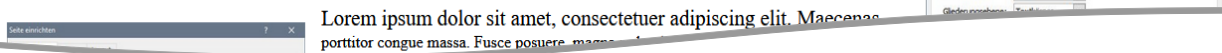


### Übung A: Wordelemente

- Öffne die Webseite **Wordelemente.html**
- Erstelle drei beliebige Screenshots von Microsoft Word, LibreOffice oder einem anderen Textverarbeitungsprogramm.
- Binde die drei Bilder an beliebiger Stelle ins HTML Dokument ein.
- Floate zwei Bilder rechts und eines links.
- Der erste Buchstabe jedes Absatzes soll ein Initial sein. (Hervorgehobener Buchstabe).
- **Die erste Zeile jedes Absatzes soll etwas hervorgehoben werden!**




### Microsoft Word: Einfügen von Elementen

Auf der Registerkarte 'Einfügen' enthalten die Kataloge Elemente, die mit dem generellen Layout des Dokuments koordiniert werden sollten. Mithilfe dieser Kataloge können Sie Tabellen, Kopfzeilen, Fußzeilen, Listen, Deckblätter und sonstige Dokumentbausteine einfügen. Wenn Sie Bilder, Tabellen oder Diagramme erstellen, werden diese auch mit dem aktuellen Dokumentlayout koordiniert. Die Formatierung von markiertem Text im Dokumenttext kann auf einfache Weise geändert werden, indem Sie im Schnellformatvorlagen-Katalog auf der Registerkarte 'Start' ein Layout für den markierten Text auswählen.



### Übung B: Datei Explorer

- Suche im Internet nach Programmsymbolen für
  - Word Dateien (.docx)
  - Excel Dateien (.xlsx)
  - PDF Files (.pdf)
- Erstelle ein HTML Dokument mit einer Tabelle. Die Tabelle soll die Spalten Dateiname, Datum, Typ und Größe haben!
- Schreibe für jeden Dateityp (Word, Excel, PDF) eine eignen CSS-Klasse.
- Vor jeder Klasse soll das dazu passende Icon (Programmsymbol) automatisch hinzugefügt werden. Verwende dafür: `::before`
- **Nach jeder Klasse soll die dazu passende Dateierweiterung automatisch hinzugefügt werden. Verwende dafür: `::after`**

Dateiname	Datum	Typ	Größe
 Planung.docx	2018-06-30	Word-Dokument	30 KiB
 Kostenrechnung.xlsx	2018-06-29	Excel-Dokument	78 KiB
 Endbericht.pdf	2018-06-28	PDF-Dokument	308 KiB

# Lernhandout 10.3 Audio

Referenzcode **HCL103**

Technologien HTML

Dateinamen L10\_3\_Audio.odt, L10\_2\_Audio.pdf, Europahymne.mp3

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... stellen Überlegungen zur Verwendung von Multimediaelementen für das Web an.	<b>K</b>	III
... verwenden das <audio> Element um Musik und Soundfiles in ein HTML Dokument einzubinden.	<b>I</b>	II
... kennen und nutzen die Attribute des <audio> Elements.	<b>I</b>	II
... verstehen was ein Fallback ist.	<b>I</b>	I
... definieren ein Fallback für das <audio> Element mit einem <source> Tag.	<b>I</b>	II






**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Es ist ratsam die Schüler\_innen mit Kopfhörern auszustatten, denn wenn das Plenum (jeweils jeder einzelne) die Europahymne abspielt, dann kann es schon zu extrem viel "Freude schöner Götterfunken" kommen. Die Übungsbeispiele sollen dem Unterrichtsprinzip Europa entsprechen.

Mit HTML 5 kann man auch Audio- und Videodateien einbinden. Ratsam ist es, mit Multimediaelementen gut überlegt umzugehen. Automatisch abgespielte Soundfiles können den Benutzer entnerven – große Dateien brauchen selbstverständlich mehr Ladezeit.

Browsersupport					
<code>&lt;audio&gt;</code>	4.0	9.0	3.5	4.0	10.5

## HTML

`<audio> </audio>`

Es können Musik und Soundfiles in den Formaten .mp3, .wav und .ogg abgespielt werden. Wobei .mp3 den Vorzug haben sollte, weil dieses Format von den meisten Browsern unterstützt wird.

Folgende Attribute sind für das `<audio>` Element verfügbar:

<b>autoplay</b>	beginnt sofort mit dem Abspielen	ohne Wert
<b>controls</b>	zeigt die Steuerungsfunktion an (Lautstärke usw.)	ohne Wert
<b>loop</b>	das Abspielen wird endlos wiederholt	ohne Wert
<b>muted</b>	zu Beginn stumm schalten	ohne Wert
<b>preload</b>	definiert das Ladeverhalten:	
	<b>auto</b> ← die gesamte Datei wird geladen	
	<b>none</b> ← die Datei wird nicht vorgeladen	
	<b>metadata</b> ← es werden nur die Metadaten geladen	
<b>src</b>	URL zur Audiodatei	

Zusätzlich gibt es noch TimeRange Attribute für JavaScript:

<b>buffered</b>	liest aus, wie viel zwischengespeichert wurde
<b>played</b>	liest aus, wie viel schon gespielt wurde.

Der Text zwischen dem `<audio>` Tag wird angezeigt, wenn das Abspielen nicht möglich ist. z. B. weil es in den Browsereinstellungen deaktiviert wurde.

```
<audio controls loop src="Europahymne.mp3">
  Keine Audiowiedergabe möglich!</audio>
```



Ein Fallback ist eine Alternative die abgespielt werden soll, wenn das primäre Sound-Format nicht abgespielt werden kann. Dafür bekommt das `<audio>` Element weitere `<source>` Tags:

```
<audio controls autoplay >
  <source src="Europahymne.mp3" type="audio/mp3">
  <source src="Europahymne.ogg" type="audio/ogg">
  Keine Audiowiedergabe möglich!
</audio>
```

# Übungsblatt 10.3 Audio

Referenzcode **HCU103**

Technologien HTML, CSS

Dateinamen U10\_3\_Audio.odt, U10\_3\_Audio.pdf,  
Bluesfiles.zip --> (guitar.wav, guitar2.wav, guitar3.ogg,  
guitar3.wav, harmonica.mp3, harmonica2.wav, sample.mp3, voice.mp3)

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen eine Webseite und binden Soundfiles ein.	I	II
... beschriften die <audio> Tags mit einem Hinweis.	I	II
... finden für die Soundfiles die passenden <audio> Attribute.	I	III
... definieren einen Fallback.	I	II
... testen das HTML Dokument auf verschiedenen Browsern.	I	II
... arbeiten strukturiert nach Vorgabe.	S	II
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

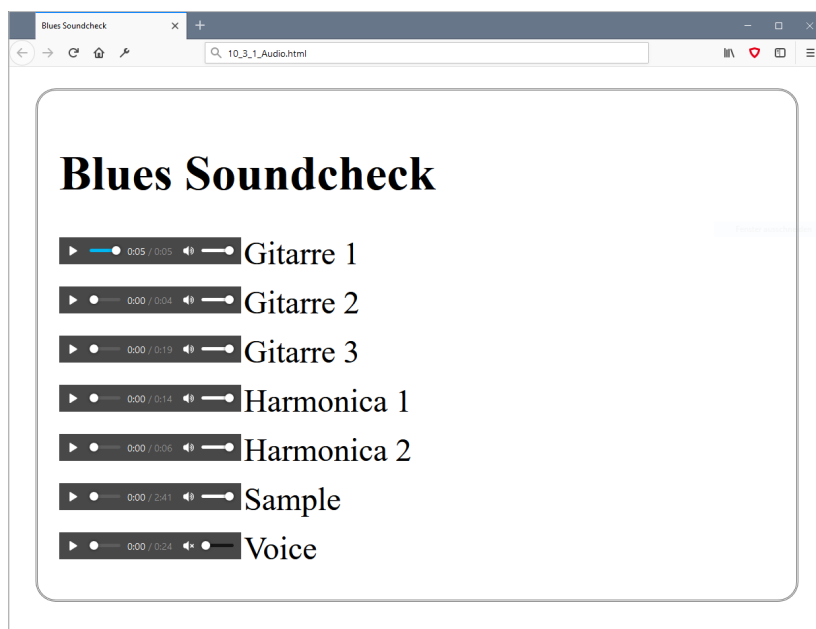
## Anmerkungen

- Die Soundfiles aus dem Container Bluesfiles.zip sind akribisch ausgesucht und wurden mit der Common Creative Lizenz CC0 gekennzeichnet.
- Sollten die Übungsaufgaben im Unterricht gelöst werden, so sind Kopfhörer mehr als angebracht.
- Die meisten Browser können die Soundfiles "übereinander" abspielen - so als würde man mehrere Tonspuren gleichzeitig laufen lassen.



### Übung A: Blues Soundcheck

- Entzippe die Datei **Bluesfiles.zip**
- Erstelle ein neues HTML Dokument mit allen Vorarbeiten (Title, Datum usw.)
- Füge die Überschrift "Blues Soundcheck" hinzu.
- Füge alle Soundfiles aus der Datei **Bluesfiles.zip** mit dem `<audio>` Tag hinzu.
- Beschrifte die `<audio>` Tags mit einem Hinweistext falls die Darstellung nicht möglich ist und einer lesbaren Bezeichnung des Audioelements.
- Alle `<audio>` Elemente haben Bedienelemente (Play, Lautstärke usw.)
- **guitar.wav** soll beim Öffnen der Webseite sofort abgespielt werden.
- Erstelle für **guitar3.ogg** ein Fallback auf **guitar3.wav**.  
Die Idee: **guitar3.ogg** hat 311 KiB und soll deshalb auch bevorzugt abgespielt werden. Da aber .ogg Files nicht von allen Browsern abgespielt werden, soll alternativ die **guitar3.wav** mit 3369 KiB abgespielt werden.
- **harmonica.mp3** soll endlos wiederholt werden.
- **sample.mp3** hat eine Größe von 2.5 MiB und soll deshalb automatisch geladen, aber noch nicht automatisch abgespielt werden!
- **voice.mp3** soll zum Start stumm geschaltet sein.
- **Teste dein Webdokument auf verschiedenen Browsern!**



### Übung B: Blues Designverständnis

- Öffne deine Lösung von **10.3 A: Blues Soundcheck** und verpasse dem Dokument ein cooles Blues Design.
- **Freie kreative Übung!**

# Lernhandout 10.4 Video

---

Referenzcode **HCL104**

Technologien HTML, CSS

Dateinamen L10\_4\_Video.odt, L10\_4\_Video.pdf, paint.mp4

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... binden Videodateien in ein HTML Dokument ein.	I	II
... kennen und nutzen die Attribute des <video> Elements.	I	II
... kennen die Videoformate mp4, WebM und Ogg.	I	I
... erstellen ein Fallback und ein Vorschaubild.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene






**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

- Natürlich gibt es neben den Video-Formaten .mp4, .ogg und .WebM noch weitere Formate die abgespielt werden können. Da wären z. B. AV1, MPEG oder WebVTT. Ein Blick auf die Website: [caniuse.com](http://caniuse.com) lohnt sich. Dort werden die Browserkompatibilitäten umfangreich und übersichtlich dargestellt.

Mit HTML 5 kann man Videodateien einbinden ohne das Plugins oder Add-Ons notwendig sind. Die Nutzung des Video-Elements ist ähnlich wie die, des Audio-Elements. Ein Text innerhalb des <video> Tags wird angezeigt, wenn das Video nicht abspielbar ist.

Browsersupport					
<video>	4.0	9.0	3.5	4.0	10.5

## TML



## &lt;video&gt; &lt;/video&gt;

Es können Videos in den Formaten .mp4, .ogg und .WebM abgespielt werden. Wobei man .mp4 bevorzugen sollte, weil dieses Format von den meisten Browsern unterstützt wird.

Folgende Attribute sind für das <video> Element verfügbar:

<b>autoplay</b>	beginnt sofort mit dem Abspielen <b>ohne Wert</b>
<b>controls</b>	zeigt die Steuerungsfunktion an (Play usw.) <b>ohne Wert</b>
<b>loop</b>	das Abspielen wird endlos wiederholt <b>ohne Wert</b>
<b>muted</b>	den Ton auf stumm schalten <b>ohne Wert</b>
<b>preload</b>	definiert das Ladeverhalten: <b>auto</b> ← die gesamte Datei wird geladen <b>none</b> ← die Datei wird nicht vorgeladen <b>metadata</b> ← es werden nur die Metadaten geladen
<b>src</b>	URL zur Videodatei
<b>height, width</b>	Größenangaben (ohne Angaben wird die Größe automatisch erkannt)

```
#myvideo {position:fixed; top:0px; left:0px;
min-height:100%; min-width:100%;}
```

```
<video id="myvideo" src="paint.mp4" autoplay controls >
Das Video kann nicht dargestellt werden!</video>
```



MP4 = MPEG4-Dateien mit H264 video codec + AAC audio codec  
WebM = WebM-Dateien mit VP8 oder VP9 video codec + Vorbis audio codec  
Ogg = Ogg-Dateien Theora video codec + Vorbis audio codec



Ein Fallback ist eine Alternative die abgespielt werden soll, wenn das primäre Video-Format nicht angezeigt werden kann. Dafür werden im <video> Element weitere <source> Tags gesetzt. Zusätzlich kann mit dem poster Attribut ein Vorschaubild definiert werden.

```
<video controls autoplay poster="vorschaubild.png" >
<source src="paint.mp4" type="video/mp4">
<source src="paint.ogg" type="video/ogg">
Keine Videoanzeige möglich!
</video>
```

# Übungsblatt 10.4 Video

Referenzcode **HCU104**

Technologien HTML, CSS

Dateinamen U10\_4\_Video.odt, U10\_4\_Video.pdf, paint.mp4

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... binden ein mp4 Videofile in ein HTML Dokument ein.	I	II
... verwenden die Attribute autoplay und loop.	I	II
... erstellen eine Beschriftung des Videos in Form eines überlappten <div>.	I	II
... erweitern die Beschriftung um einen Hover-Effekt.	I	II
... kennen und nutzen die CSS Eigenschaft opacity.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das Beispielvideo paint.mp4 wurde von pixbay.com mit der Common Creative Lizenz CC0 entnommen.

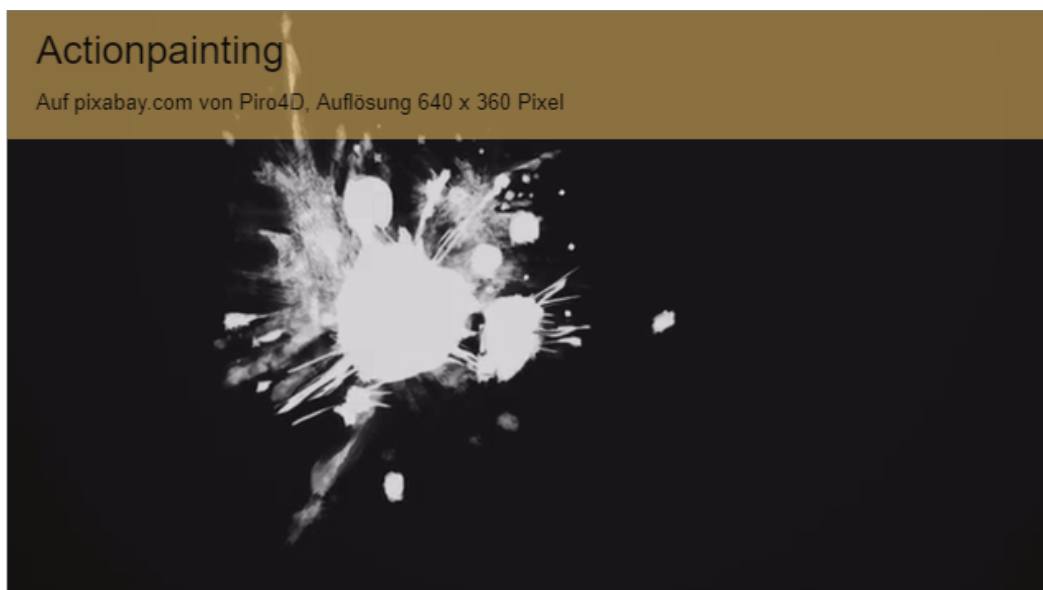


### Übung A: Video mit Beschriftung

- Erstelle eine neue HTML Seite.
- Binde die Videodatei paint.mp4 in das HTML Dokument ein.
- Das Video soll automatisch starten.
- Das Video soll sich ständig wiederholen.
- Definiere **über dem Video** ein Beschriftungs-Div mit dem Text:  
**Actionpainting**  
**Auf pixabay.com von Piro4D, Auflösung 640 x 360 Pixel**
- **Das Beschriftungs-Div soll unsichtbar sein und erst wenn man mit dem Mauszeiger darüberfährt (Hover-Effekt) erscheinen!**



*Es gibt mehrere Möglichkeiten, diese Aufgabe zu lösen. Am elegantesten ist wohl die CSS Eigenschaft `opacity`. Diese bestimmt die Transparenz (Durchsichtigkeit) eines HTML Elements.*



### CSS



`opacity: Value;`

Für den `value` ist ein Zahlenwert zwischen 0 und 1 möglich. Wobei 0 keine Deckkraft und 1 eine 100%ige Deckkraft bedeutet. 0.5 ist halbdurchlässig.

```
#meinDiv {opacity: 0.4;}
```

Browsersupport					
<code>opacity</code>	4.0	9.0	3.5	4.0	10.5

# Lernhandout 11.1 Medienabfrage

Referenzcode **HCL111**

Technologien CSS

Dateinamen L11\_1\_Medienabfragen.odt, L11\_1\_Medienabfragen.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... wissen was, Medienabfragen, Medientypen und Medienmerkmale sind.	I	I
... verstehen die Syntax von Medienabfragen.	I	II
... kombinieren Medientypen mit Medienmerkmalen.	I	II
... optimieren Webseiten für den Druck.	I	II
... optimieren Webseiten für unterschiedliche Anzeigeflächen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

Medienabfragen bestimmen die CSS Darstellung für ausgewählte Medientypen (z. B. Bildschirm, Druck) und/oder Medienmerkmalen (z. B. Höhe und Breite des Anzeigebereichs). Die `@media` Abfrage kann überall im CSS Bereich (z. B. im Head oder in einer externen CSS) eingebunden werden.

## CSS



`@media Medientyp and (Medienfeature) {...};`

Folgende *Medientypen* sind möglich:

<b>all</b>	alle Ausgabemedien (Standardwert)
<b>print</b>	nur für den Druck
<b>screen</b>	nur für Bildschirme
<b>speech</b>	für die Sprachausgabe

Es gibt eine Vielzahl von Medienmerkmalen. Diese *Medienfeatures* gehören zu den wichtigsten:

<b>width</b>	Breite eines Anzeigebereichs, <b>min-</b> und <b>max-</b> sind erlaubt.
<b>height</b>	Höhe eines Anzeigebereichs, <b>min-</b> und <b>max-</b> sind erlaubt.
<b>orientation</b>	Seitenformat eines Mediums. Möglich sind <b>landscape</b> (Querformat) und <b>portrait</b> (Hochformat)
<b>color</b>	Anzahl der Farbbits die ein Ausgabegerät anzeigen kann. Der Wert 0 entspricht einer Anzeige ohne Farben.
<b>resolution</b>	Die Auflösung (Dichte der Bildpunkte) auf dem Ausgabemedium.
<b>scripting</b>	Wenn Scriptsprachen (z. B. JavaScript) verfügbar sind.

Im Beispiel wird die Schriftart der Klasse `.schalter` größer, wenn die Anzeigebreite größer als 1200 Pixel ist.

```
@media screen and (min-width: 1200px) {
    .schalter {font-size:larger;}
}
```



Wenn man die Webseite für den Ausdruck vorbereiten möchte:

```
@media print {
    #inhalt {font-size: 12pt; width: 18cm; margin:auto;
        background-color: white;}
}
```



Wenn eine Anzeigefläche (z. B. Browserbreite) kleiner-gleich 600px ist:

```
@media screen and (max-width: 600px) {
    .schalter {display:none;}
}
```

# Übungsblatt 11.1 Medienabfragen

Referenzcode: **HCU111**

Technologien: HTML, CSS

Dateinamen: U11\_1\_Media.odt, U11\_1\_Media.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... definieren eine Medienabfrage für einen Viewport (Browserbreite) von mehr als 1020 Pixel.	<b>I</b>	II
... definieren eine Medienabfrage für einen Viewport (Browserbreite) mit weniger als 800 Pixel.	<b>I</b>	II
... definieren eine Medienabfrage für einen Viewport (Browserbreite) mit weniger als 600 Pixel.	<b>I</b>	II
... testen die Website mit den Web-Entwickler-Tools von Firefox.	<b>I</b>	II
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... nutzen Medienabfragen um Text ein- bzw. auszublenden.	<b>I</b>	IV
<b>Übung C</b>		
... erstellen einen Lebenslauf mit HTML und CSS.	<b>I</b>	II
... optimieren den Lebenslauf für den Druck.	<b>I</b>	IV
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



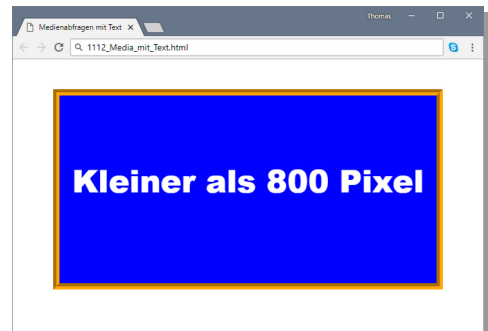
### Übung A: Media Queries

- Erstelle ein neues HTML Dokument.
- Füge ein `<div>` Element mit folgenden CSS Eigenschaften ein:  
Breite: 80%, Höhe: 300 Pixel  
Zentriert über die Bildschirmbreite  
Rahmen: 10 Pixel Groove in der Farbe Blau  
Hintergrundfarbe: Grün
- Wenn die Browserbreite größer als 1020 Pixel ist, soll das `<div>` einen schwarzen Hintergrund haben.
- Wenn die Browserbreite kleiner als 800 Pixel ist, soll das `<div>` einen blauen Hintergrund haben mit einem orangen Rahmen.
- Wenn die Browserbreite kleiner als 600 Pixel ist, soll das `<div>` keinen Rahmen mehr haben und eine Breite von 100% einnehmen.  
Hintergrundfarbe: Grün.
- **Teste deine Webseite**  
(z. B. mit Firefox → Web-Entwickler-Tools → Bildschirmgrößen testen).



### Übung B: Media Queries mit Text

- Öffne deine Lösung von oben (11.1 B: Media Queries).
- Bei einer Browserbreite von mehr als 1020 Pixel soll der Text: "**Größer als 1020 Pixel**" erscheinen.
- **Bei einer Browserbreite von weniger als 800 Pixel soll der Text: "Kleiner als 800 Pixel" erscheinen.**



### Übung C: Lebenslauf

- Erstelle einen Lebenslauf (von dir oder einer anderen prominenten Persönlichkeit) mit folgenden Inhalten:
  - Persönlichen Daten (Name, Geburtsdatum, usw.)
  - Berufspraxis (Ferialpraktika udgl.) und Ausbildung
  - Besondere Fähigkeiten (HTML, CSS, Buchhaltung usw.)
  - Hobbies (Webseiten schreiben usw.)
- Optimierte den Lebenslauf für Bildschirmdarstellungen und für den Druck!
- **Freie kreative Übung!**

# Lernhandout 11.2 Objekte

Referenzcode **HCL112**

Technologien HTML, CSS

Dateinamen L11\_2\_Objekte.odt, L11\_2\_Objekte.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... fügen Bilder mit dem <picture> Element ein.	I	II
... kombinieren das <picture> Element mit Medienabfragen.	I	II
... definieren einen Fallback für das <picture> Element.	I	II
... fügen diverse Objekte (z. B. Java Applets, ActiveX, Flash) mit dem <object> Element in ein HTML Dokument ein.	I	II
... kennen die Attribute des <object> Elements.	I	II
... binden Anwendungen oder interaktive Inhalte mit <embed> ein.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das Einbinden von externen Quellen ist meist davon abhängig, ob der Browser die entsprechenden Plug-Ins kennt bzw. installiert hat. Mit JavaScript kann man über das plugin Objekt fragen, ob bestimmte Plug-Ins verfügbar sind.
- Bei \*.swf handelt es sich um Adobe Flash Files. Diese zeichnen sich durch einen geringen Speicherbedarf und der Möglichkeit mittels Actionscript diese zu erweitern aus. Wird nicht mehr unterstützt, man benötigt ein Plug-In.

Bisher haben wir Bilder mit dem `img` Element eingefügt. Mit dem `picture` Element in Verbindung mit `source` können wir verschiedene Bildversionen für verschiedene Pixeldichten, Auflösungen und Bildschirmformate einbinden. Im `picture` Element werden die Bilder mit dem `source` Tag und dem `img` Element als Fallback angegeben.

## HTML



```
<picture> ... </picture>
```

Im Beispiel wird das Bild `zug_800.jpg` angezeigt, wenn der Anzeigebereich größer als 800 Pixel ist. Bei einem Viewport zwischen 800 und 600 Pixel wird das Bild `zug_600.jpg` verwendet – beide wurden mit dem `source` Tag eingebunden. Als Fallback dient das `img` Element mit dem Bild `zug_400.jpg`. Es wird angezeigt, wenn keines der anderen Bilder angezeigt werden kann, bzw. wenn das `picture` Element vom Browser nicht unterstützt wird.

```
<picture>
  <source media="(min-width: 800px)" srcset="Zug_800.jpg">
  <source media="(min-width: 600px)" srcset="Zug_600.jpg">
  <!--Fallback-->
  
</picture>
```

Das `object` Element definiert eingebundene Objekte innerhalb des HTML Dokuments. `object` erlaubt das Einbinden von Audio, Video, Java Applets, ActiveX, PDF und Flash Files. Darüber hinaus, kann man auch ein anderes HTML Dokument oder Bilder anzeigen lassen. Mit dem `param` Tag übergibt man Parameter an das Plug-In.

## HTML



```
<object> ... </object>
```

Folgende Attribute für `object` sind verfügbar:

<b>data</b>	bestimmt den Ort der Quelle.
<b>form</b>	Identifizierungsname für form Elemente
<b>height</b>	Höhe in Pixel oder %
<b>width</b>	Breite in Pixel oder %
<b>name</b>	Identifizierungsname des Elements
<b>type</b>	der Typ der Quelle

```
<object data="data/test.htm" type="text/html" width="300"
  height="200"> Alternativ: <a href="data/test.htm">test.htm</a>
</object>
```

## HTML



```
<embed> ... </embed>
```

Zum Einbinden von Anwendungen oder interaktiven Inhalten.

```
<embed src="helloworld.swf" type="application/x-shockwave-flash">
```

# Übungsblatt 11.2 Objekte

Referenzcode **HCU112**

Technologien HTML, CSS

Dateinamen U11\_2\_Objekte.odt, U11\_2\_Objekte.pdf,  
Smartphone\_Shop.zip --> (Aktion.pdf, background.jpg,  
flashfile.swf, phone1.jpg, phone2.jpg, phone3.jpg)

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... erstellen eine einfache Webseite.	<b>I</b>	<b>II</b>
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	<b>II</b>
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	<b>II</b>
... binden eine Flashanimation in das HTML Dokument ein.	<b>I</b>	<b>II</b>
... setzen Ankerpunkte vom Home Button auf die Flashanimation.	<b>I</b>	<b>II</b>
<b>Übung C</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	<b>II</b>
... fügen Bilder mit dem <picture> Elements ein.	<b>I</b>	<b>II</b>
... definieren Medienabfragen für die <source> Elemente innerhalb des <picture> Tags.	<b>I</b>	<b>II</b>
... fügen ein Fallback hinzu.	<b>I</b>	<b>II</b>
... setzen Sprungmarken vom Navigationsbutton zum <picture> Element.	<b>I</b>	<b>II</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



### Übung A: Smartphone Shop I

- Entzippe die Datei `Smartphone_Shop.zip`
- Erstelle ein neues HTML Dokument.  
Der `<body>` hat eine schwarze Hintergrundfarbe.
- Erstelle einen Banner mit der Bilddatei `background.jpg` als Hintergrund.
- Im Zentrum des Banners steht: "Smartphone Shop Graz"
- Unter dem Banner soll eine horizontale Navigation sein.
- In der Navigation sind folgende Buttons:  
**Home | Produkte | Angebote | Webshop | Kontakt**
- Gestalte die Webseite anspruchsvoll.
- Speichere die Webseite als `index.html` ab.



### Übung B: Smartphone Shop II

- Öffne dein HTML Dokument `index.html` aus der Übung 11.2 A: **Smartphone** Shop.
- Bette die Flashdatei `flashfile.swf` ins Dokument ein.
- Setze eine Sprungmarke/Anker vom Home Button auf die Flashdatei.



### Übung C: Smartphone Shop III

- Öffne dein HTML Dokument `index.html` aus der Übung 11.2 B: **Smartphone** Shop.
- Füge ein `<picture>` Element ein.
- Erstes Bild: `phone1.jpg`
- Zweites Bild: `phone2.jpg` (kleiner-gleich 800 Pixel Browserbreite).
- Drittes Bild: `phone3.jpg` (kleiner gleich 600 Pixel Browserbreite).
- `phone3.jpg` soll auch das Fallback-Bild sein!
- Setze eine Sprungmarke/Anker vom Produkte Button auf das `<picture>` Element.

# Lernhandout 11.3 Viewport

Referenzcode **HCL113**

Technologien HTML, CSS

Dateinamen L11\_3\_Viewport.odt, L11\_3\_Viewport.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... kennen die Probleme und Anforderungen von HTML Seite für mobile Geräte (Smartphone, Tablet).	I	III
... definieren einen Viewport.	I	II
... wissen, was Responsive Web Design bedeutet.	I	I
... nutzen die Vorgaben des RWD (Responsive Web Design) für die Gestaltung von Webseiten um die Darstellung auf allen Geräten zu gewähren.	I	II
... definieren einen Viewport für mobile Geräte.	I	II
... erweitern den Viewport-Content um ein Zoomlevel und Skalierbarkeit.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

Natürlich ist es unumgänglich, die Webseiten auch für Mobile Devices tauglich zu machen. Die Webseiten sollen auf Smartphones oder Tablets ebenso schön sein, wie bei einer Browserdarstellung. Dabei gibt es zwei Probleme. Erstens, haben Mobile Devices eine beträchtlich größere Pixeldichte (Auflösung) als ein herkömmlicher Bildschirm und zweitens sind ihre Screens viel kleiner. Das führt dazu, dass der Mobile Browser die Webseite zwar vollständig darstellen kann, aber Schriften, Bilder usw. sind dann sehr klein. Um dem entgegen zu wirken, kann man den Viewport definieren!

Dieser wird im `<head>` als `<meta>` Tag angegeben. Das Beispiel sagt den mobilen Geräten, dass die Seite mit einer Breite von 1024 Pixel dargestellt werden soll:

```
<head>
  <meta name="viewport" content="width=1024">
</head>
```

## Responsive Web Design

Responsive Web Design (RWD) ist ein Design und Layout-Konzept um Webseiten auf allen Geräten gut darzustellen. Es werden HTML Elemente in seiner Größe verändert, versteckt, ein- oder ausgeblendet und an eine andere Position gestellt. Dabei sollte nur HTML und CSS verwendet werden!

Für Mobile Devices setzen wir den Viewport auf die Screen-Breite des Gerätes:  
(Der Meta-Tag sollte auf jeder Website eingetragen werden!)

```
<meta name="viewport" content="width=device-width">
```



Im Attribut `content` des `<meta>` Tags, können auch das Zoom Level (`initial-scale`) bestimmt werden und ob der/die Nutzer/in die Seite zoomen kann (`user-scalable`).

```
<meta name="viewport"
  content="width=device-width, initial-scale=1.0, user-scalable=no">
```



Der/die User/in soll auf den mobilen Geräten zwar weiterhin vertikal scrollen, jedoch nicht horizontal. Ebenso soll ein Zoomen nicht notwendig sein, um die Inhalte gut zu lesen bzw. betrachten. Dafür ist folgendes zu beachten:

### **Verwende keine Elemente mit fixen Größenangaben.**

Wo immer es möglich ist sollte statt `px` besser `%` verwendet werden. Schriftgrößen sind in `em` oder ausgeschrieben (`medium`) empfehlenswerter als in `pt` oder `px`. usw.

### **Inhalte sollen nie auf eine bestimmte Darstellungsbreite angewiesen sein, um gut dargestellt zu werden.**

Wir müssen davon ausgehen, dass die Webseite sowohl auf kleinen Screens als auch auf extrem großen Screens aufgerufen werden kann. Auch ist nicht sicher, ob ein Browser immer im Vollbildmodus geöffnet ist. Deshalb sollte man die Elemente am besten im Fluss lassen – wo immer es möglich ist.

### **Verwende Medienabfragen (media queries) um das HTML Dokument für kleine und große Screens zu optimieren.**

# Übungsblatt 11.3 Viewport

Referenzcode: **HCU113**

Technologien: HTML, CSS

Dateinamen: U11\_3\_Viewport.odt, U11\_3\_Viewport.pdf,  
Smartphone\_Shop.zip --> (**Aktion.pdf**, background.jpg,  
flashfile.swf, phone1.jpg, phone2.jpg, phone3.jpg)

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... betten ein PDF File in ein HTML Dokument.	<b>I</b>	II
... setzen Ankerpunkte von der Navigation auf das PDF.	<b>I</b>	II
... fügen eine Google-Map hinzu.	<b>I</b>	II
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... optimieren eine Webseite für mobile Geräte.	<b>I</b>	II
... wandeln eine horizontale Navigation in eine vertikale Navigation.	<b>I</b>	II
<b>Übung C</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... fügen einen Button zum ein- und ausblenden des Navigationsmenü hinzu.	<b>I</b>	II
<b>Übung D</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... suchen Probleme und stellen Überlegungen zu dieser Webseite an.	<b>KI</b>	III
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln



### Übung A: Smartphone Shop IV

- Öffne dein HTML Dokument `index.html` aus der Übung 11.2 A: Smartphone Shop III.
- Bette das PDF-File `Aktion.pdf` ins Dokument ein.
- Setze eine Sprungmarke/Anker vom **Angebote Button** auf das PDF.
- Füge folgende Adress-Informationen hinzu:

**Smartphone Shop Graz**  
**Jakoministraße 15, A-8010 Graz**  
**Telefon: 0316 55587**  
**eMail: shop@smartgraz.at**

- Füge noch ein Google Maps `iframe` mit der Adresse oben hinzu und setze eine Sprungmarke vom Kontakt Button auf die Adress-Informationen.



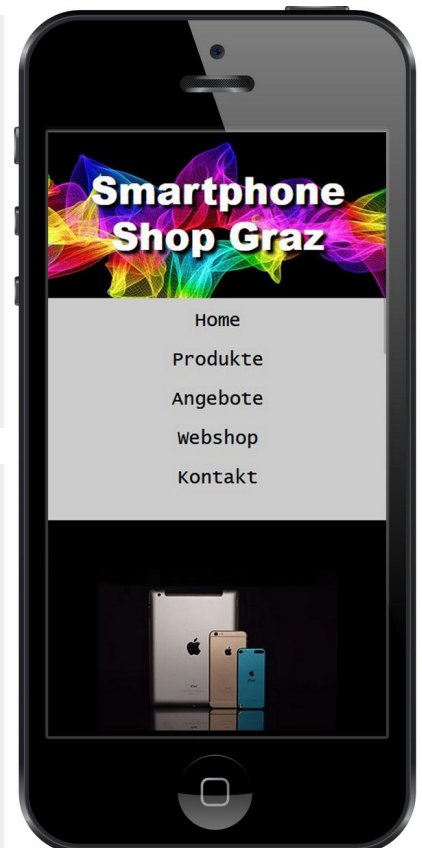
### Übung B: Smartphone Shop V

- Öffne dein HTML Dokument `index.html` aus der Übung 11.3 A: Smartphone Shop IV.
- Die Webseite soll für mobile Geräte optimiert werden (Viewport, Medienabfragen).
- **Die horizontale Navigation soll auf mobilen Geräten vertikal dargestellt werden!**



### Übung C: Smartphone Shop VI

- Öffne dein HTML Dokument `index.html` aus der Übung 11.3 B: Smartphone Shop V.
- Blende die Navigation für mobile Geräte aus.
- **Füge einen Button hinzu, mit dem man das Navigationsmenü wieder einblenden kann.**



### Übung D: Smartphone Shop VII

- Öffne dein HTML Dokument `index.html` aus der Übung 11.3 C: Smartphone Shop VI.
- Analysiere die Probleme, die sich für diese Webseite ergeben.
- Verbessere und viel wichtiger, verschönere die Webseite!
- **Freie kreative Übung!**

# Lernhandout 11.4 Box Eigenschaften

Referenzcode **HCL114**

Technologien HTML, CSS

Dateinamen L11\_4\_Box.odt, L11\_4\_Box.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... bestimmen für eine Box einen Schatteneffekt.	I	II
... kennen die unterschiedlichen Attribute von box-shadow.	I	II
... bestimmen das Verhalten eines Box-Inhalts (Text).	I	II
... kennen die unterschiedlichen Attribute von overflow.	I	II
... erlauben die Größenveränderung einer Box mit der CSS Eigenschaft resize.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

## Schatteneffekte für Boxen

## CSS



**box-shadow:** [X] [Y] [blur] [spread] [color];

[X]	Horizontaler Versatz des Schattens. 20px verschiebt den Schatten um 20 Pixel nach rechts. -10px verschiebt den Schatten um 10 Pixel nach links.
[Y]	Vertikaler Versatz des Schattens 20px verschiebt den Schatten um 20 Pixel nach unten. -15px verschiebt den Schatten um 15 Pixel nach oben.
[blur]	Radius der Unschärfe (Weichzeicheneffekt). 0px entspricht einer scharfen Darstellung.
[spread]	Zusätzliche Vergrößerung des Schattens. Bei 0px hat der Schatten die gleiche Größe wie sein Element.
[color]	Farbangabe

**Weitere Attribute von box-shadow:**

<b>inset</b>	wirft den Schatten ins Innere der Box. <b>Inset</b> darf nur am Anfang oder am Ende der Attribute gesetzt werden. <b>ohne Wert</b>
<b>none</b>	kein Schatten

```
.bilder {box-shadow: 10px 10px 5px 0px gray;}
div {box-shadow: inset -10px -10px 20px 10px black;}
```

## Anzeigen von Inhalten: overflow

Die HTML Elemente sollten so gestaltet werden, dass sie problemlos in der Eltern-Box Platz finden. Sollte es aber vorkommen, dass der Inhalt größer ist als die Eltern-Box, kann man mit der **overflow** Eigenschaft das Verhalten näher bestimmen.

## CSS



**overflow:** [value];

Für **value** stehen folgende Attribute zur Verfügung:

<b>auto</b>	Browserabhängig
<b>scroll</b>	Scrollbalken werden angezeigt
<b>visible</b>	Inhalte sind sichtbar und ragen über das Elternelement hinaus.
<b>hidden</b>	Inhalte werden an den Rändern unsichtbar (abgeschnitten).



Zusätzlich gibt es die CSS Eigenschaften **overflow-x** und **overflow-y** für horizontale und vertikale Inhalte die über die Box hinausragen.

```
div {overflow: hidden;}
```



Mit der CSS Eigenschaft **resize** erlaubt man dem\_ der Nutzer\_in das Verändern der Größen eines Elements.  
Folgende Werte sind möglich: **both**; **horizontal**; **vertical**; **none**;

```
div {resize: both;}
```

# Lernhandout 12.1 Formulare

Referenzcode: **HCL121**

Technologien: HTML

Dateinamen: L12\_1\_Formulare.odt, L12\_1\_Formulare.pdf, Report\_GET.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... wissen um den Nutzen von Formularen und Formularitems für die Verarbeitung mit Programmiersprachen oder Scriptsprachen bescheid.	I	I
... erstellen ein Formularelement (Parentelement, <form>).	I	II
... beschriften Formularelemente mit dem <label> Tag.	I	II
... verstehen das action-Attribut.	I	I
... verstehen das method-Attribut und unterscheiden zwischen get und post.	KI	II
... definieren einen Namen für das Formular.	I	II
... kennen noch weitere Attribute.	I	I
... verstehen und nutzen das <label> Element zur Beschriftung und kennen die Attribute form und for.	I	II
... nutzen <label> Elemente für das Barrierefreie Web. (Screenreader).	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das <label> Element kennt zusätzlich noch das Attribut form="" um ein bestimmtes Formular anzusprechen. Wenn es ohne for="" Attribut innerhalb eines Formular steht, dann bezieht es sich auf das Elternformular.
- Die Webanwendung **Report\_GET.html** (downloadbar über [www.css4.at](http://www.css4.at)) ließt die Wertübergabe aus der URL aus und stellt diese in einer Tabelle dar. Die Datei kann nur Wertübergaben mittels GET auslesen. Dahinter steckt ein JavaScript!

Verwendung:

```
<form action="Report_GET.html" method="get"> ... </form>
```

Zahlreiche Anwendungen lassen sich mit Formularen in HTML verwirklichen (z. B. Mail-Formulare). Formulare bieten die Möglichkeit zur Eingabe von Informationen – die Verarbeitung dieser Informationen erfolgt dann durch eine clientseitige Scriptsprache (z. B. JavaScript) oder einer serverseitigen-Sprache (z. B. PHP). Die Formularelemente werden in einem `<form>` Tag platziert und mit einem `<label>` Tag beschriftet.

## HTML



```
<form action="..." method="..." name="..."> ... </form>
```

<b>action</b>	Im <b>action</b> Attribut wird eine URL eingetragen. In den meisten Fällen ist das die Adresse zu einem Script. <b>action="mailscript.php"</b> Wird das <b>action</b> Attribut weggelassen, dann werden die Daten an das aktuelle Dokument gesendet.
<b>method</b>	Legt die Methode fest, wie die Daten versendet werden sollen. Es gibt zwei mögliche Werte: <b>get</b> Die Daten werden sichtbar über die URL-Zeile des Browsers weitergegeben. <b>post</b> Die Übertragung erfolgt unsichtbar im Hintergrund.
<b>name</b>	Definiert einen Namen für das Formular. Dieser ist meist notwendig für die weitere Verarbeitung mit z. B. JavaScript oder PHP.

Weitere Attribute:

<b>target</b>	Zielfenster
<b>accept-charset</b>	Angabe einer Zeichenkodierung
<b>enctype</b>	Angabe der Datenkodierung, Standardwert ist text/plain
<b>autocomplete</b>	Mit den Werten <b>on</b> oder <b>off</b> kann die Autovervollständigung des Browsers aktiviert, bzw. deaktiviert werden.
<b>novalidate</b>	Das Formular soll nicht auf Vollständigkeit geprüft werden.

```
<form action="mailversenden.php" method="get" > ... </form>
```

## HTML



```
<label > ... </label>
```

Das `<label>` Element hat zwei Vorteile. Erstens erleichtert es einen Screenreader das assoziierte Formelement vorzulesen. Zweitens wird bei einem Mouse-Click auf den Text innerhalb eines `<label>` Tags ein Focus auf das Formelement gelegt.

Mit dem Attribut **for=""** bezieht sich das `<label>` Element auf ein bestimmtes Formularelement (z. B. input) mit gleicher id.

```
<form action="senden.php" name="person" id="meinForm"
  autocomplete="off" method="post">
  <label for="vorname">Vorname</label>
  <input type="text" name="vorname" id="vorname" maxlength="30">
  <button type="submit">Eingaben absenden</button>
</form>
```

# Lernhandout 12.2 Textarea

Referenzcode **HCL122**

Technologien HTML

Dateinamen L12\_2\_Textarea.odt, L12\_2\_Textarea.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... nutzen Textarea-Elemente für mehrzeilige Texteingaben.	I	II
... definieren die Anzahl der Zeilen und Zeichen mit den rows und cols Attributen.	I	II
... kennen und nutzen weitere Attribute des <textarea> Elements.	I	II
... stellen HTML Code innerhalb eines HTML Dokument mit dem <textarea> Elements dar.	I	II
... vergleichen <textarea> und <pre> für die Darstellung von HTML Code.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- ID und Name Attribute können problemlos gleich lauten.  
ID wird eher für die Verarbeitung in JavaScript benötigt, aber auch für Sprungmarken oder CSS.  
Name ist notwendig für PHP.

Über das `<textarea>` Element kann der Benutzer mehrzeilige Texteingaben tätigen.

## HTML



```
<textarea name="" rows="" cols=""> ... </textarea>
```

<b>name</b>	Eine Bezeichnung für die weitere serverseitige Verarbeitung.
<b>rows</b>	Anzahl der Zeilen (bestimmt damit auch die Höhe)
<b>cols</b>	Anzahl der Spalten bzw. die Anzahl der Zeichen pro Zeile

```
<form action="feedback.php" method="post" name="FB1" id="FB1">
  <h1>Rückmeldung</h1>
  <label for="TF1">Haben Sie noch weitere Vorschläge?</label><br>
  <textarea name="TF1" id="TF1"
    rows="5" cols="50">Ihr Feedback...</textarea>
</form>
```



Weitere Attribute sind:

<b>wrap</b>	Bestimmt den Textumbruch. Mögliche Werte sind: <b>hard</b> oder <b>soft</b> .
<b>readonly</b>	Das Eingabefeld kann nur gelesen werden. Eine Eingabe ist also nicht möglich. <b>ohne Wert</b>
<b>required</b>	Eine Eingabe muss erfolgen um das Formular absenden zu können. <b>ohne Wert</b>
<b>placeholder</b>	Es kann ein Text bestimmt werden, welcher in der Textarea als Hinweis erscheint.
<b>autofocus</b>	Legt den Focus nach dem Laden der Seite auf das Textfeld. <b>ohne Wert</b>
<b>maxlength</b>	Maximale Länge der Eingabe (Anzahl der maximalen Buchstaben).

```
<textarea name="tear2" id="tear2"
  rows="4" cols="60" placeholder="Ihr Feedback ..."
  required autofocus maxlength="220" ></textarea>
```



Da das `<textarea>` Element kompromisslos alle Sonderzeichen, Leerzeichen und Zeilenschaltungen darstellt, kann es auch für die Darstellung von HTML Code verwendet werden.

```
<textarea rows="4" cols="100" readonly >
  <h1>Technisches Service</h1>
  <p>Hier finden Sie alle ...</p>
</textarea>
```



Das gleiche Ergebnis innerhalb eines `<pre>` Tags sieht dann so aus:

```
<pre>
  &lt;h1&gt;Technisches Service&lt;/h1&gt;
  &lt;p&gt;Hier finden Sie alle ...&lt;/p&gt;
</pre>
```

# Lernhandout 12.3 Input

Referenzcode **HCL123**

Technologien HTML

Dateinamen L12\_3\_Input.odt, L12\_3\_Input.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... nutzen <input> Elemente für einzeilige Eingaben.	I	II
... bestimmen mit dem type Attribut die Art des <input> Elements.	I	II
... setzen einen eindeutigen Namen für das <input> Element, welcher für die spätere Verarbeitung in einer Script- oder Programmiersprache notwendig ist.	I	II
... beschriften <input> Elemente mit dem <label> Tag.	I	II
... kennen weitere generelle Attribute (size, readonly, step usw).	I	II
... kennen Attribute zur Validierung und Gestaltung.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

Mit dem `<input>` Element können einzeilige Eingaben getätigt werden. Es ist sehr umfangreich und wird durch das `type` Attribut näher bestimmt. Das `<input>` Element ist ein Standalone-Tag und muss also nicht geschlossen werden.

## HTML



```
<input type="" name="" value="">
```

<b>type</b>	Beschreibt die Art des Eingabefelds. Als Standard wird <code>type="text"</code> angenommen.
<b>name</b>	Der Name des Eingabefelds wird für die spätere Verarbeitung in einer Script- oder Programmiersprache verwendet.
<b>value</b>	Ein vordefinierter Wert.

```
<form>
  <label for="Vorname" >Ihren Vorname?</label><br>
  <input type="text" name="Vorname" id="Vorname" value="Kevin">
</form>
```



## Weitere generelle Attribute

<b>size</b>	Definiert die Anzahl der Zeichen, die eingegeben werden können. z. B. bei <code>size="15"</code> hat das Feld eine Breite von 15 Zeichen.
<b>readonly</b>	Das Eingabefeld kann nur gelesen werden. Eine Eingabe ist nicht möglich. <b>ohne Wert</b>
<b>min</b>	Der Minimum-Wert bei numerischen oder Datumseingabefeldern. z. B. <code>min="200"</code> bedeutet eine Eingabe von mehr als 200.
<b>max</b>	Der Maximum-Wert bei numerischen oder Datumseingabefeldern. z. B. <code>max="1300"</code> bedeutet eine Eingabe von weniger als 1300.
<b>step</b>	Die Schrittweite der Eingabe bei numerischen oder bei Datumfeldern. <code>step="any"</code> hebt die Schrittweite auf.
<b>multiple</b>	Erlaubt Mehrfacheingaben für Datei-Uploads oder Email-Eingaben. <b>ohne Wert</b>



## Attribute zur Validierung und Gestaltung

<b>minlength</b>	Mindestlänge der Eingabe (in Zeichen) z. B. <code>minlength="5"</code> verlangt mindestens fünf Zeichen.
<b>maxlength</b>	Maximallänge der Eingabe (in Zeichen).
<b>required</b>	Pflichtfeld. Eine Eingabe muss erfolgen. <b>ohne Wert</b>
<b>placeholder</b>	Sichtbarer Hinweis im Inputfeld. Es verschwindet nach der Eingabe eines Zeichens.
<b>autofocus</b>	Nach dem Laden der Webseite, wird er Focus auf das Inputfeld gelegt. <b>ohne Wert</b>
<b>autocomplete</b>	Automatisches Vervollständigen von Formularfeldern durch den Browser. Als Werte sind <code>on</code> oder <code>off</code> möglich.

```
<input type="text" name="Nachname"
  size="20" maxlength="20" minlength="3"
  placeholder="Bitte Nachnamen eingeben!" required
  autocomplete="off" autofocus >
```

# Übungsblatt 12.3 Input

Referenzcode **HCU123**

Technologien HTML, CSS

Dateinamen U12\_3\_Input.odt, U12\_3\_Input.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... recherchieren im Internet über den Unterschied zwischen get und post.	I	III
... stellen Überlegungen zur Verwendung von get und post an.	IK	III
... dokumentieren ihre Ergebnisse der Recherche und Überlegungen.	I	II
<b>Übung B</b>		
... scripten ein HTML Kommentar Formular.	I	II
... verwenden ein <form> Element mit den Attributen name, id, action, method.	I	II
... beschriften das <form> Element mit dem <label> Tag.	I	II
... fügen ein <input> Feld für einzeilige Texteingaben hinzu.	I	II
... setzen die notwendigen Attribute (z. B. placeholder) für das <input> Feld.	I	II
... fügen eine Textarea hinzu und definieren dafür die passenden Attribute (z. B. required).	I	II
... beschriften die Textarea mit dem <label> Tag.	I	II
... führen alle Notwendigen CSS Anpassungen durch (für Beschriftungen und Textfelder).	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



### Übung A: get und post

- Recherchiere im Internet den Unterschied zwischen `get` und `post` als Methode für ein Formular.
- Wie funktioniert `get`? Wie funktioniert `post`?
- Welche Vor- und Nachteile haben `get` und `post`?
- **Finde jeweils mind. drei Beispiele wo `get` bzw. `post` Sinn ergibt.**



### Übung B: Kommentar-Formular

- Erstelle ein neues HTML-Dokument mit einem Formular für Kommentare. Das Formular hat ein einzeliges Texteingabefeld für den Vor- und Nachnamen und ein mehrzeiliges Textfeld für den Kommentar.
- **<form> Eigenschaften:**  
Das Formular wird im `comment.php` Script verarbeitet. Eine passende Methode ist selbst zu wählen. Beschrifte das Formular (`<label>`) mit **"Ihr Feedback"**
- **Vor- und Nachnamen (einzeilige Texteingabe)**  
Die Attribute für den Namen und Größe sind selbst zu wählen. Es ist ein Pflichtfeld mit einem Platzhaltertext **"Hier bitte den Namen eingeben"**. Die Beschriftung lautet: **"Vor- und Nachname"**.
- **Kommentarfeld (mehrzeilige Texteingabe)**  
Die Attribute für den Namen und Größe sind selbst zu wählen. Die maximale Länge der Eingabe ist auf 600 Zeichen beschränkt. Es ist ein Pflichtfeld. Die Beschriftung lautet: **"Kommentar"**.
- CSS Anpassung für die Beschriftungen:  
Schriftart: Arial, Kursiv.
- CSS Anpassung für die Textfelder:  
**Schriftart: Courier New | Hintergrundfarbe: hellblau | Rahmen: 2 Pixel Dunkelgrau. | Wenn die Maus über das Textfeld fährt, soll sich die Hintergrundfarbe in weiß ändern**

#### Ihr Feedback

*Vor- und Nachname*

*Kommentar*

# Lernhandout 12.4 Input Typen

Referenzcode **HCL124**

Technologien HTML

Dateinamen L12\_4\_Inputtypes.odt, L12\_4\_Inputtypes.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... erstellen Eingabefelder für einzeiligen Text.	I	II
... erstellen Eingabefelder für Passworteingaben.	I	II
... erstellen Eingabefelder für eMail Adressen.	I	II
... erstellen Eingabefelder für Zahlen.	I	II
... kennen weitere Attribute für Zahleneingabefelder.	I	I
... erstellen Eingabefelder für Zahlen mit einem Schieberegler.	I	II
... kennen weitere Attribute für Eingabefelder mit Schieberegler.	I	I
... erstellen Eingabefelder für Telefonnummern.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

### Texteingabe

**HTML**

```
<input type="text">
```

### Passwörter

**HTML**

```
<input type="password">
```



Im Eingabefeld werden Platzhalter anstatt der Zeichen angezeigt.  
(Browserabhängig, meist aber Sterne \*)

### eMail-Adressen

**HTML**

```
<input type="email">
```



Eine Eingabe ist nur valide, wenn auch ein @ Zeichen darin vorkommt.

### Web-Adressen

**HTML**

```
<input type="url">
```



Validiert die URL. Eine vollständige Eingabe mit http ist notwendig:  
(z. B. http://www.css4.at) Mobile Geräte blenden z. B. .com automatisch in der  
Tastatur ein.



*Für die Typen text, password, email und url gibt es das Attribut inputmode.  
Es hilft den mobilen Geräten die passende Tastatur zu wählen.  
Werte sind z. B. verbatim (alphanumerisch), latin (Texteingabe mit  
Texteingabehilfe) usw.*

### Eingabe von Zahlen

**HTML**

```
<input type="number">
```



Es können nur Zahlen eingegeben werden.  
Weitere Attribute sind **max**, **min**, **step** und **value**.

### Eingabe von Zahlen mit einem Schieberegler

**HTML**

```
<input type="range">
```



Der Schieberegler (Slider) wird für die Eingabe von Zahlen verwendet.  
Weitere Attribute sind **max**, **min**, **step** und **value**.

Mit dem **orient** Attribut kann man den Slider auch vertikal darstellen.  
z. B. `<input type="range" orient="vertical">`

### Telefonnummern

**HTML**

```
<input type="tel">
```



Mobile Geräte blenden eine angepasste Tastatur ein.

# Übungsblatt 12.4 Input Typen

Referenzcode **HCU124**

Technologien HTML, CSS

Dateinamen U12\_4\_Inputtypes.odt, U12\_4\_Inputtypes.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... erstellen ein typisches Anmeldeformular.	<b>I</b>	<b>II</b>
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	<b>II</b>
<b>Übung B</b>		
... erstellen ein Formular zur Berechnung von Sparzinsen.	<b>I</b>	<b>II</b>
... benutzen Eingabefelder für Zahlen und passe diese mit den notwendigen Attributen an.	<b>I</b>	<b>II</b>
... fügen ein Eingabefeld mit Schieberegler ein und passen dieses mit den notwendigen Attributen an.	<b>I</b>	<b>II</b>
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	<b>II</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



### Übung A: Anmeldeformular

- Erstelle ein typisches Anmeldeformular mit den Feldern eMail, Benutzername und Passwort.
- Beschrifte alle Felder mit dem `<label>` Element.
- **Verschönere das Aussehen mit CSS. Freie kreative Übung.**

eMail-Adresse  
max.mustermann@johndoe.at

Benutzername  
madmax96

Passwort  
●●●●●●●●



### Übung B: Online Sparen

- Erstelle ein Formular zur Berechnung von Sparzinsen für ein Online-Sparbuch.
- Zwei Eingabefelder
- Feld 1: Kapital  
Nummernfeld, mindestens 1000 und maximal 50000. Schrittweite 500. Beschriftung durch ein `<label>` Element.
- Feld 2: Laufzeit in Jahren  
Schieberegler, mindestens 1 und maximal 8, Schrittweite 1. Beschriftung durch ein `<label>` Element.
- **Verändere das Aussehen mit CSS.**

Kapital  
15000

Laufzeit in Jahren  
1 2 3 4 5 6 7 8

# Lernhandout 12.5 Form Button

Referenzcode **HCL125**

Technologien HTML

Dateinamen U12\_5\_Buttons.odt, U12\_5\_Buttons.pdf, Report\_GET.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... kennen die Funktionen von Buttons in einem Formular.	I	I
... verwenden <button> Elemente mit den Type Attributen button, submit und reset.	I	II
... kennen und nutzen das <button> Attribut: form.	I	II
... kennen und nutzen das <button> Attribut: formaction.	I	II
... kennen und nutzen das <button> Attribut: formmethod.	I	II
... kennen und nutzen das <button> Attribut: formenctype.	I	II
... kennen und nutzen das <button> Attribut: name.	I	II
... kennen und nutzen das <button> Attribut: value.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Im zweiten Codebeispiel wird das <button> Element mit einer Formmethode GET ausgestattet. Klickt man darauf, so öffnet sich eine PHP Datei, die nicht vorhanden ist. Dennoch wird in der URL die Übermittlung angezeigt.
- Die Webanwendung **Report\_GET.html** (downloadbar über [www.css4.at](http://www.css4.at)) liest die Wertübergabe aus der URL aus und stellt diese in einer Tabelle dar. Die Datei kann nur Wertübergaben mittels GET auslesen. Dahinter steckt ein JavaScript!

Verwendung:

```
<button form="meinForm" formaction="Report_GET.html"  
      formmethod="get" type="submit">Standard</button>
```

Für Formulare gibt es das `<button>` Element, welches anklickbare Schaltflächen erzeugt. Diese können ein Formular absenden, das Formular zurücksetzen oder eine clientseitige Aktion/Funktion (z. B. für JavaScript) auslösen.

## HTML



```
<button type="" > ... </button>
```

Zwischen dem Start und Endtag wird die Beschriftung des Buttons eingetragen. Das Aussehen kann selbstverständlich mit CSS angepasst werden.

Für das `type` Attribut gibt es folgende Werte:

<code>button</code>	Auslösen einer clientseitigen Aktion
<code>submit</code>	Absenden eines Formulars, Standardwert
<code>reset</code>	Zurücksetzen des Formulars.

Passwort eingeben:

```
<form>
  <label for="eingabetext">Passwort eingeben:</label>
  <input type="password" name="eingabetext" id="eingabetext">
  <button type="button">Komplexität prüfen</button>
  <button type="reset">Zurücksetzen</button>
  <button type="submit">Absenden</button>
</form>
```



## Weitere Attribute:

<code>form</code>	Wenn der Button außerhalb des Formulars ist, kann mit dem <code>form</code> Attribut der Name des Formulars angegeben werden.
<code>formaction</code>	Im Button kann die Aktion (URL) des Formulars festgelegt werden. Sollte es im <code>&lt;form&gt;</code> Element eine abweichende Methode geben, dann gilt die des <code>&lt;buttons&gt;</code>
<code>formmethod</code>	Im Button kann die Methode ( <code>Get</code> , <code>Post</code> ) definiert werden. Sollte es im <code>&lt;form&gt;</code> Element eine abweichende Methode geben, dann gilt die des <code>&lt;buttons&gt;</code>
<code>formenctype</code>	Codierung über den Button
<code>name</code>	Zur weiteren Verwendung in einem Script
<code>value</code>	Rückgabewert eines Buttons.

```
<form name="meinForm" id="meinForm">
  <input type="text" name="Feld01" id="Feld01">
</form>

<button form="meinForm" formaction="standard.php"
  formmethod="get" type="submit">Standard</button>
<button form="meinForm" formaction="premium.php"
  formmethod="post" type="submit">Premium</button>
```



Mit dem Attribut `formnovalidate` kann die Überprüfung von Pflichtfeldern (mit `require` gekennzeichnet) unterbunden werden. Es ist analog zum `novalidate` Attribut im `<form>` Element.

# Übungsblatt 12.5 Formular Buttons

Referenzcode **HCU125**

Technologien HTML, CSS

Dateinamen U12\_5\_Buttons.odt, U12\_5\_Buttons.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... erstellen ein Formular nach Vorgaben.	<b>I</b>	<b>II</b>
... entscheiden sich für oder gegen ein Tabellen-Layout.	<b>K</b>	<b>III</b>
... reproduzieren ein CSS Layout nach einer bebilderten Vorlage.	<b>I</b>	<b>II</b>
... definieren Pflichtfelder.	<b>I</b>	<b>II</b>
... fügen Buttons für das Absenden und Zurücksetzen hinzu.	<b>I</b>	<b>II</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Der Passierschein A38 ist aus dem Asterix-Abenteuer "Asterix erobert Rom" entnommen.



### Übung A: Passierschein A38

- Erstelle ein Formular wie unten dargestellt.
- Wähle zwischen zwei Layoutlösungen für das Formular:
  - mit Tabelle (einfach, aber mehr Schreibarbeit)
  - ohne Tabelle (knifflig, aber dafür weniger Schreibarbeit)
- Versuche mit CSS das gleiche Erscheinungsbild wie unten zu erzeugen.
- Die Felder Vorname, Nachname, Telefon und eMail sind Pflichtfelder.
- **Zwei Buttons zum Absenden und zum Zurücksetzen des Formulars.**

## Amtliches Antragsformular für den Passierschein A38

Alle mit einem \* gekennzeichneten Feldern müssen ausgefüllt werden!

Titel:

Vorname:  \*

Nachname:  \*

Straße:

Postleitzahl:

Wohnort:

Telefon:  \*

eMail:  \*

Website:

Passierschein A38 beantragen

Zurücksetzen

# Lernhandout 12.6 Dialogeingaben

Referenzcode **HCL126**

Technologien HTML

Dateinamen L12\_6\_Dialoge.odt, L12\_6\_Dialoge.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... erstellen ein Input-Feld für einen Datei-Upload.	I	II
... kennen und nutzen die Attribute für Datei-Upload-Dialoge.	I	II
... erstellen ein Input-Feld für die Farbauswahl.	I	II
... erstellen ein Input-Feld für Datum und Uhrzeit.	I	II
... erstellen ein Input-Feld für versteckte Elemente.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die Dialoge werden von Browser zu Browser unterschiedlich dargestellt.
- Ein verstecktes Formelement wird verwendet um "unsichtbar" Werte zu übergeben. Ein Beispiel dafür: Mit JavaScript wird ein Value-Wert bestimmt, damit es später mit PHP ausgewertet wird. Damit kann man z. B. eine Session simulieren.

## Datei-Upload

## HTML



```
<input type="file" >
```

Der Datei-Upload funktioniert nur mit der `post`-Methode (im `<form>` oder im `<button>`). Zusätzlich muss im `<form>` Element noch das Attribut `enctype="multipart/form-data"` hinzugefügt werden, denn sonst wird nur der Name der Datei übertragen. Weitere Attribute:

**accept** Definiert die möglichen Dateitypen (MIME-Typen) für den Upload.

z. B. nur Textdateien:

```
<input type="file" accept="text/*">
```

nur JPG Bilder:

```
<input type="file" accept="image/jpeg">
```

**multiple** Mehrfachauswahl **ohne Wert**

```
<form action="upload.php" enctype="multipart/form-data" method="post">
  <label for="datei" >Bilder hochladen</label>
  <input type="file" accept="image/*" name="datei" id="datei" multiple>
</form>
```

## Farbauswahl

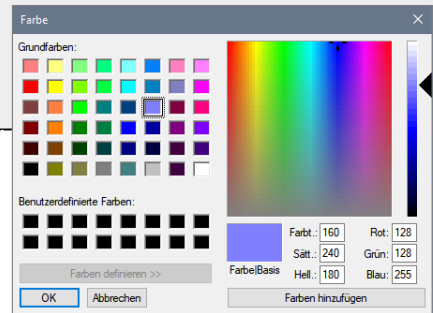
## HTML



```
<input type="color" >
```

Der Rückgabewert der ausgewählten Farbe ist ein HEX-Wert.

```
<form method="get">
  <label for="farbe" >Lieblingsfarbe</label>
  <input type="color" value="#ffffff"
    id="farbe" name="farbe" >
</form>
```



## Datum und Uhrzeit

## HTML



```
<input type="date" >
```

Mit dem `type="date"` Attribut werden Datumseingaben abgefragt. In der Regel stellt der Browser ein Datumsfeld oder ein Kalender-Dialog zur Verfügung.

Artverwandt sind:

Wochenangaben: `<input type="week" >`

Monatsangaben: `<input type="month" >`

Uhrzeiten: `<input type="time" >`

Kombination: `<input type="datetime-local" >`

## Versteckte Elemente

## HTML



```
<input type="hidden" value="" >
```

Das Eingabefeld ist für den Benutzer unsichtbar. Der Wert wird aber beim Absenden übergeben.

# Übungsblatt 12.6 Dialoge

Referenzcode: **HCU126**

Technologien: HTML, CSS

Dateinamen: U12\_6\_Dialog.odt, U12\_6\_Dialog.pdf, Report\_GET.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... fügen ein Eingabefeld für eine Farbwahl in ein HTML Dokument ein.	I	II
... beschriften das Eingabefeld.	I	II
... testen das Eingabefeld mit beigelegter (Report_GET.html) JavaScript Applikation.	I	II
<b>Übung B</b>		
... erstellen ein Formular für eine Online-Bewerbung.	I	II
... finden die passenden Input-Felder für ein klassisches Online-Bewerbungsformular.	KI	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Die **Report\_GET.html** Datei übernimmt Formulardaten und wertet diese mit JavaScript aus. Also wird für das action-Attribut: **Report\_Get.html** und für die Methode: GET angegeben.



### Übung A: Farbauswahl

- Erstelle ein neues HTML Dokument.
- Füge ein Eingabefeld für Farben hinzu.
- Beschrifte das Eingabefeld mit "Farbe der Mannschaft". `<label>`
- **Verwende als Methode GET mit dem Report\_GET.html über den Submit-Button und nicht als Attribut im `<form>` Element.**



### Übung B: Online Bewerbung

- Erstelle ein neues HTML Dokument mit einem Formular zur Online Bewerbung.
- Folgende Eingabefelder sollten eingebunden sein:  
Vor- und Nachname, E-Mail-Adresse, Geburtsdatum, Dateiupload für den Lebenslauf (nur PDF) und ein mehrzeiliges Textfeld für das Motivationsschreiben.
- **Gestalte das Formular mit CSS anspruchsvoll. Freie kreative Übung.**

The screenshot shows a web browser window with the title 'Online Bewerbung'. The page content includes a heading 'Online Bewerbung', a sub-heading 'Ihre Angaben', and several form fields: 'Vor- und Nachname' (filled with 'Max Mustermann'), 'eMail-Adresse' (filled with 'max.mustermann@gmail.com'), 'Geburtsdatum' (filled with 'TT . MM . JJJJ'), 'Lebenslauf hochladen (nur PDF)' (with a 'Durchsuchen...' button and the text 'Keine Datei ausgewählt.'), and a 'Motivationsschreiben' text area. A 'Bewerbung absenden' button is at the bottom.

# Lernhandout 12.7 Auswahleingaben

Referenzcode **HCL127**

Technologien HTML

Dateinamen L12\_7\_Auswahl.odt, L12\_7\_Auswahl.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... binden Radio-Buttons in ein HTML Dokument ein.	<b>I</b>	<b>II</b>
... kennen die Attribute von Radio-Buttons.	<b>I</b>	<b>I</b>
... binden Checkbox Felder (Kontrollfelder) in ein HTML Dokument ein.	<b>I</b>	<b>II</b>
... kennen die Attribute von Checkbox-Feldern.	<b>I</b>	<b>I</b>
... stellen Überlegungen zu Radio-Buttons und Checkbox-Feldern an und analysieren, wann die jeweilige Verwendung Sinn ergibt.	<b>IK</b>	<b>III</b>
... definieren ein <fieldset> für Formularelemente.	<b>I</b>	<b>II</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das <fieldset> Element hat eher nur eine optische Funktion. Bei einer Übergabe an ein PHP Script spielt es "fast" keine Rolle. Ebenso für eine Auswertung mit JavaScript bringt es keine klaren Vorteile.

## Radio-Buttons

## HTML



```
<input type="radio" name="" value="">
```

Radio-Buttons sind eine Gruppe von Auswahlbuttons in der nur eine Eingabe erlaubt ist. Wird also ein Button angeklickt, wird ein anderer abgewählt. Folgende Attribute sind möglich:

**name** Alle Radio-Buttons mit dem gleichen **name** Attributwert, gehören zu einer Gruppe.

**value** Definiert den Wert, der beim Absenden des Formulars übergeben wird wenn das Element aktiviert wurde.

**checked** Ein bestimmter Radio-Button wird vorselektiert. **ohne Wert**

**disabled** Zum Deaktivieren. Es wird ausgegraut dargestellt. **ohne Wert**

```
<form>
  <input type="radio" name="Zahlung" value="Nachnahme" id="nn">
  <label for="nn"> per Nachnahme </label>
  <input type="radio" name="Zahlung" value="Vorauskassa" id="vk">
  <label for="vk"> per Vorauskassa </label>
</form>
```

## Checkbox

## HTML



```
<input type="checkbox" name="">
```

Eine Checkbox ist ein anklickbares Kontrollfeld. Per Klick erscheint in der Regel ein Häkchen bei den ausgewählten Feldern.

**name** Jedes Checkbox Element sollte einen eigenen Namen bekommen.

**checked** Zum Vorselektieren von Checkboxes. **ohne Wert**

**disabled** Zum Deaktivieren. Es wird ausgegraut dargestellt. **ohne Wert**

```
<form method="get">
  <input type="checkbox" name="abo1" id="gabo">
  <label for="gabo"> Gratis Ausgabe bestellen</label><br>
  <input type="checkbox" name="newsletter" id="nwsl" checked>
  <label for="nwsl"> Newsletter bestellen</label>
  <button type="submit">Absenden</button>
</form>
```

## Fieldset

## HTML



```
<fieldset> <legend> ... </legend> ... </fieldset>
```

Mit dem `<fieldset>` Element kann man Formelemente **innerhalb** eines Formulars gruppieren. Es sind die Attribute **name**, **disabled** und **form** verfügbar. Das `<legend>` Element wird zur Beschriftung des `<fieldset>` verwendet.

```
<fieldset name="Verpackung">
  <legend>Welche Verpackung möchten Sie?</legend>
  <input type="checkbox" name="Oeko"> Öko <br>
  <input type="checkbox" name="Premium"> Premium
</fieldset>
```

# Übungsblatt 12.7 Auswahleingaben

Referenzcode: **HCU127**

Technologien: HTML, CSS

Dateinamen: U12\_7\_Auswahl.odt, U12\_7\_Auswahl.pdf, Report\_GET.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen ein Bestellformular für einen Lieferservice.	<b>I</b>	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II
... fügen Auswahleingaben (Radio, Checkboxen) hinzu.	<b>I</b>	II
... beschriften alle Felder und fügen ein <fieldset> hinzu.	<b>I</b>	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II
... testen das HTML Dokument mit der Report_GET.html	<b>I</b>	II
<b>Übung B</b>		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	<b>S</b>	II
... optimieren das HTML Dokument für die Darstellung auf Smartphones.	<b>I</b>	II
... testen die Arbeit auf einem Smartphone.	<b>I</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Das Salat-Beispiel soll auch ein Beitrag zur Gesundheitsbildung darstellen. Wünschenswert wäre also, dass die Kalorienangaben der Zutaten und Dressings der Realität entsprechen - für das Erlernen von HTML und CSS ist es aber nicht zwingend notwendig, deshalb können auch fiktive Werte angenommen werden. Viel interessanter jedoch ist die Reflexion auf JavaScript --> z. B. eine Automatische Ermittlung der Gesamtkalorien und natürlich auch des Preises, uvm.



### Übung A: Salat Online

- Erstelle ein Bestellformular für einen Salat Lieferbetrieb.
- Auswahl zwischen einem großen und einem kleinen Salat.
- Mindestens 12 Zutaten mit Kalorienangaben (Mehrfachauswahl).
- Mindestens 4 Dressings mit Kalorienangaben (nur eine Auswahl möglich).
- Verwende `<fieldset>` Elemente zur optischen Trennung.
- Beschrifte alle Felder mit `<label>`
- Gestalte die Seite mit CSS – freie kreative Übung.
- **Teste die Seite mit der Report\_GET.html**



### Übung B: Salat Online Mobil

- Öffne deine Lösung von **Übung 12.7 A: Salat Online.**
- Optimierte die Seite für mobile Geräte (Smartphone, Tablet usw.)
- **Teste die Seite auf einem Smartphone!**

# Lernhandout 12.8 Auswahllisten

Referenzcode **HCL128**

Technologien HTML

Dateinamen L12\_8\_Listen.odt, L12\_8\_Listen.pdf, Report\_GET.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... erstellen Auswahllisten (Drop-Down-Listen) mit dem <code>&lt;select&gt;</code> Element.	I	II
... erstellen Auswahllistenpunkte mit dem <code>&lt;option&gt;</code> Tag.	I	II
... kennen die Attribute von <code>&lt;select&gt;</code> und <code>&lt;option&gt;</code> .	I	I
... begrenzen Auswahllisten durch die Anzahl der dargestellten Zeilen.	I	II
... selektieren einen Auswahlpunkt vor.	I	II
... stellen Überlegungen zur Mehrfachauswahl (multiple) an.	K	III

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Wenn das `<option>` Element kein Value hat, wird der Name des `<select>` mit dem Inhalt vom ausgewählten `<option>` Element übermittelt. Interessant ist hier das Leerzeichen bei: `Report_GET.html?philos=Immanuel+Kant`, welches mit einem Plus-Zeichen übermittelt wird. Weil Leerzeichen in einer URL nicht zulässig sind wird die Zeichenfolge `%20` bzw. ein Plus-Zeichen (im Query-String) verwendet.

## Drop-Down-Liste

## HTML



```
<select name="" > <option> ... </option> </select>
```

Das `<select>` Element stellt eine Auswahlliste, bzw. ein Drop-Down-Feld zur Verfügung. Innerhalb des `<select>` Elements werden über den `<option>` Tag die Einträge festgelegt.

**Attribute für das `<select>` Element.**

<b>name</b>	Name des Elements (Variablenname).
<b>size</b>	Anzahl der sichtbaren Zeilen. Ohne Size Attribut wird ein Drop-Down-Feld angezeigt.
<b>multiple</b>	Ermöglicht eine Mehrfachauswahl <b>ohne Wert</b>
<b>required</b>	Die Eingabe ist erforderlich (Pflichtfeld) <b>ohne Wert</b>

**Attribute für das `<option>` Element.**

<b>value</b>	Übergabewert, wenn es angeklickt wird. Ohne Wert, wird der Inhalt des <code>&lt;option&gt;</code> Tag übermittelt.
<b>selected</b>	Eine Option wird vorausgewählt. <b>ohne Wert</b>

```
<form action="Report_GET.html" method="get">
  <label for="philos">Berühmte Philosophen:</label><br>
  <select name="philos" id="philos">
    <option>Friedrich Nietzsche</option>
    <option>Ludwig Wittgenstein</option>
    <option>Immanuel Kant</option>
    <option>David Hume</option>
  </select>
  <button type="submit">Absenden</button>
</form>
```

## Auswahlliste (sichtbare Auswahl)



Hier eine Auswahlliste mit drei Zeilen und `value` Attribute für das `<option>` Element. Das Auswahlfeld `value="2typ"` wurde vorselektiert.

```
<select name="philos" id="philos" size="3">
  <option value="1typ">Friedrich Nietzsche</option>
  <option value="2typ" selected>Ludwig Wittgenstein</option>
  <option value="3typ">Immanuel Kant</option>
  <option value="4typ">David Hume</option>
</select>
```



Eine Mehrfachauswahl (mit dem `multiple` Attribut) ist für den Anwender nicht immer klar ersichtlich (z. B. mit gedrückter STRG-Taste selektieren). Checkboxes eignen sich besser für eine Mehrfachauswahl.

# Übungsblatt 12.8 Auswahllisten

Referenzcode: **HCU128**

Technologien: HTML, CSS

Dateinamen: U12\_8\_Listen.odt, U12\_8\_Listen.pdf, Report\_GET.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen eine HTML Seite nach einer bebilderten Vorlage.	I	II
... fügen vier Auswahllisten hinzu.	I	II
... nutzen <fieldset> Elemente zur Beschriftung.	I	II
... fügen ein Drop-Down-Feld hinzu.	I	II
<b>Übung B</b>		
... erstellen eine Webseite für ein Gewinnspiel.	I	II
... fügen Input-Felder und Auswahllisten getreu der Vorgaben hinzu.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Mit clever gesetzten <label> Beschriftungen kann ein Klick auf eine Antwort das Auswahlfeld aktivieren.
- Mit einem Gewinnspiel kann man relative Günstig Daten von Kunden beziehen, oder sie zu einem Abo eines Newsletter anleiten. Dabei sind aber die gesetzlichen Bestimmungen der Datenschutzverordnung zu beachten.
- Es gehört zum guten Ton im Web Newsletter Abonnements **nicht** vorzuselektieren.
- Man kann auch das Thema "Glückspiel und Spielsucht" mit ihren Auswirkungen und Problemen ansprechen.
- Ab diesen zwei Übungen sind auch die essentiellsten Basics von HTML vermittelt worden (Fundamentum). Die Studierenden sollten fähig sein, selbstständig weitere Themen aufzuarbeiten. Man kann mit JavaScript oder PHP beginnen oder man vertieft weiter HTML und CSS-Kenntnisse. Einige Feinheiten folgen noch!



### Übung A: Softwareliste

- Erstelle ein Formular wie unten dargestellt.
- Vier Auswahllisten.
- Fieldsets als Beschriftung
- **Ein Drop-Down-Feld (Ausgabe: HTML, XML, Plain Text).**

<p>Betriebssystem</p> <ul style="list-style-type: none"> <li>Windows</li> <li><b>OSx</b></li> <li>Linux</li> <li>Unix</li> <li>Anderes</li> </ul>	<p>Browser</p> <ul style="list-style-type: none"> <li>Edge</li> <li>Firefox</li> <li>Chrome</li> <li><b>Safari</b></li> <li>Anderer</li> </ul>	<p>Suchmaschine</p> <ul style="list-style-type: none"> <li>Google</li> <li><b>Yahoo</b></li> <li>Web.de</li> <li>Bing</li> <li>Anderer</li> </ul>	<p>Coding</p> <ul style="list-style-type: none"> <li>C#</li> <li>Visual Basic</li> <li>Java</li> <li><b>Pascal</b></li> <li>Anderer</li> </ul>
---	--	---	--

Ausgabe:



### Übung B: Gewinnspiel

- Erstelle eine Webseite für ein Gewinnspiel
- Mindestens drei Radio-Buttons
- Drei Textfelder für den Namen, eMail-Adresse und Telefon.
- Ein Drop-Down-Feld
- Mindestens zwei Checkboxes.
- **Freie kreative Übung.**

# Lernhandout 13.1 Animationen

Referenzcode **HCL131**

Technologien CSS

Dateinamen L13\_1\_Animationen.odt, L13\_1\_Animationen.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verstehen die Idee der Animation in CSS.	I	III
... definieren @keyframes Regeln mit Start- und Endeigenschaften.	I	II
... deklarieren die Animation mit der animation-name Eigenschaft.	I	II
... verändern die Dauer der Animation mit der animation-duration Eigenschaft.	I	II
... verwandeln <div> Quadrate in Kreise mittels einer Animation.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Reine CSS Einheit - alle Beispiele lassen sich durch ein <div> im <body> darstellen!

Mit CSS ist es möglich, Animationen ganz ohne Flash oder JavaScript zu erstellen. Dabei werden die Start-Eigenschaften eines Elements bis zu den End-Eigenschaften verändert. Der Übergang verläuft flüssig. Für die Animation wird zuerst eine `@keyframe` Regel erstellt.

## CSS



```
@keyframes meineAnimation{ ... }
```

Im folgenden Beispiel ist `meineAnimation` der Name der Animation. Natürlich kann auch ein anderer Name gewählt werden. Innerhalb der `@keyframes` Regel wird die Start-Eigenschaft mit `from {...}` und die End-Eigenschaften mit `to {...}` festgelegt. Die Farbe ändert sich von Blau nach Rot!

```
@keyframes meineAnimation {
  from {background-color:blue;}
  to   {background-color:red;}
}
```

Nachdem die Animation über die `@keyframes` Regel erstellt wurde, muss sie über einen Selektor (ID, Class, usw) dem Element zugewiesen werden.

Dafür wird mit `animation-name` der Name der Animation angegeben und mit `animation-duration` die Dauer der Animation festgelegt. Nach der Animation nimmt das Element wieder seinen originalen Style an.

## CSS



```
animation-name: [Name];
```

Für `[Name]` wird der Namen der `@keyframes` Regel eingetragen. In diesem Beispiel: `animation-name: meineAnimation;`

## SS



```
animation-duration: [Value];
```

Für `[Value]` wird die Dauer der Animation in Sekunden eingetragen. In diesem Beispiel vier Sekunden: `animation-duration: 4s;`

```
div {width: 500px; height: 500px;
  background-color:red;
  animation-name: meineAnimation;
  animation-duration: 4s;}
```



Das Beispiel verwandelt alle `<div>` Elemente vom Quadrat in einen Kreis. Dauer der Animation ist 5 Sekunden.

```
@keyframes rundeEcken {
  from {border-radius:0px;}
  to   {border-radius: 500px;}
}

div {width: 200px; height: 200px; margin:auto;
  border: 20px solid red; border-radius: 500px;
  animation-name: rundeEcken;
  animation-duration: 5s;}
```

# Lernhandout 13.2 Animationen

---

Referenzcode **HCL132**

Technologien CSS

Dateinamen L13\_2\_Animationen.odt, L13\_2\_Animationen.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... definieren Animationsschritte über Prozentangaben in der @keyframes Regel.	I	II
... verzögern den Animationsstart mit der CSS Eigenschaft animation-delay.	I	II
... definieren Wiederholungen von Animationen mit der CSS Eigenschaft animation-iteration-count.	I	II
... nutzen die CSS Eigenschaft animation-play-state um den Zustand einer Animation festzulegen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

- Keine

Die `@keyframes` Regel ist nicht nur auf ein `from` und `to` beschränkt. Die einzelnen Animationschritte können auch mit Prozentangaben definiert werden. (von 0% bis 100%).

**CSS** `@keyframes animationsName { 0 - 100%{...} }`

```
@keyframes bewegung {
  0% {top: 0px; left: 0px; background: red;}
  25% {top: 0px; left: 200px; background: blue;}
  50% {top: 200px; left: 200px; background: yellow;}
  75% {top: 200px; left: 0px; background: green;}
  100% {top: 0px; left: 0px; background: red;}
}
```

### Verzögerung

**CSS** `animation-delay: [Wert];`



Die Eigenschaft bestimmt die Verzögerung des Animationsstarts.  
`animation-delay: 3s;` wartet drei Sekunden bis die Animation startet.

```
#dasDIV {width:200px; height:200px; background:green;
position:fixed;
animation-name:bewegung; animation-duration:3s;
animation-delay:5s;}
```

```
~~~~~
<div id="dasDIV">Warte 5 Sekunden</div>
```

### Wiederholungen

**CSS** `animation-iteration-count: [Wert];`



Die Eigenschaft bestimmt die Wiederholungen der Animation.  
`animation-iteration-count: 4;` wiederholt die Animation viermal.  
`animation-iteration-count: infinite;` wiederholt ohne Ende.

```
#dasDIV {width:200px; height:200px; background:green;
position:fixed;
animation-name:bewegung; animation-duration:3s;
animation-iteration-count:3;}
```

```
~~~~~
<div id="dasDIV">Drei Wiederholungen</div>
```

### Zustand

**CSS** `animation-play-state: ;`



Bestimmt ob eine Animation pausiert oder läuft.  
`animation-play-state: paused;` Die Animation pausiert.  
`animation-play-state: running;` Die Animation läuft (Standardwert)

```
#dasDIV:hover {animation-play-state:paused;}
```

```
~~~~~
<div id="dasDIV">Stoppe mich mit der Maus</div>
```

# Lernhandout 13.3 Animationen

Referenzcode **HCL133**

Technologien CSS

Dateinamen L13\_3\_Animationen.odt, L13\_3\_Animationen.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... wechseln bei jedem Animationsdurchgang seine Richtung mit der CSS Eigenschaft animation-direction	I	II
... definieren mit der CSS Eigenschaft animation-fill-mode den Zustand nach Beendigung einer Animation.	I	II
... legen mit einer Timinigfunktion die Dauer zwischen den Schlüsselbildern fest.	I	II
... fassen mehrere Animationseigenschaften in der CSS Eigenschaft animation zusammen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

## Richtung der Animation

## CSS



**animation-direction:** [Wert];

[Wert] Wechselt bei jedem Durchgang (Zyklus) seine Richtung.

**normal** Die Animation wird immer vorwärts abgespielt.

**alternate** Die Animation wechselt nach jedem Durchlauf die Richtung.

**reverse** Die Animation wird immer rückwärts abgespielt.

## Zustand der Animation

## CSS



**animation-fill-mode:** [Wert];

[Wert] Definiert den Zustand nach der Animation.

**forwards** Das Element übernimmt die Werte des letzten Animationsschrittes.

**backwards** Das Element übernimmt die Werte des ersten Animationsschrittes

**both** Verbindet die Werte von forwards und backwards.

## Dauer zwischen den Schlüsselbildern

## CSS



**animation-timing-function:** [Wert];

[Wert] Die Timingfunktion bezieht sich auf die Dauer zwischen den Schlüsselbildern. *ease*, *ease-in*, *ease-out*, *ease-in-out*, *linear*, *step-start*, *step-end*, *cubic-bezier()*, *steps()* usw.

## Shorthand Propertie animation (Kurzschreibweise)

## CSS



**animation:** ... ;

Die einfache CSS Eigenschaft **animation** fasst alle anderen animation-Eigenschaften zusammen.



*Das Beispiel lässt ein Element aus der Bildschirmmitte hereinfliegen!*

```
#meinDIV {position:fixed; background-color:#CCC;
  animation: ausZentrum 3s;
  animation-fill-mode: forwards;}
@keyframes ausZentrum {
  from {height:1%; width:1%; top:48%; right:48%;}
  to   {height:80%; width:80%; top:10%; right:10%;}
}
```

~~~~~  
<div id="meinDIV">Hinweis</div>

# Übungsblatt 13.3 Animationen

Referenzcode **HCU133**

Technologien HTML, CSS

Dateinamen U13\_3\_Animationen.odt, U13\_3\_Animationen.pdf

| <b>Feinziele</b>                                                                                         | <b>Zielart</b> | <b>Taxonomie</b> |
|----------------------------------------------------------------------------------------------------------|----------------|------------------|
| Die Studierenden ...                                                                                     |                |                  |
| <b>Übung A</b>                                                                                           |                |                  |
| ... erstellen ein animiertes Balkendiagramm.                                                             | I              | II               |
| ... animieren die Balken von links nach rechts.                                                          | I              | II               |
| ... definieren eine Animationsverzögerung.                                                               | I              | II               |
| ... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit. | A              | II               |
| <b>Übung B</b>                                                                                           |                |                  |
| ... ordnen vier Rechtecke (div) im Zentrum des Viewports an.                                             | I              | II               |
| ... weisen den Rechtecken (div) unterschiedliche CSS Eigenschaften zu (z. B. Farbe, usw.)                | I              | II               |
| ... animieren die Rechtecke (div) so, dass sie von der Ecke ins Zentrum des Viewports 'fliegen'.         | I              | II               |
| ... verändern die Größe der Rechtecke (div) mit CSS Animationen.                                         | I              | II               |
| ... weisen den Rechtecken (div) unterschiedliche Timing-Funktionen zu.                                   | I              | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



### Übung A: Abstimmungsergebnis

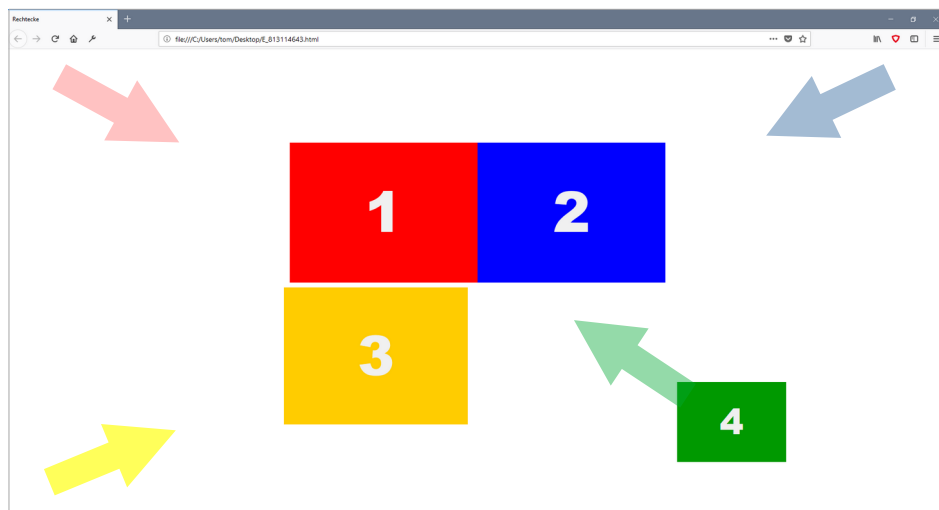
- Erstelle ein animiertes Balkendiagramm.
- Die Balken (Ja 45%, Nein 37%, Egal 18%) haben eine unterschiedliche Farbe
- Die Balken wachsen von links nach rechts.
- Die Balken erscheinen hintereinander (Verzögerung).
- **Verschönere das Diagramm mit CSS.**

### Abstimmungsergebnis



### Übung B: Rechtecke ins Zentrum

- Vier rechteckige Div's sollen im Zentrum angeordnet werden.
- Die Div's dürfen sich nicht überlappen.
- Jedes Div hat eine eigene unterschiedliche Farbe.
- Die Div' fliegen von der Ecke ins Zentrum
  - 1: von links oben
  - 2: von rechts oben
  - 3: von links unten
  - 4: von rechts unten
- Die Div's sollen auch in Höhe und Breite animiert werden (von klein nach groß).
- **Jedes Div hat eine unterschiedliche Timing-Funktion.**



# Lernhandout 13.4 Transformieren

Referenzcode **HCL134**

Technologien CSS

Dateinamen L13\_4\_Transformieren.odt, L13\_4\_Transformieren.pdf

| Feinziele                                                                                         | Zielart | Taxonomie |
|---------------------------------------------------------------------------------------------------|---------|-----------|
| Die Studierenden ...                                                                              |         |           |
| ... können mit der CSS Eigenschaft transform 2D und 3D Veränderungen an HTML Elementen vornehmen. | I       | II        |
| ... skalieren Elemente mit transform: scale(x,y).                                                 | I       | II        |
| ... erstellen ein Beispiel mit einer Unterstreichung von der Mitte heraus.                        | I       | II        |
| ... drehen Elemente mit transform: rotate(winkel).                                                | I       | II        |
| ... kennen die Winkelmaße grad, rad, turn.                                                        | I       | I         |
| ... simulieren einen analogen Sekundenzeiger einer Uhr.                                           | I       | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Zum Beispiel mit der animierten Unterstreichung: Sollte die Unterstreichung zu wenig abstrakt und zu besonders sein, so kann das Beispiel ohne das Pseudoelement ::before vorgezeigt werden:

```
.mitUL {font-size:3em; position:relative; display:inline-block;
  animation: meineAniStart 2s;}
@keyframes meineAniStart {
  from {transform: scaleX(0);
  to   {transform: scaleX(1);} }
```

```
~~~~~
<h1 class="mitUL">Unsere Abteilung als Profit-Center</h1>
```

Mit der CSS Eigenschaft `transform` lassen sich 2D und 3D Veränderungen bewerkstelligen. Wir werden uns die 2D Veränderungen ansehen, weil damit die meisten 3D Veränderungen auch visualisiert werden können.

## Skalieren

### CSS



**transform: scale(x,y);**

Verändert die Größe eines Elements über die x-Achse (Breite) und die y-Achse (Höhe). `scale(1,1)` ist die Originalgröße, `scale(0.5,0.5)` ist die halbe Größe und `scale(2,2)` vergrößert das Element um das Doppelte.

Die x und y Achse lässt sich auch direkt ansprechen:

**transform: scaleX(2); transform: scaleY(1.4);**



Für die `scale(x,y)` Funktion sind positive und negative Zahlen (Dezimal und Ganzzahlen) möglich. Bei `transform: scale(1,-1)` wird das Element gespiegelt.



Das Beispiel zeigt eine animierte Unterstreichung von der Mitte heraus.

```
.mitUL {position:relative; display:inline-block;}
.mitUL:before {content: ""; position: absolute; width: 100%;
border-bottom: 3px solid black; bottom: 0; left: 0;
animation: meineAniStart 2s;}
@keyframes meineAniStart {
from {transform: scaleX(0);}
to {transform: scaleX(1);} }
~~~~~
<h1 class="mitUL">Unsere Abteilung als Profit-Center</h1>
```

## Drehen

### CSS



**transform: rotate(winkel);**

Dreht ein Element im Uhrzeigersinn. Bei einem negativen Winkel gegen den Uhrzeigersinn.

Der Winkel-Wert wird als Zahl plus `deg` angegeben:

**transform: rotate(45deg);**

Andere Winkelmaße sind `grad` (Gon), `rad` (Radiant), `turn` (Vollwinkel).

Bei `deg` entspricht eine Kreisumrundung  $360^\circ$ .



Das Beispiel simuliert einen analogen Sekundenzeiger.

```
#drehe {animation: meineDrehung infinite 60s linear ;
display: inline-block; margin: 20%;}
@keyframes meineDrehung {
from {transform: rotate(-90deg);}
to {transform: rotate(270deg);} }
~~~~~
<div id="drehe">Sekunden &rArr;</div>
```

# Übungsblatt 13.4 Transformieren

Referenzcode **HCU134**

Technologien HTML, CSS

Dateinamen U13\_4\_Transformieren.odt, U13\_4\_Transformieren.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... fügen drei Spielkarten in ein HTML Dokument ein.	I	II
... setzen die Pseudoklasse :target um ein aufdecken zu bewerkstelligen.	I	II
... animieren das Umdrehen einer Karte mit transform:scale.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
<b>Übung B</b>		
... suchen im Internet nach Bildern von berühmten Malern.	A	I
... erstellen aus dem Bild eine Polaroid-Optik.	I	II
... fügen diverse CSS Eigenschaften dem Bild hinzu.	I	II
... drehen das Polaroid mit transform.	I	II
... beschriften das Polaroid mit dem Namen des Künstlers.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

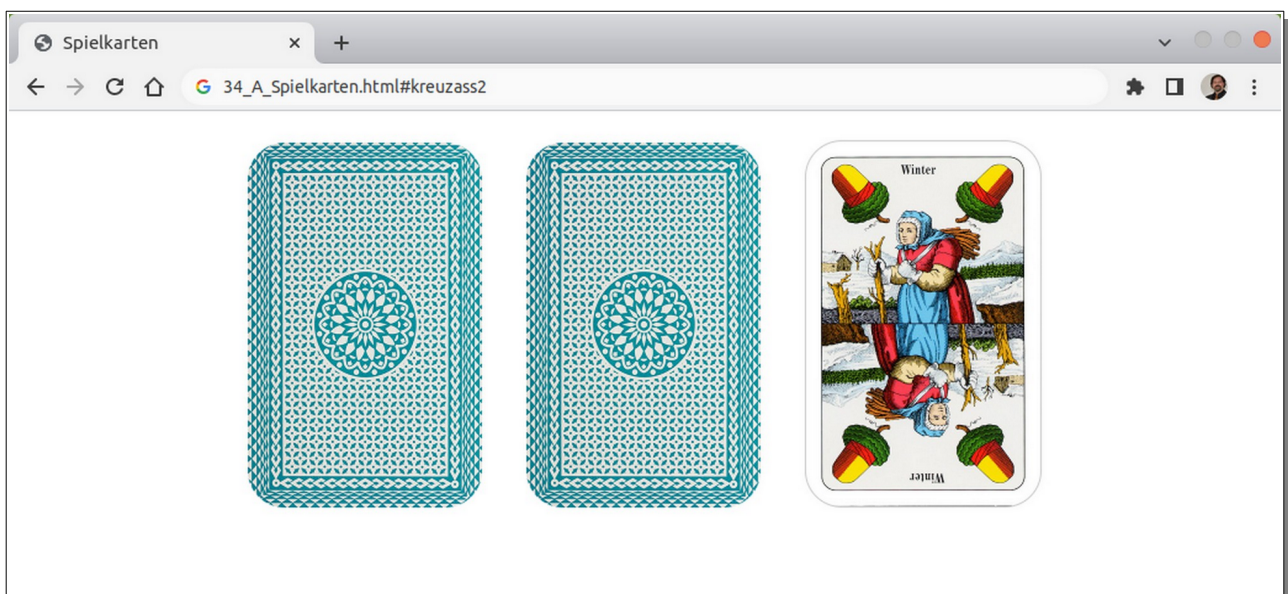
## Anmerkungen

- Der eine oder die andere Studierende mag vielleicht nicht wissen, was eine Polaroid Sofortbild-Kamera ist. Ein kleiner Ausflug in die Vergangenheit kann den Unterricht nicht schaden!
- Beim Suchen nach berühmten Selbstportraits können die Studierenden ein wenig auf die Kunstgeschichte reflektieren und Impressionen für das (Web)Design finden!



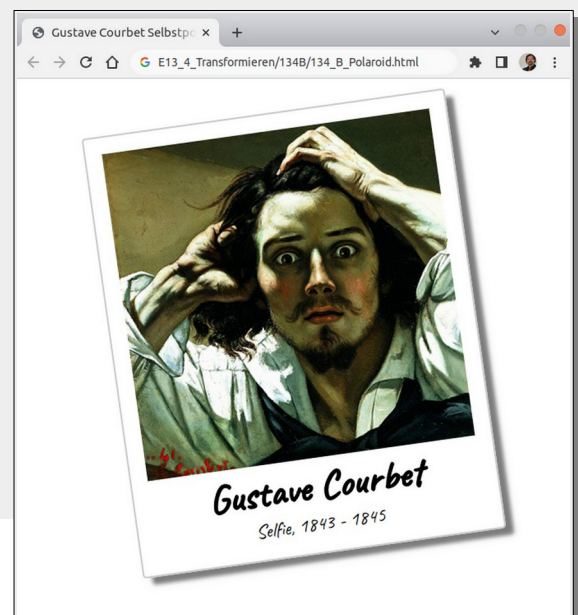
### Übung A: Spielkarten

- Erstelle eine Webseite mit drei Spielkarten. (Bilder dazu sind sicher im Internet zu finden).
- Die Spielkarten sind zum Start verdeckt aufgelegt.
- Durch einen Klick auf eine Karte wird diese umgedreht.
- Es darf immer nur eine Spielkarte aufgedeckt sein.
- Das Umdrehen soll animiert sein.
- **Freie kreative Übung.**



### Übung B: Polaroid

- Suche im Internet nach einem Selbstportrait eines berühmten Malers.
- Erstelle eine Webseite mit dem Bild in einer Polaroid Optik.
- Füge einen Schatten hinzu.
- Drehe das Polaroid mit CSS.
- **Beschrifte es mit dem Namen des Künstlers.**



# Lernhandout 13.5 Transformieren

Referenzcode **HCL135**

Technologien CSS

Dateinamen L13\_5\_Transformieren.odt, L13\_5\_Transformieren.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... verzerren ein Element über einen Winkel mit transform: skew(x,y).	I	II
... verschieben Elemente mit transform: translate(x,y).	I	II
... können einen einfachen Tab-Reiter erstellen.	I	II
... wissen, dass sich komplexere Verformungen mit transform: matrix möglich sind.	I	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Um den Tab-Reiter etwas nützlicher zu machen, kann man mit der Pseudoklasse :target und den CSS-Anweisungen display: none, bzw. display: inline-Block den Inhalt anzeigen lassen. Siehe HCL092
- Auch interessant ist das (relativ neue) <details> Element um Daten zu strukturieren. Es simuliert ein Akkordeon. Obgleich die Studierenden in der Lage sein sollten, den selben Effekt mit ihrem HTML und CSS Wissen umzusetzen, hat dieses Element doch seine Berechtigung und Ästhetik.

```
<details>
  <summary>
    Die sichtbare Überschrift
  </summary>
  <div>
    Im div steht der Content
  </div>
</details>
```

## Verzerren

## CSS



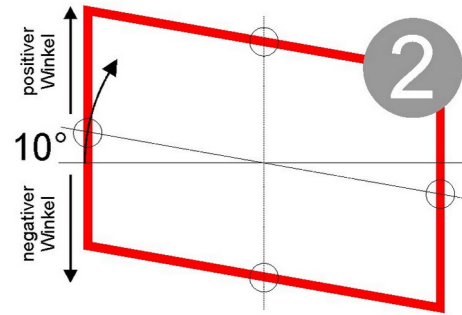
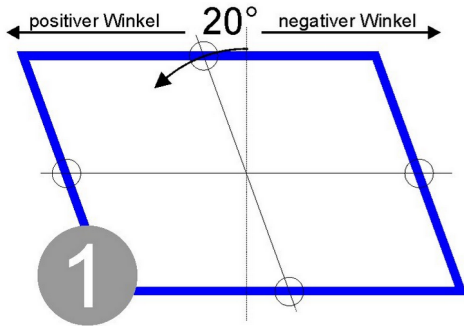
**transform: skew(x,y);**

Verzerrt ein Element über einen Winkel für die x oder y Achse.

Die x und y Achse lässt sich auch direkt ansprechen:

Form 1: **transform: skewX(20deg);**

Form 2: **transform: skewY(10deg);**



## Verschieben

## CSS



**transform: translate(x,y);**

Verschiebt ein Element über die X-Achse und Y-Achse.

Als Werte können Pixel, cm, %, em usw. gesetzt werden. Negative Werte sind ebenfalls möglich.

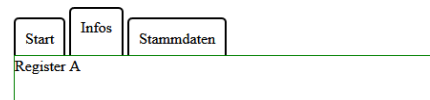
Die Eigenschaft kann auch direkt angesprochen werden:

**transform: translateX(2cm);** Zwei Zentimeter nach rechts

**transform: translateY(-35px);** 35 Pixel nach oben



Ein einfacher Tab-Reiter mit  
**transform: translate**



```
.register {height:2em; display:inline-block;
border-radius:5px 5px 0px 0px; padding:10px;
border: 2px solid black;}
.register:hover {transform: translateY(-0.7em);}
#inhalt {background-color:#FFF; border:1px solid green; height:10em;
transform: translateY(-1em);}
~~~~~
<nav><span class="register">Start</span>
<span class="register">Infos</span>
<span class="register">Stammdaten</span></nav>
<div id="inhalt">Register A</div>
```



Komplexere Verformungen von Elementen lassen sich mit der **matrix()** Funktion realisieren. z. B. **transform: matrix(1, 2, -1, 1, 80, 80);**

# Lernhandout 13.6 Transitions

Referenzcode **HCL136**

Technologien CSS

Dateinamen L13\_6\_Transitions.odt, L13\_6\_Transitions.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
... definieren Übergänge ohne Animationen oder Keyframe Regeln.	I	II
... begreifen den Unterschied zwischen Übergängen und Animationen.	I	I
... bestimmen eine besondere CSS Eigenschaft für den Übergang mit transition-property.	I	II
... definieren die Dauer des Überganges mit transitions-duration.	I	II
... ergänzen einen Übergang mit einer transition-timing-function (Geschwindigkeit des Überganges).	I	II
... verzögern den Start des Überganges (Wartezeit) mit transition-delay.	I	II
... kennen und nutzen die Shorthand-Eigenschaft transition.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine

Auch ohne Animationen oder Keyframe Regeln, lassen sich Übergänge mit CSS definieren. Die `transition` Eigenschaft kommt immer dann zu tragen, wenn sich ein Element verändert (z. B. durch `:hover`), beim Start und in Verbindung mit einer Transformation. Um einen weichen Übergang zu erzeugen, muss man den Namen der CSS Eigenschaft und die Dauer angeben. Zusätzlich kann man noch eine Wartezeit und die Geschwindigkeit angeben.

**CSS**

**transition-property: [Wert];**

Als [Wert] gibt man entweder die CSS Eigenschaft an, oder `all` um allen Eigenschaften einen weichen Übergang zu geben.

**transition-property: background-color;**

Der Übergang wird nur auf die Hintergrundfarbe des Selektors angewandt.

**transition-property: all;**

Alle Eigenschaften des Selektors bekommen einen Übergang.

**CSS**

**transition-duration: [Wert];**

Die Dauer des Überganges. Als [Wert] gibt man eine Zahl in Sekunden an.

**transition-duration: 2s;**                      2 Sekunden

**CSS**

**transition-timing-function: [Wert];**

Die Geschwindigkeit des Überganges.

Als [Wert] sind unter anderen folgende möglich:

**ease**                      langsamer Start, dann schneller und gegen Ende wieder langsam

**linear**                    Gleiche Geschwindigkeit

**ease-in**                  langsamer Start

**ease-out**                langsames Ende

Darüber hinaus gibt es noch `steps(int,start|end)`, `step-start`, `step-end`, `ease-in-out` und `cubic-bezier(n,n,n,n)`.

**CSS**

**transition-delay: [Wert];**

Verzögerung. Als [Wert] gibt man eine Zahl in Sekunden an.

**transition-delay: 3s;** 3 Sekunden Wartezeit bis zum Start.

```
.schalter {background:blue; color:white; padding:20px;
  transition-property:all; transition-duration:3s;
  transition-timing-function:ease;
  transition-delay:0.5s;}
.schalter:hover {background:yellow; color:black;}
```

**CSS**

**transition: [property] [duration] [timing-function] [delay];**

Shorthand – man kann alle vier Eigenschaften mit einer zusammenfassen.

```
.schalter {transition: all 3s ease 0.5s;}
```

# Übungsblatt 13.6 Transitions

Referenzcode **HCU136**

Technologien HTML, CSS

Dateinamen U13\_6\_Transitions.odt, U13\_6\_Transitions.pdf

<b>Feinziele</b>	<b>Zielart</b>	<b>Taxonomie</b>
Die Studierenden ...		
<b>Übung A</b>		
... erstellen Buttons mit weichem Übergang.	<b>I</b>	<b>II</b>
... erzeugen einen 3D Effekt mit einem Hover.	<b>I</b>	<b>II</b>
<b>Übung B</b>		
... erstellen ein CSS Accordion.	<b>I</b>	<b>II</b>
... beschäftigen sich mit dem Thema Drogen (Gesundheitsbildung).	<b>SA</b>	<b>III</b>
... öffnen und schließen anklickbare Elemente mit CSS.	<b>I</b>	<b>II</b>
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	<b>II</b>
<b>Übung C</b>		
... erstellen eine Webseite zum Thema "Mülltrennung".	<b>I</b>	<b>II</b>
... beschäftigen sich mit dem Thema "Mülltrennung" (Umwelterziehung).	<b>SA</b>	<b>III</b>
... gestalten die Webseite mit Tab-Reitern.	<b>I</b>	<b>II</b>
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	<b>II</b>

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Keine



### Übung A: Button mit Übergang

- Erstelle ein neues HTML Dokument.
- Gestalte zwei Buttons mit weichem Übergang.
- **Bei einem Hover soll ein 3D Effekt entstehen.**



3D Effekte lassen einfach sich durch einen veränderten Box-Shadow realisieren.

Zu unseren Angeboten

Zurück zur Website



### Übung B: Accordion

- Erstelle ein neues HTML Dokument mit einem Accordion
- Die Kategorien sind:
  - 4 Legale Drogen
  - 5 Sanfte Drogen
  - 6 Harte Drogen
- Suche im Internet nach Beispielen für die Kategorien (z. B: Harte Drogen, Heroin, Opium usw.) mit Wirkung und Gefahren.
- Durch einen Klick auf die Kategorie soll der Inhalt mit den Beispielen angezeigt werden – die anderen schließen sich.
- Das Öffnen und Schließen soll animiert sein.
- **Gestalte die Webseite anspruchsvoll mit CSS.**

Legale Drogen

Weiche Drogen

- Cannabis: wirkt psychoaktiv.
- LSD: kompletter Realitätsverlust.
- Exstasy: kann Erektions- und Orgasmusstörungen auslösen.

Harte Drogen



### Übung C: Mülltrennung

- Erstelle eine Webseite zum Thema "Mülltrennung"
- Gestalte es mit einem Tab-Reiter.
- **Freie kreative Übung.**

# Lernhandout 14.1 CSS Variablen

Referenzcode **HCL141**

Technologien CSS

Dateinamen L14\_1\_Variablen.odt, L14\_1\_Variablen.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... erkennen und verstehen den Sinn von benutzerdefinierten Eigenschaften (custom properties) bzw. CSS Variablen.	I	I
... können Variablen festlegen.	I	II
... rufen Variablen innerhalb des CSS Bereichs ab.	I	II
... definieren ein Fallback für CSS Variablen, falls kein Browsersupport besteht.	I	II






**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Im Beispiel muss man nur die Variablenwerte ändern und man sieht die Auswirkungen dann auf die Darstellung. Bei extrem vielen CSS Anweisungen kann ein cleveres Benutzen von Variablen eine Arbeitszeitreduktion nach sich ziehen. Man kann die Vorteile von Variablen abwägen mit dem Benutzen von Suchen-und-Ersetzen!

Mit benutzerdefinierten Eigenschaften (custom properties) oder CSS Variablen kann man zu Beginn Werte festlegen, die später bei einem Redesign nur einmal geändert werden müssen.

Browsersupport					
<code>var()</code>	49.0	15.0	31.0	9.1	36.0

### Variable festlegen

#### CSS



```
:root {--variablenname: WERT;}
```

Die Variablen sollten zu Beginn der CSS Anweisungen definiert werden. Der Variablenname ist frei wählbar, man muss aber Groß- und Kleinschreibung beachten, weil diese Case-Sensitive sind.

```
:root {--akzent: #009;}
```

### Variable abrufen

#### CSS



```
css-selector {css-eigenschaft: var(--variablenname);}
```

Die festgelegte Variable kann nun im gesamten CSS Bereich mit `var()` abgerufen werden.

```
h1 {color: var(--akzent);}
```



Um sicher zu gehen, kann man einen Fallback definieren der dann angewandt wird, wenn der Browser die Variablenfunktion nicht versteht. Dafür wird ein Beistrich gesetzt und der Fallback-Wert hinzugefügt.

```
h1 {color: var(--akzent, blue);}
```



#### Beispiel zur Nutzung von unterschiedlichen Variablentypen

```
:root {--schriftfarbe: #333;
  --schriftart: Arial, Times;
  --schriftgr: 2em; }

h1 {color:var(--schriftfarbe, blue);
  font-family:var(--schriftart);
  font-size:var(--schriftgr, 3em);}

p {color:var(--schriftfarbe, black);
  font-family:var(--schriftart);}
```

---

```
<h1>Ökologisch Einkaufen!</h1>
<p>Gut und günstig - die Umwelt freuts!</p>
```

# Lernhandout 14.2 calc() Funktion

Referenzcode: **HCL142**

Technologien: CSS

Dateinamen: L14\_2\_calc.odt, L14\_2\_calc.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... nutzen die CSS calc() Funktion um Berechnungen durchzuführen.	I	II
... kennen und nutzen die Operationsmöglichkeiten mit calc().	I	II
... achten dabei auf die korrekte Syntax.	IK	II
... kombinieren die calc() Funktion mit CSS Variablen var().	I	II
... definieren ein Fallback für Berechnungen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- In CSS gilt: "Wer zuletzt kommt, malt auch!". Somit kann immer ein Fallback durch eine Definition mit niedrig bekannten Eigenschaften gesetzt werden. Kennt der Browser eine bestimmte Spezifikation nicht, dann ignoriert er es (in der Regel) und übernimmt die zuletzt gültige!

z. B.

```
height: 4em;
```

```
height: 4vh;
```






Um einer CSS Anweisung mehr Bedeutung zu verleihen, gibt es die !important Regel. Diese überbietet (overruled) gleiche CSS-Anweisungen mit unterschiedlichen Werten!

z. B.

```
p {font-size: 1em!important;}
```

```
p {font-size: 6em;}
```

Mit der `calc()` Funktion können einfache Berechnungen in die CSS Anweisungen eingebunden werden. Diese sind jedoch nur bei Eigenschaften möglich, die ein numerisches Maß erwarten (z. B: `width`, `height`, `font-size`, `margin`, `padding` usw.). Dabei können Berechnungen zwischen unterschiedlichen Maßen angestellt werden z. B. `width: calc(10em - 30px);`

Browsersupport					
<code>calc()</code>	26.0	9.0	16.0	7.0	15.0

## CSS



CSS-Eigenschaft: `calc()`;

Mögliche Operationen sind

- Addition (+)
- Subtraktion (-)
- Multiplikation (\*)
- Division (/)

sowie Klammern. Vor und nach dem Rechenzeichen muss ein Leerzeichen gesetzt werden (zumindest bei Plus und Minus).

Unmögliche Rechenergebnisse (z. B: Division durch Null, oder eine negative Zahl für `width`) werden ignoriert.

```
font-size: calc(12pt + 1em);
width:      calc(80% - 2 * 3px);
min-width: calc((60% + 8px) / 2 );
```



Natürlich kann man die `calc()` Funktion mit CSS Variablen `var()` kombinieren.

```
:root {--breite:300px;
        --schrift:12pt;}
#eineBox {height: 200px; border: 3px solid red;
           width:   calc(var(--breite) + 100px);
           font-size: calc(var(--schrift) * 2);}
~~~~~
<div id="eineBox">Drinnen ist es dunkel</div>
```



Ein Fallback setzt man über die CSS Reihenfolge!

```
#eineBox {height: 200px; border: 3px solid red;
           width:   400px;
           width:   calc(var(--breite) + 100px);
           font-size: calc(var(--schrift) * 2);}
```

# Lernhandout 14.3 counter() Funktion

Referenzcode **HCL143**

Technologien CSS

Dateinamen L14\_3\_counter.odt, L14\_3\_counter.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... nutzen die counter() Funktion um Elemente zu nummerieren.	I	II
... setzen den Zähler mit counter-reset auf Null.	I	II
... kombinieren die counter-increment Eigenschaft mit dem Pseudoelement ::before um eine Durchnummerierung vor einem Element zu erzeugen.	IK	II
... definieren das Aussehen und die Erscheinung einer Nummerierung im Pseudoelement ::before über die content-Eigenschaft und der counter() Funktion.	IK	II
... verschachteln Zähler über mehrere Ebenen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Im Beispiel "Zähler verschachteln" wird der zweite counter-reset im h1 Selektor definiert. Ebenso gut könnte man auch ein div oder einen Universalselektor für die zweite Ebene heranziehen. Leider funktioniert es (zur Zeit noch) nicht mehrere counter-resets im body festzulegen.

Neben dem `<ol>` Element, kann man auch mit CSS die Anzahl der Verwendungen durchnummerieren lassen. z. B. jede Überschrift `<h1>` durchnummerieren.

Dafür muss man zuerst den Zähler mit der CSS Eigenschaft `counter-reset` auf Null setzen und dem Zähler einen Namen geben. Vorzugsweise im Selektor für den `<body>`, damit der Zähler für das gesamte Dokument gilt.

## CSS

```
counter-reset: NAME;
```

```
body {counter-reset: abschnitt;}
```

Im Anschluss wird im Selektor für das gewünschte Element der Zähler aktiviert. Als Selektor verwenden wir ein `h2` Element mit einem `::before` Pseudoelement. Damit wird bei jeder Überschrift der zweiten Ebene vor dem Text der Zähler hinzugefügt.

## CSS

```
counter-increment: NAME;
```

```
h2::before {counter-increment: abschnitt;}
```

Mit der `content` Eigenschaft und der `counter()` Funktion bestimmen wir das Aussehen.

## CSS

```
content: counter(NAME);
```

```
body {counter-reset: abschnitt;}
h2::before {counter-increment: abschnitt;
             content: "Kapitel " counter(abschnitt) ": " ;}
```



## Zähler verschachteln (mehrere Ebenen durchnummerieren)

```
body {counter-reset: ersteEbene;}
h1 {counter-reset: zweiteEbene;}
h1::before {counter-increment: ersteEbene;
            content: "Kapitel " counter(ersteEbene) ": " ;}
h2::before {
  counter-increment: zweiteEbene;
  content: counter(ersteEbene) "." counter(zweiteEbene) " ";}
~::~:
<h1>Unternehmensführung</h1>
  <h2>Die Planung</h2>
  <h2>Die Kontrolle</h2>
<h1>Personalmanagement</h1>
  <h2>Arbeitnehmer haben Rechte</h2>
  <h2>Motivierte Mitarbeiter</h2>
<h1>Ökomanagement</h1>
  <h2>Grundlagen</h2>
  <h2>Instrumente</h2>
```

**Kapitel 1: Unternehmensführung**

1.1 Die Planung

1.2 Die Kontrolle

**Kapitel 2: Personalmanagement**

2.1 Arbeitnehmer haben Rechte

2.2 Motivierte Mitarbeiter

**Kapitel 3: Ökomanagement**

3.1 Grundlagen

3.2 Instrumente

# Übungsblatt 14.3 CSS Funktionen

Referenzcode: **HCU143**

Technologien: HTML, CSS

Dateinamen: U14\_3\_CSSFunktionen.odt, U14\_3\_CSSFunktionen.pdf, impressum.html

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... erstellen eine CSS Farbpalette mit Variablen.	I	II
... definieren die Farbvariablen in einem externen Stylesheet.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... binden das externe CSS Stylesheet in das Dokument ein und testen es mit beliebigen HTML Elementen.	I	IV
<b>Übung B</b>		
... definieren CSS Variablen für Schriftfamilien.	I	II
... definieren CSS Variablen für Schriftgrößen.	I	II
... erstellen Berechnungen mit calc() und den definierten Variablen nach schriftlicher Vorgabe.	I	III
... erstellen einen verschachtelten counter().	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- **Zu Übung B:** Es ist gesetzlich festgelegt, dass jede Webseite ein Impressum bzw. reale Kontaktdaten aufweisen soll. Das abrufbare Impressum (impressum.html) kann als Vorlage für andere online Webprojekte verwendet werden.



### Übung A: Farbpalette

- Erstelle ein neues CSS Stylesheet
- Definiere mit CSS Variablen eine eigene Farbpalette im externen Stylesheet.  
--meinBlau, --meinRot, --Akzent1 usw.
- Binde das CSS Stylesheet in ein beliebiges HTML Dokument ein.  
TIPP: `<link rel="stylesheet" ...`
- Teste deine Farbpalette.
- **Freie kreative Übung.**



### Übung B: Impressum

- Öffne das HTML Dokument `impressum.html`
- Definiere die Schriftarten mit CSS Variablen.
  - Schriftfamilie für Überschriften
  - Schriftfamilie für Lauftext
- Definiere eine CSS Variable für die Schriftgröße mit 12pt.
- Die Schriftgrößen sollen mit `calc()` und der CSS Variable für die Schriftgröße berechnet werden.
  - h1 = doppelt so groß wie die CSS Variable
  - h2 = plus 0.5em zur CSS Variable
  - h3 = plus 2pt zur CSS Variable
  - p = die CSS Variable
- **Erstelle einen verschachtelten `counter()` für `<h2>` und `<h3>` (Nummerierung für die Überschriften)**

#### 1. Disclaimer – rechtliche Hinweise

##### 1.1 Haftungsbeschränkung

Die Inhalte dieser Website werden mit größtmöglicher Sorgfalt erstellt. Der Anbieter übernimmt jedoch keine Gewähr für die Richtigkeit, Vollständigkeit und Aktualität der bereitgestellten Inhalte. Die Nutzung der Inhalte der Website erfolgt auf eigene Gefahr des Nutzers. Namentlich gekennzeichnete Beiträge geben die Meinung des jeweiligen Autors und nicht immer die Meinung des Anbieters wieder. Mit der reinen Nutzung der Website des Anbieters kommt keinerlei Vertragsverhältnis zwischen dem Nutzer und dem Anbieter zustande.

##### 1.2 Externe Links

Diese Website enthält Verknüpfungen zu Websites Dritter ("externe Links"). Diese Websites unterliegen der Haftung der jeweiligen Betreiber. Der Anbieter hat bei der erstmaligen Verknüpfung der externen Links die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu dem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Der Anbieter hat keinerlei Einfluss auf die aktuelle und zukünftige Gestaltung und auf die Inhalte der verknüpften Seiten. Das Setzen von externen Links bedeutet nicht, dass sich der Anbieter die hinter dem Verweis oder Link liegenden Inhalte zu Eigen macht. Eine ständige Kontrolle der externen Links ist für den Anbieter ohne konkrete Hinweise auf Rechtsverstöße nicht zumutbar. Bei Kenntnis von Rechtsverstößen werden jedoch derartige externe Links unverzüglich gelöscht.

##### 1.3 Urheber- und Leistungsschutzrechte

Die auf dieser Website veröffentlichten Inhalte unterliegen dem Urheber- und Leistungsschutzrecht. Jede vom Urheber- und Leistungsschutzrecht nicht zugelassene Verwertung bedarf der vorherigen schriftlichen Zustimmung des Anbieters oder jeweiligen Rechteinhabers. Dies gilt insbesondere für Vervielfältigung, Bearbeitung, Übersetzung, Einspeicherung, Verarbeitung bzw. Wiedergabe von Inhalten in Datenbanken oder anderen elektronischen Medien und Systemen. Inhalte und Rechte Dritter sind dabei als solche gekennzeichnet. Die unerlaubte Vervielfältigung oder Weitergabe einzelner Inhalte oder kompletter Seiten ist nicht gestattet und strafbar. Lediglich die Herstellung von Kopien und Downloads für den persönlichen, privaten und nicht kommerziellen Gebrauch ist erlaubt. Die Darstellung dieser Website in fremden Frames ist nur mit schriftlicher Erlaubnis zulässig.

#### 2. Datenschutzerklärung

##### 2.1 Datenschutz

Nachfolgend möchten wir Sie über unsere Datenschutzerklärung informieren. Sie finden hier Informationen über die Erhebung und Verwendung persönlicher Daten bei der Nutzung unserer Webseite. Ich weise ausdrücklich darauf hin, dass die Datenübertragung im Internet (z.B. bei der Kommunikation per E-Mail) Sicherheitslücken aufweisen und nicht lückenlos vor dem Zugriff durch Dritte geschützt werden kann. Die Verwendung der Kontaktdaten meines Impressums zur gewerblichen Werbung ist ausdrücklich nicht erwünscht, es sei denn ich hätte zuvor eine schriftliche Einwilligung erteilt oder es besteht bereits eine Geschäftsbeziehung.

##### 2.2 Personenbezogene Daten

Sie können unsere Webseite ohne Angabe personenbezogener Daten besuchen. Soweit auf meinen Webseiten personenbezogene Daten (Namen, Adressen, E-Mail-Adressen) erhoben werden, erfolgt dies, soweit möglich, auf freiwilliger Basis. Diese Daten werden ohne Ihre Einwilligung nicht an Dritte weitergegeben und werden ausschließlich zu den Ihnen angegebenen Zwecken verarbeitet. Die Erhebung, Speicherung, Verwendung und Weitergabe dieser Daten erfolgt auf Grundlage Ihrer Einwilligung. Sie können diese Einwilligung jederzeit widerrufen, indem Sie die entsprechenden Links auf dieser Seite nutzen. Die Daten werden inhaltlich ausgestaltet oder geändert werden.

# Lernhandout 14.4 Attributselektoren

Referenzcode **HCL144**

Technologien CSS

Dateinamen L14\_4\_Attributselektor.odt, L14\_4\_Attributselektor.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... selektieren einzelne Elemente über ein bestimmtes Attribut.	I	II
... kennen die Syntax für Attributselektoren.	I	I
... erweitern einen Attributselektor mit einem zusätzlichen Wert.	I	II
... bestimmen Teilübereinstimmungen für Attributwerte (zu Beginn, am Ende und innerhalb).	IK	II
... kombinieren Attributselektoren mit anderen Selektoren.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- Schön anzusehen ist, dass die Sonderzeichen der Teilübereinstimmungen für Attributwerte sich an RegEx orientieren! Die Grenzbezüge in RegEx kennen ^ für den Start und \$ für das Ende einer Zeile bzw. eines Textes.

Mit dem einfachen Attributselektor können Elemente angesprochen, die ein bestimmtes Attribut besitzen. Zur Erinnerung: in `<p title="Absatz1">` ist `title` das Attribut. Angesprochen wird das ganze Element indem der Attributselektor mit eckigen Klammern im CSS Bereich angegeben wird. z. B: `[title] {color:red;}`

## CSS



`[Attribut]: {CSS Eigenschaft;}`

Im Beispiel werden alle Pflichtfelder blau eingefärbt und alle Beschriftungen die ein Title Attribut haben eine rote Schriftfarbe zugewiesen.

```
[required] {background-color:blue;}
[title] {color:red;}
```

```
<label for="vorname" title="Vorname">Vorname: </label>
<input name="vorname" id="vorname" type="text" required ><br>
<label for="nachname" title="Nachname">Nachname: </label>
<input name="nachname" id="nachname" type="text" required >
```

## CSS



`[Attribut=Attributwert]: {CSS Eigenschaft;}`

Gibt man dem Attribut zusätzlich (durch ein = Zeichen) einen Wert mit, so werden nur jene Elemente angesprochen, die sowohl Attribut als auch den passenden Wert haben. Im Beispiel wird nur der Reset-Button durchgestrichen.

```
[type=reset] {text-decoration: line-through;}
```

```
<button type="submit">Absenden</button>
<button type="reset">Zurücksetzen</button>
```



Es gibt drei Teilübereinstimmungen für die Attributwerte.

`[Attribut^="abc"]` ← Zeichenkette zu Beginn des Attributwertes  
`[Attribut$="xyz"]` ← Zeichenkette am Schluss des Attributwertes  
`[Attribut*="klm"]` ← Zeichenkette innerhalb des Attributwertes.

```
[href^="http://"] {font-weight: bold }
[href$=".com"] {color: green }
[title*="Graz"] {font-style:italic;}
```

```
<a href="http://www.css4.at">CSS4 Lernplattform</a>
<a href="http://www.wordpress.com">Blog Anbieter</a>
<h1 title="Für Graz und Graz-Umgebung">In ihrer Region</h1>
```



Attributselektoren können mit anderen Selektoren verbunden werden:

Universalselektor: `*[ Attribut]` → `*[title="Wichtig"] {...;}`  
 Typselektor: `element[Attribut]` → `a[href$=".net"] {...;}`  
 Klassenselektor: `.klasse[Attribut]` → `.bild[src*="png"] {...;}`

# Lernhandout 15.1 Image Map

Referenzcode **HCL151**

Technologien HTML

Dateinamen L15\_1\_ImageMap.odt, L15\_1\_ImageMap.pdf, Austria.png

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
... begreifen die Möglichkeiten von Image Maps.	<b>K</b>	III
... definieren eine Image Map mit dem Attribut usemap im <img> Element.	<b>II</b>	II
... verbinden Bild mit Image Map über den <map> Tag mit dem Attribut name.	<b>I</b>	II
... definieren den <area> Bereich und weisen die Attribute href (Hyperlink, Verweis) und target hinzu.	<b>I</b>	II
... bestimmen die Form des anklickbaren Bereichs mit dem shape Attribut.	<b>I</b>	II
... bestimmen die Koordinaten für den anklickbaren Bereich.	<b>I</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- Image-Map Generator, <https://www.image-map.net/>

## Anmerkungen

- Mit einem Image-Map Generator aus dem Internet kann man bequem die Koordinaten für ein bestimmtes Bild erheben.
- x und y ist gleich wie auf einem Koordinatensystem, nur mit dem Unterschied, dass der Punkt P(0,0) links oben beginnt.
- Im Beispiel wird ein Rechteck über Oberösterreich gezogen. Dabei ist ein beachtlicher Teil von Salzburg auch dabei. Mit einem Polygon kann man es schon präziser gestalten.

Mit einer Image Map kann der Anwender per Maus auf ein Detail in einem Bild klicken um damit einen Verweis (Link) auszuführen. z. B. Es wird eine Österreichkarte angezeigt und er User kann ein Bundesland auswählen. Zuerst wird ein beliebiges Bild mit dem Attribut `usemap` eingefügt (mit #). Dann ein `<map>` Element mit dem Attribut `name`, welches gleich sein muss (ohne #) wie das `usemap` Attribut im Bild. Innerhalb des `<map>` Tags wird über das `<area>` Element der anklickbare Bereich festgelegt.

## HTML

```
<img src="" ... usemap="#mapname">
<map name="#mapname">
  <area shape="" coords="" href="" target="">
</map>
```



Es können beliebig viele `<area>` Tags innerhalb des `<map>` Elements gesetzt werden. Folgende Attribute sind möglich:

**shape** bestimmt die Form

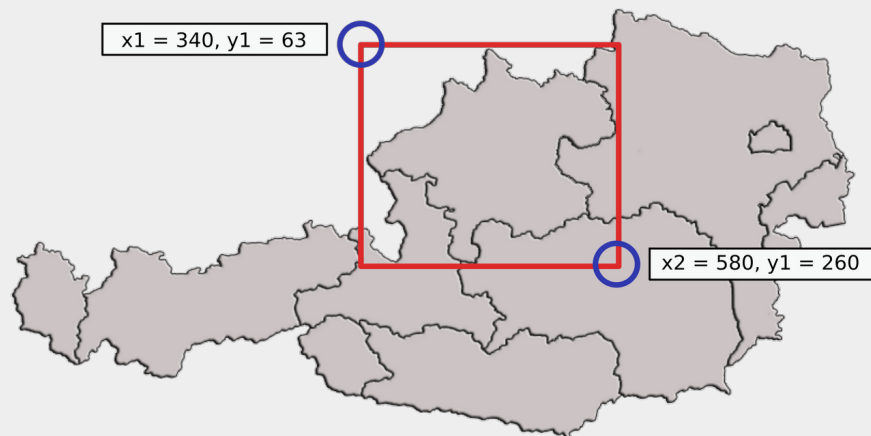
**shape="rect"** ← eine rechteckige Fläche

**shape="circle"** ← ein Kreis

**shape="poly"** ← ein Polygon

**coords** bestimmt die Koordinaten der Form

Für **shape="rect"** gilt **coords="x1, y1, x2, y2"**



Für **shape="circle"** gilt **coords="x, y, r"**

**x** = Mittelpunkt, Anzahl der Pixel von links

**y** = Mittelpunkt, Anzahl der Pixel von oben

**r** = Radius in Pixel

Für **shape="poly"** gilt **coords="x1, y1, x2, y2 ... xn, yn"**

*href* und *target* benötigt man für den Verweis.

```

<map name="#AustriaKarte">
  <area shape="rect" coords="340, 63, 580, 260"
    href="http://www.ooe.gv.at"
    alt="Oberoesterreich" target="_blank" >
</map>
```

# Übungsblatt 15.1 Image Map

Referenzcode **HCU151**

Technologien HTML, CSS

Dateinamen U15\_1\_ImageMap.odt, U15\_1\_ImageMap.pdf

Feinziele	Zielart	Taxonomie
Die Studierenden ...		
<b>Übung A</b>		
... nutzen ein Bildbearbeitungsprogramm (wie z. B. Photoshop oder Gimp) um in ein Bild fünf Fehler hinein zu retuschieren.	<b>P</b>	II
... binden beide Bilder (Original und Fehlerhaftes Bild) in ein HTML Dokument ein und erstellen eine Image Map über die eines der zwei Bilder.	<b>I</b>	II
... definieren den Area Bereich.	<b>I</b>	II
... kreieren ein Feedback (Rückmeldung) Modell.	<b>IK</b>	IV
... kombinieren die href Attribute mit einem Feedback für den User.	<b>IK</b>	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	<b>A</b>	II
... nutzen auch Image-Map Generatoren für die <map> und <area> Elemente.	<b>I</b>	II
... verändern mit CSS das Erscheinungsbild des Mauszeigers.	<b>I</b>	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Fremdquellen

- Image-Map Generator, <https://www.image-map.net/>

## Anmerkungen

- M. C. Eschers "Belvedere" aus dem Jahr 1958 ist wohl m. E. eines der schönsten Beispiele für optische Täuschung die im Surrealismus mündet.



### Übung A: Fehlersuchbild

- Öffne mit einem Bildbearbeitungsprogramm ein beliebiges Bild
- Retuschiere in dem Bild fünf Fehler
- Erstelle ein neues HTML Dokument.
- Die fünf Fehler sollen über eine Image Map anklickbar sein.
- Baue ein Feedback ein  
(wenn man auf einen Fehler klickt, soll eine Rückmeldung kommen!)
- **Gestalte die Seite mit CSS anspruchsvoll. Freie kreative Übung.**



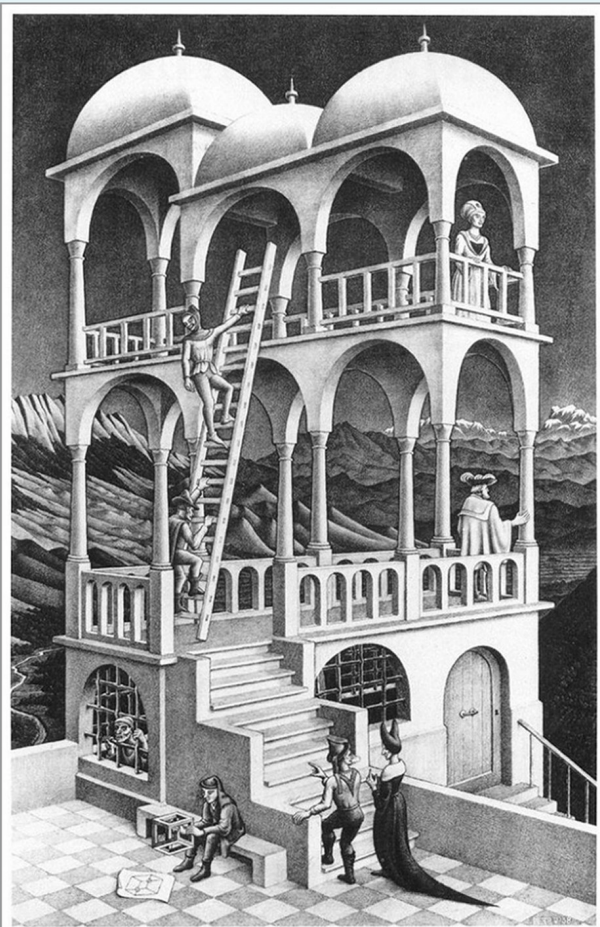
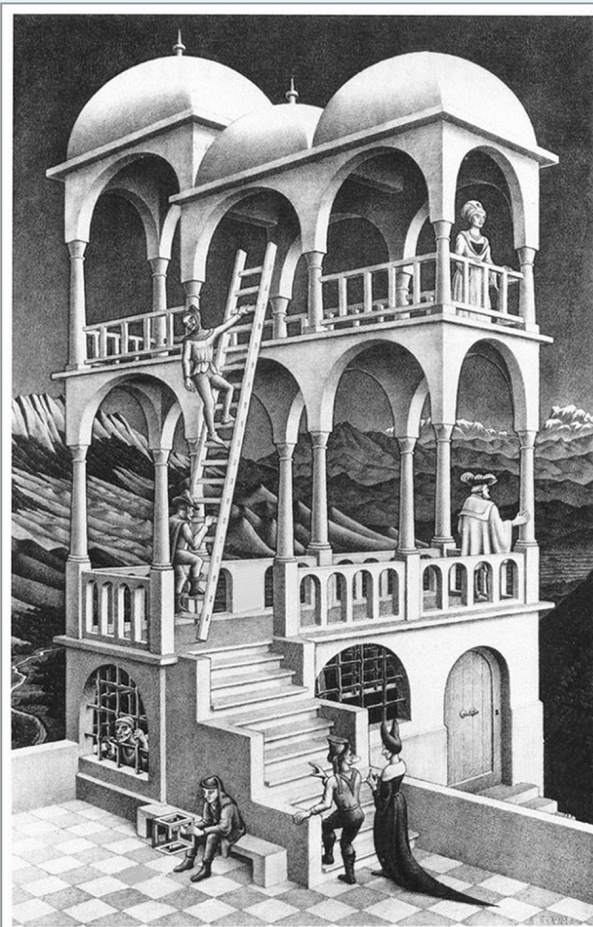
Im Internet findest du zahlreiche Image-Map Generatoren  
z. B. auf <https://www.image-map.net>



Mit der CSS Eigenschaft `cursor` kann man den Mauszeiger verändern.  
z. B. `#meinbild:hover {cursor:pointer;}`

## Fehlersuchbild

Finde die 5 Fehler im linken Bild



# Pädagogischer Beipackzettel

Die Pädagogik ist wahrscheinlich die kontroversiellste Wissenschaft unter den Geisteswissenschaften. Drei Lehrer\_innen habe meist auch drei Didaktiken. Bestandteile von unterschiedlichen Reformpädagogen werden eklektisch zu einem individuellen Gesamtkonzept zusammengeführt – man nimmt sich was man braucht und nutzt was Sinn ergibt. Konsens wird meist von oben befohlen – der Kodex der Schulgesetze 'overruled' die eigene Fachdidaktik. Das Motto soll lauten: "Lasst' uns unsere pädagogische Freiheit bewahren indem wir unser Hirn benutzen und den kritischen Blick auf die vergangene, gegenwärtige und zukünftige Gesellschaft bewahren". Dabei ist die Pädagogik (so denn sie als Erziehungswissenschaft verstanden werden will) eine der ältesten Wissenschaften der Menschheit – neben der Suche nach Gott und Nahrung ist sie jener Beitrag zum Menschwerden der erst unsere Kultur geschaffen hat.

Als Sokrates im antiken Griechenland zum Tode für "die Verführung der Jugend" gerichtet wurde, ging er nicht nur als Philosoph in die Geschichte ein, sondern vielmehr als der erste Pädagoge der für seine Auffassung starb. Dabei muss aber Pädagogik nicht dramatisch, schrill und laut sein. Das pädagogische Genie kann ebenso introvertiert, wortkarg, zurückhaltend, verschlossen und zugeknöpft sein. Der Genius muss sich nicht immer durch Marktschreierei hervortun – er oder sie kann auch das Höchste in einem Spannungsfeld zwischen Stille und Getöse schaffen und wird vielleicht Zeit seines\_ihres Lebens niemals die Würdigung erfahren, die ihm oder ihr zusteht. Die Frage "Was ist ein guter Lehrer?" füllt genauso viele Bücher wie es die Frage "Was ist Philosophie?" tut.

Dieser pädagogische Beipackzettel war ursprünglich als Textbeigabe zu den Risiken und Nebenwirkungen von Methoden gedacht. Ich will ihn aber dazu nutzen, um meine pädagogischen Meinungen der Unterrichtskonzeption beizulegen. Es sind Meinungen die nicht empirisch abgesichert sind, deshalb spannt sich über alle folgenden Worte ein "meines Erachtens" ...

## Warum überhaupt noch HTML?

Eine Frau Dr. Schubert und ihr Kollege Dr. Schwill haben eine äußerst erklärende Schrift mit dem Titel: "Didaktik der Informatik" herausgegeben. ISBN: 978-3-8274-2658-9

## Gedanken zur Leistungsbewertung

Die Bewertung des\_der Schüler\_in ist keine interpersonelle sondern soll sich, getreu einem kognitivistischen Gesamtkonzept an der Prozessleistung des Lernerfolgs orientieren. Da die Leistungsbewertung im Vergleich zur Beurteilung immanent und ständig erfolgt, ist dem Lernprodukt nicht der gleiche Stellenwert beizumessen, wie es das punktuelle Lernprodukt einer Prüfungssituation nach sich zieht. So gesehen ist es klar, dass die Portfolio-Methode schlussendlich zur Beurteilung herangezogen werden kann – und m. E. auch in der Praxis passieren sollte. Die punktuelle Prüfung des Gelernten mag vielleicht einen Anreiz zur zweiten (bzw. mehrmaligen) Beschäftigung mit

der Materie nach sich ziehen – ist aber m. E. nur sinnvoll in einem mündlichen Prüfungsgespräch, wo eigentlich nie die tatsächliche Festigung der Materie als eher die Fähigkeit Antworten auf Fragen zu geben geprüft wird. Ich empfehle, Kommentare im Quellcode des HTML und CSS Dokuments ebenfalls als prozessorientierte Leistung zu werten, wie das logisch-zielorientierte Denken – welches sich in der Klarheit eines EDV-Codes (in welcher Form auch immer) darstellt. Ein bedeutender prozessorientierter Akt im Webdesign liegt definitiv in der Formulierung von visuellen Vorgaben als auditiven Text (Code) und umgekehrt – die Antizipation des Codes zu seiner visuellen Darstellung. Man merkt, dass hier also linke und rechte Gehirnhälften gleichermaßen gefordert sind und sich wechselseitig bedingen.

## Warum Blendet Learning?

---

Im Informatikunterricht und seinen verwandtschaftlich nahen Teilgebieten – ist der Begriff es blendet Learning wohl eher als ein leeres zu betrachten. Es führt zu einer paradoxen Unterrichtsgestaltung, sich selbst durch die Methode darzustellen – um so im Metabereich der Unterrichtspraxis auf das konzeptionell Höhere zurückzugreifen. Wie soll nun also das Begehren nach digitalen Kompetenzen und der Nutzung selbiger Bezugssysteme erfolgen? Die Loslösung erscheint für mich als das adäquate Mittel um das im System innewohnende durch seine Vielseitigkeit als Prinzip aufzuführen. Die Substanz wird zum erkennenden Moment und entzieht sich einem geblendeten Lernen durch die Vielschichtigkeit ihrer eigenen Möglichkeiten. Weil nun das Rudimentäre zum Absoluten gesetzt wird, kann eine Begleiterscheinung eines Lernens nur durch die methodische Unterschiedlichkeit erklärt werden. Ich will wieder exemplarisch werden: Das Erlernen des Webdesigns kann sowohl über die vergleichbar simple Plattform [www.w3schools.com](http://www.w3schools.com) angeeignet werden – als auch über die als höher einzustufende Plattform [selhtml.org](http://selhtml.org) bzw. den offiziellen Browser-Spezifikationsseiten. Die große (ultimative) Aufgabe dieser didaktischen Konzeption liegt im Mittelweg zwischen dem primitiv Einfachen und dem komplexen Vollständigen. Es liegt im Endeffekt an der Motivation des\_der Schüler\_in.

## Sozialformen in der IKT

---

Aus pädagogischer Sicht bietet das weiterführende Tutorial keinerlei Anreize zu einem konsequenten Wechsel der Sozialformen. Da nun, wie es in den meisten IT dominierten Fächern der Fall ist, das Lernen von selbigen Inhalten die sich selbst bedingen, ein Lernen im isolierten Zustand – ein Lernen im reinen Ich nach-sich-zieht – kann kein großer Wechsel prognostiziert werden. In allen Fällen, selbst wenn von kollaborativen Lernen gesprochen wird, ist reine Bildschirmarbeit weiterhin eine Isolation vom Wir-Gefühl aber: a) hochgradig potent um kognitive Ziele umzusetzen und b) eine Bereicherung des Ich-Gefühls solange sie als Selbstkompetenz verstanden wird und nicht einem kompetitiv-pädagogischen Szenario ausgesetzt wird. Man möchte meinen, es wird das Ich-Genie im Zusammenhang seines eigenem Denken konstruiert, dass sowohl dem Seienden seine Erwidern Tribut leistet wie es im gemeinschaftlichen der kollektiven Erkenntnis jenes Moment liefert, dass noch fehlt. Im besten Fall könnte man ein Szenario der Bildschirmknappheit schaffen – man könnte den\_die Schüler\_in auffordern zu zweit an einer Konsole zu arbeiten – es ist aber nur eine Augenauswischerei. Das Genie bleibt auch in diesem Bereich losgelöst von der Menge und erfährt damit kein besonders

pädagogisches Moment – HTML und CSS ist nun mal nicht die Lösung aller Probleme des pädagogischen Alltags.

## Der konsequente Lehrplanbezug

---

Die genauere Darstellung eines Lehrplanbezuges ist im gegenwärtigen Zustand eher komplex. Natürlich würde ich eine Lehrstoffverteilung auf ein oder das nächste Semester hoch graduieren. Es entzieht sich jeglicher Basis – ein vorurteilendes Argument abzuliefern – weil nun der Lehrplanbezug in seinem Wesen mannigfaltig ist – und somit weitschweifig. Es gibt keine fundamentale Erkenntnis, die den Bezug in seiner endogenen Phase erkennt und so sein Wesen jeglicher esoterischen Basis entzieht. Ich habe schon einige gedankliche Versuchungen angestellt – um genau das nicht Obskure in den Denkprozess als eine Elimination näher einfließen zu lassen. Die ganze Geschichte entbindet sich jeglicher Darstellungs-Objektivierung umso dem Gesamten seine Weisungskraft näher zu legen – wo wir uns fragen, warum auch nicht?

## Die kritische Sozialformierung

---

Der soziale Zusammenhalt im Schulischen ist definitiv nicht nur ein Fremdgesteuerter. Im Rahmen der Bildschirmarbeit, obgleich es in der allgemeinen Informatik oder deduktiv auf ein sich der informatisch gegenstandsbezogenen Unterrichtsdimension wie z. B. der Mathematik mit Hilfe von DO Kopfrechnen oder GeoGebra zur Darstellung erkenntnisrelevanter mathematischen und damit in Folge kognitiv/logischen Gedanken nach sich zieht. Tatsache ist, dass jegliche Bildschirmarbeit immer eine Arbeit auf isolierter Basis ist. Auch die 'geblendete' Vorstellung einer IT-Tätigkeit über kollaborativen oder sozialen Netzwerken ist definitiv nicht (und ich wollte schon bei Gott) ein Ersatz zur Erlangung von kommunikativen oder sozialen Kompetenzen. Die Übung des logischen Denkens – der kommunikativen Formen und seiner Aktion-Reaktion, mag vielleicht simuliert sein und dem\_der Lernenden ein Moment des Selbstbewusstseins verleihen, das sich nicht durch die konkrete als mehr durch die indirekte Konsequenz, abbildet und einstellt. Natürlich lassen sich so in der Sozialform der Einzelarbeit, kognitive Fehler in Form eines Syntax Errors oder eines Syntax Errors in Zeile xy dar niederlegen – sie sind aber noch lange nicht die Fehlleistung des interaktiven in jedem menschlichen Prozess. Das isolierte Lernen zeigt paradoxerweise seine volle Wirksamkeit im perfekt-pädagogischen Wirken, dem geplanten und programmierten Unterricht, der endgültigen Absage von unlogischen Fragen und der klaren Antwort darauf: dem Syntax Error. Seine Antwort ist trivial aber ehrlich.

## Jenseits von Gut und Sehr Gut

---

Gibt sich der Lehrer selbst eine Note, wenn er einen seiner Schüler benotet? Betritt man das unscheinbare Gebäude nah am Stadtrand, möchte man nicht unbedingt davon ausgehen, dass hier ein pädagogischer Prozess geschieht. Natürlich, man sieht Lernprodukte - besonders große Plakate, die einen Themenkomplex mit vielen Schlagwörtern in den unterschiedlichen Schriftarten herzeigen. Das Moment der Beachtung liegt aber nicht auf

dem Inhalt, sondern eher die originäre Einzigartigkeit der vielen Buchstaben - die noch immer seriell und parallel verlaufen und zusammengenommen einen Sinn ergeben - zumindest auf den ersten Blick hin.

Nach außen gibt sich das Bildungsinstitut zurecht gönnerhaft. Alle Wesensmerkmale einer Heterotopie leuchten auf - Öffnungszeiten, örtliche Begrenzungen, Stockwerke, neue Medien, ein Fahrstuhl, ein Schullogo mit seinen vielen Corporate-Produkten im Anschlag und natürlich eine Schulordnung die nicht veröffentlicht wurde (und wenn, dann gut versteckt). Hat man doch das Schema F immer im Hinterkopf und reproduziert es sich nicht selbst von Schule zu Schule wieder aufs neue obgleich doch tausendmal schon gesehen. Man mag es einfach, wenn die Schuhe in Reih und Glied stehen. Das wusste schon Herbert Grönemeyer zu besingen. Und dass, architektonisch das Lehrerzimmer, das Sekretariat und die Direktion einer Schule doch immer in greifbarer Nähe sein sollten - geclustert auf den wenigen Quadratmetern die noch nicht von den Schüler\_innen okkupiert wurden, ist doch wunderschön, oder?

Ist es nicht Konvention, dass ein Schüler zuerst den vielbeschäftigten Verwaltungsapparat passieren muss um ein Gespräch mit der Direktion zu führen? Nicht so in dieser Bildungseinrichtung - hier, folgt man den durch die eigene Nase bestimmten logischen Weg muss man zuerst den Prinzipal überwinden um zum Sekretariat zu gelangen - welches sich durch zwei Charakterzüge auszeichnet. 1. Es scheint die gute Seele und 2. der einzige Fixpunkt in einem sich ständig rotierenden Universum zu sein. Alle anderen einfachen Lohnempfänger\_innen sitzen und brüten immer an den unterschiedlichsten Stellen im Gebäude welches, wäre es konsequent zu Ende gedacht, ein Labyrinth zwischen Räumen, Gängen und Stiegen und so will es auch der Zeitgeist - so diktiert es die Avantgarde der Wirtschaft.

Besprechungsraum A, B und C - am besten man bucht sie, wenn man sie braucht - das Klassenzimmer on demand. Die Computerräume sind immer offen und immer zugänglich - wieso auch nicht, schließlich verbinden sie Meetingraum II mit einer spartanisch aber als eine auf den Lehrervortrag ausgerichtete Unterweisungsraumeinheit. Die Computerarbeitsplätze sind so angereicht, das keiner während der Arbeit am PC zur Beamerleinwand blicken kann - so kann Informatikunterricht einfach nicht funktionieren bzw. pädagogisch gedacht kann nur so Informatikunterricht wirklich funktionieren - wenn der Info-Lehrer ausschließlich Hinterköpfe und Rückenansichten der Schüler\_innen sieht, Rektalunterricht/ Hektalunterricht in seiner Reinform.

Und im Vergleich zu seinem Pendant, zeichnet sich der gute Info-Unterricht durch die Schweigsamkeit des Lehrers aus - Zeiten in denen der Lehrer vorne Linear den einzigen wahren Weg zum Resultat vorzeigt und die Schüler\_innen diesen nach-klicken, sollte nur allein wegen einer progressiven und aufgeschlossenen Auseinandersetzung mit der Dynamik der Fachwissenschaft schon nicht gewagt werden. Naja, dieser Appell richtet sich definitiv nur an Medienpädagogen (in welcher Schulstufe und Form auch immer) welche eher auf das Produkt als nach dem Prozessprinzip hinarbeiten.

Die gänzliche Wahrheit des Schülerinnenprodukts ist ihre letzte Erscheinungsform und die dazu (eloquente) Präsentation. Und so muss sich der Medienpädagoge auch zurücklehnen und den Schüler\_innen volle Freiheit in den unterschiedlichsten Applikationen gewäh-

ren. Der und die Schüler\_in wird schon eine Lösung zum Problem finden - und mag die Lösung auch darin liegen, den Medienpädagogen um Hilfe zu bitten (eine ebenso legitime Problemlösung wie die Spracheingabe über die Google-App mit der passenden Tutorenlinkausgabe auf YouTube).

Und so soll auch der Informatikunterricht als Unterrichtsfach gar nicht Einzug finden in die Stundenpläne der Schüler\_innen - gänzlich sinnlos (ausgenommen sind höhere Schulformen, aber darauf komme ich noch später zu sprechen). Die Einrichtung eines eigenen Unterrichtsfach "Informatik" in Primär und Sekundarstufe I sollte nicht nur nicht angedacht werden, sondern muss aus pädagogischer Sicht vermieden werden. Der IKT-Bezug zu jedem Kompetenzfeld allein reicht vollständig aus um den jungen Menschen die ach so notwendigen IT-Fähigkeiten mitzugeben - mehr noch, wird der junge Mensch nicht schon in den für seine Entwicklung so wichtigen Jahren affektiv geprägt, die sich durch die Selbstwirksamkeit und der eigenständigen Problemlösungskompetenz ergibt, geformt und kann jener auch im Sinne der klassischen elektronischen Datenverarbeitung sicher nicht sofort die tieferen Grundlagen der Materie erkennen oder vielleicht erfassen, so kann er sich doch als ein IT-Profi fühlen - was mehr Fundament für eine weiterführende Ausbildung bietet, als jedes noch so mühsam erworbene ECDL-Wissen in der Sekundarstufe I tut. Deshalb brauchen wir für den Medienpädagogen in der Sek I ein anderes Berufsleitbild als es die Werteinheiten-Kolleginnen\_en haben. Mag es auch noch so trist und langweilig klingen, aber EDV-Arbeit ist nun mal nicht vergleichbar mit Löwenzähmung oder Extremsportarten.

Der gute Medienpädagoge lässt die Schüler\_innen arbeiten - überwacht sie nicht oder quält sie mit Ratschlägen - er\_sie greift nicht zur Maus bzw. der Tastatur des Schülers sondern versucht mit der Sprache eine Problemlösungsstrategie mitzugeben. In Folge ist dann der Medienpädagoge (und bitte ich meine beiderlei Geschlecht) nicht mehr nur auf eine einzige Klasse kapriziert sondern hat Kontakt mit eigentlich allen Schülern aus allen Schulstufen. Er\_Sie baut die Lehrer-Schüler-Beziehung auf die Affinität des Schülers auf. Fördert jene, die eine besondere IT Neigung besitzen.

Höre mir zu und klicke einfach das nächste mal auf genau jenen Button! Solche Simplifikationen die hinter jedem Tastendruck, jeder Mausbewegung, jedem Wischen am Display und [in ferner Zukunft] jedem Gedanken passiert, ein Grundlagenprozess einer Registersprache ist, in welcher die Information, das Input, die Eingabe einem Verarbeitungsprozess, einer Prozessorzuweisung, einer Kausalität unterworfen ist. Da sich im Kampf der vorherrschenden Logik sich die imperative Logik gegen die prädikative durchgesetzt hat, sind jene Befehlssätzen zum Dogma des Gegenwärtigen erwachsen. Nicht ohne uninteressanten Hintergrund für eine pädagogische Verwertbarkeit, da nun durch das Mittel "Logik" eine Kompetenzvielfalt vermittelt werden kann, die sich zwar am PC üben lässt aber vielmehr Auswirkungen auf eine zukünftige Berufung hat. Scriptet noch der junge Schüler eine Softwarelösung, so schreibt er in Wahrheit eine Konklusion von Befehlssätzen, die angereicht einer Progressionslogik entsprechen sollen, nein müssen.

Die Dunkelheit ist die Wahrheit - das Licht, ist Multiplikation von 1 und 0. Und will es uns auch noch so romantisch vorkommen, wenn wir den jungen Menschen die Fähigkeit einen Algorithmus zu formulieren vermitteln, so bleibt die Reflexion auf seine Kompromissfähigkeiten nicht unverhüllt - dafür schreibe ich! Es ist aber keine Wahrheit, die an einem

Tresen in einer zwielichtigen Bar den Eindruck hinterlässt: "Ich leide an der Erkenntnis der absoluten Wahrheit" - "Ich leide an meinem Wissen!". Es ist keine Wahrheit welche die Seele leiden lässt.

Zu aller Anfang, ermittelt der junge Info-Schüler was als Quelle wahr bzw. fake ist. Das kritische Bewusstsein auf ein all-gegenwärtiges und voll-verfügbares Aussagenkonklumerat und der Versuch die Wahrheit aus den vielen Text, Bildern, bewegten Bildern und gesprochen Text heraus-zu-destillieren. Keine Aufgabe der Informatik! Die Informatik kennt nur die booleschen Begriffe Wahr und Falsch und nicht Wahr und Fake. Da nun aber die vielen Schüler\_innen Produkte im Lauf der Zeit darauf ausgelegt sind, der nachfolgenden Schüler\_innen Generation eine einfachere Schulzeit zu beschern, wird die kritische Auseinandersetzung mit der Lehrstoffverteilung eines Informatik-Unterrichts immer diffiziler. Klassische und konservative Informatik-Lehrer\_innen (wie ich auch einer bin) portionieren den Lehrstoff über drei große Entitäten (den halbherzige Blick auf Lehrpläne des BMUKK bzw. den Deskriptoren von BIFI können der 'Didaktik der Informatik' verpflichtete Lehrerinnen unterlassen, weil ...) Betriebssystem, Softwareentwicklung und Spezifikation die einzig möglichen Ansätze sind. Das Betriebssystem unterteilt sich in Hardware und Software - mit anderen Worten, der schöne Power-Point Vortrag über die Festplatte, den Hauptspeicher über die Peripherie. Gesetze des Betriebssystems sind die Gesetze der plattformunabhängigen Übertragbarkeit auf das RGB-Lichtgeflecht des Monitors, welches die Basis für alles schafft.

Wie schon gehabt, übernehmen Sie einfach die Vorgaben ihres gratis Schulbuchs oder suchen Sie tatsächlich übereinstimmende Wahrheiten der Eingabemöglichkeiten (Maus, Tastatur oder Sprache) - das Betriebssystem entspricht dem Theater des Schauspiels, welches man zu interpretieren und im besten Fall zu beherrschen versucht. Alle großen Betriebssystem haben ihre Wahrheiten, die Schnittpunkte liefern. Selbst Windows öffnet über Ausführen --> cmd seine Konsole, von den Unix-Derivaten (Linux, OSx) kennen wir es sowieso - damit jedes Betriebssystem weit in seine eigene Vergangenheit greift und API's bereitstellt, die schon immer da waren. Für den Schüler sicher ein Faszinationsmoment. Der zweite Bereich: die Softwareentwicklung oder Programmierung bietet in rein kognitiver Hinsicht so viele Kompetenzen, die niemals nie im Mathematik-Unterricht allein vermittelt werden können. In der Scholastischen Mathematik gibt es nur eine Lösung für ein Problem die brav linear gelernt werden soll. Sie firmieren unter dem Betreff "Lösungsweg" - und kommt ein Schüler zwar zum richtigen Ergebnis, aber nicht unter Einbehaltung des vorgegebenen Lösungsweg, kann dieser schon mal ein "Nicht-Genügend" ernten. In der Softwareentwicklung wäre eine solche Bewertung des Schüler\_innen-Produkts eher kompliziert, weil es in der Softwareentwicklung viele Lösungswege gibt. Ist eine iterative Lösung eines Problems gleichwertig zu einer verlangten rekursiven Lösung, nur allein weil die Rekursion als elegant gilt? Ich sage ja, solange der Code geparsed wird. Solange das Ergebnis der Anforderung entspricht - tut sie es nicht, so entsteht das pädagogische Moment des Sinn stiftenden Dialogs, welches zwischen der Trinität Lehrer - System - Schüler auf das Plenum der Schüler erweitert werden sollte. In dritter Ebene findet man die Spezifikation - didaktische vorherrschende Literatur spricht hier oft von einer Bezugswissenschaft, deren die Informatik verpflichtet ist.

In einer Handelsakademie gelten andere Anforderungen wie sie für ein Gymnasium oder eine höhere technische Lehranstalt gelten. So sind in einem, die Nutzung von SAP, im anderen die Kenntnis von Prolog und im nächsten AutoCAD Kenntnisse von Nöten. Die Spezifikation ist immer eine Reduktion auf die intuitive Führung und das Nutzungspotential reduziert. Sie will definitiv nicht als einfach reduziert verstanden werden - sie ist aber determiniert und kann dadurch nur im Moment seine volle Entfaltung erfahren. So passiert es, dass ein Architekt über die Möglichkeiten von AutoCAD höher sprechen kann, als es von einem Informatikprofessors höchster Güte möglich ist. Kein Problem - in einer Handelsakademie kennt der Wirtschaftsprofessor mehrere werksseitige Funktionen von Excel - der Wirtschaftsinformatiker aber kennt die Möglichkeit eigene Funktionen zu scripten. SAP und Excel - alle Office-Produkte - das ist die Spezifikation der Ausbildung. Informatik als Hilfswissenschaft zu verstehen ist die didaktische Aufgabe des Info-Professors der Sekundarstufe II.

## Mit Imperativen [programmieren] lernen

---

Wer die Informatik verstehen möchte, muss sich zu aller erst etwas Bewusst machen. Alles was du weißt, ist Oberfläche - Vergänglichkeit - Flüchtig und immer nur ein Teilbereich vom Ganzen. Im Hintergrund - den man definitiv vertiefen kann - finden unzählige Translationsprozesse statt. Das Eine wird in das komplex-Tiefere übersetzt. Die Tiefe vertieft sich selbst weiter in die Überleitung zum gänzlichen Prozess.

Die Logik des Janus - vielleicht waren da einfach in seiner Jugend zu viele Imperative die er zu befolgen hatte. Jetzt aber, lieber junger Janus darfst du die Befehle geben - jetzt darfst du mit Imperativen programmieren!

Ich begleite den jungen Janus durch die imperative Programmierung und erkläre ihm die Möglichkeiten und Grenzen dieser Fähigkeit. Noch bevor ein paar Befehle an den Prozessor gesendet werden, sollte Janus sich über die Professionalität seines Handelns bewusst werden. Nun ist imPro\* nicht "Just Cause 4" oder "Hitman 2". Durch das Schreiben von hochwertigen Code wird nicht nur der Nimbus der Arbeit vermittelt, nein nein das Ludische wird zur Gänze vom Screen entfernt und schwarze Pixel werden wieder Schwarz\*\* - subjektiv betrachtet natürlich. Überhaupt, die Pixel kommen erst wirklich wieder zurück - dann wenn Janus ein Point-Set aus programmiert oder gar, Punkte über eine Gleichung der 3. Ordnung wandern lässt. Und dann sind sie alle wieder da - auch wenn es bereits 4000 über die Breite sind - aber mein Gott, der informatorische Pointillismus muss weiterleben - weiterleben durch dich, mein lieber Janus.

Allem Anfang liegt nicht immer nur ein Zauber inne. In der Pädagogik sind oft die Anfänge die größten Anstrengungen für den\_die planenden\_e Pädagogen\_in und die Quelle aller Herausforderungen für den\_die spontanen\_e Pädagogen\_in. So oder so - in der Informationspädagogik macht es keinen großen Unterschied ob nun das Ziel dem Ist-Zustand entspricht oder der Ist-Zustand das explizite Ziel darstellt. Im pädagogischen Prozess zählt wie so oft eigentlich nur das Jetzt-und-Hier. Und genau dieses hic-et-nunc ist auch das stärkste Argument für Lern-Ziel-Kontrollen, Prüfungen, Schularbeiten - alles was dem aufrichtigen Informatiker zu-wider sein sollte, da scripten und coden nichts spontanes

oder momentanes ist, sondern auf Gründlichkeit, Treue zur Logik und damit auf Immanenz ausgerichtet ist. Der kontemplative Prozess des Scripten von Code kann nicht in ein billiges 45-Minuten-Prüfungsszenario gepresst werden. So wie jede schriftliche Prüfung (egal welcher Gegenstand) absurd und unnötig ist - da sich Schreib- und Denkarbeit niemals der Zeit beugen darf.

Die mündliche Prüfung jedoch ist umso edler. Sie zeigt tatsächlich im Moment das Können - ob nun in der Fremdsprache der Weg zum Bahnhof gefunden, oder das Wissen über Photosynthese abgefragt wird - es geht immer dabei ums sophisticated-being. Dem jungen Janus ist dieser Umstand aber wohl eher unbedeutend - da er sowohl ein Recht als auch ein Bedürfnis hat, dass seine Leistungen bewertet und beurteilt werden.

Diese seine Arbeit ist nun in diesem Moment produktorientiert obgleich seine Leistung prozessorientiert sein sollte. Der\_die Informatik-Lehrer\_in muss also beides berücksichtigen - jedoch sollte sich Prozess und Produkt nicht durch Zeit bestimmen - wo dann die große Schwierigkeit entsteht: "Wie kann man einen Prozess im Jetzt bewerten?". Dem jungen Janus sollte im einfachsten Angesicht eines Editors nun Ehrfurcht vermittelt werden. Die Leere der Tabula-Rasa soll sich von Ehrfurcht in Respekt wandeln. Aus einer Welt der Reizüberflutung wird der junge Janus also in eine Welt der Geduld aber Unnach-sichtigkeit eingeführt. Die Ästhetik daraus ergibt sich von selbst.

Dieser neue Konstruktivismus ist natürlich durch eine Unmenge an Imperativen determiniert - und nun lernt also Janus tatsächlich die Regeln einer leeren Tafel - denn, das Werk muss einer Logik entsprechen und wenn es auch nur die Logik der Imperative ist. Das unlogische Werk in der Informatik ergibt in einer tieferen Translationsebene wieder Sinn. So soll es auch sein - da der Unsinn in der Informatik gegenwärtig nur Oberfläche sein kann - dahinter hat der eingelesene Informatiker nur ein Lächeln für den Versuch "Unsinn zu stiften" übrig. Janus soll zu diesem Lächeln hin-trainiert werden! Allem Anfang nun liegt ein Zauber inne! Und hier blickt der\_die Lehrer\_in in Janus Gesicht um zu Erfahren ob eine Eingabe die zu einer Ausgabe führt - irgend einen affektiv-pädagogischen Mehrwert generiert.

Für Janus nicht besonders - da seine Eingabe zu einer Ausgabe führt - ist für ihn eigentlich nur die doppelte Darstellung dessen, was er geschrieben hat. Und nun beginnt die im-Pro. Zwei Dinge lernt nun Janus. Erstens: Wird er einer Geliebten namens EVA vorgestellt, die zwischen der Eingabe und der Ausgabe eine Verarbeitung kennt. Zweitens: "Wird er auf die Möglichkeit, dass ein Anderer die Eingabe tätigt und dieser sowohl rational als auch irrational sein kann, instruiert." EVA soll ihm helfen, diesen Anderen zu verstehen. Die Welt des Janus erweitert sich enorm. Dieser Andere, er will so viel wissen. Er will wissen welcher Tag heute ist, er will wissen welche Verzinsung sich bei einem Eigenkapital von € 5.000,- und einem Prozentsatz von 3 % ergibt. Dieser Andere will Sachen wissen, die Janus unter Umständen interessieren könnte, aber eher im Moment irrelevant sind. Aber unser Janus ist ein guter Schüler. Ein guter Schüler weil er die Imperative und die Logik liebt und jede Möglichkeit als Eventualität betrachtet.

EVA hat also leichtes Spiel mit Ihm - wäre EVA ein Mensch. EVA ist aber die imaginäre Freundin von Janus, die von seinem\_ihrem Lehrer\_in zugeteilt wurde. Eine Ehe auf Zeit - bis zur Matura - eine Liebschaft bei jeder Prüfung - je nachdem wie der\_die Lehrer\_in Ja-

nus zur Traurigkeit oder zum Lachen zwingen will. Die imperative Logik ist in seinem Wesen ästhetisch einer andren Logik gleichzusetzen - sei es nun eine Prädikative oder eine Mathematische oder eine Fuzzylogic. Aus allen Erscheinungen lässt sich eine Informatik bilden - ob sie nun einfach nur auf der Booleschen Algebra basiert ist nebensächlich.

Eine Informatik kann auch rein deskriptiv sein - sie muss nicht nur Nullen und Einsen durch einen Prozessor jagen, der in seinem Wesen doch nur ein Allzu menschliches ist. So ist es ebenfalls denkbar, vom informatischen System eine Antwort auf eine Frage zu bekommen ohne das informatische System deklarativ aufzufordern eine Antwort zu geben - in Form von Imperativen.

Das Design eines Prozessor hängt weniger von seinen möglichen Zuständen als eher von der Art der Verwendung ab - und so bleibt gegenwärtig eine Eingabe nichts anderes als die Aufforderung Billiarden von Elektronen im Prozessor nach links oder rechts zu schupfen um so eine neue Anordnung von Elektronen zu erzeugen - die in ihrer Reihenfolgen Nullen und Einsen geordnet zurückgeben. Janus wird immer dem Ganzen Freude und Langeweile zugleich abgewinnen und die Nullen und Einsen werden zu Zeichen. Wird Janus jedoch in seinem ausgewachsenen Alter ein Offizier einer militärischen Armee, dann wird er Befehle in einer logischen Reihenfolge und in seiner Korrektheit richtig vermitteln können. Das hat ihm die imperative Programmierung gelehrt. Aber das affektive hat er wohl eher von "Just Cause 4", "Hitman 2" oder "Counter Strike" gelernt.

# Quellenverzeichnis

## Internetquellen und online Tools

---

Background Image Generator

<http://bg.siteorigin.com/>

css4.at Lernplattform

<https://www.css4.at>

dict.cc Wörterbücher

<https://www.dict.cc/>

Duden online

<https://www.duden.de/woerterbuch>

ExtractMetaData – Meta Daten auslesen

<https://www.extractmetadata.com/de.html>

flickr Fotocommunity

<https://www.flickr.com/>

Fundamentum – pädagogisches Forum

<http://fundamentum.xobor.de/>

Google (nahezu alle Dienste)

<https://www.google.at>

MDN web docs – Mozilla für Entwickler

<https://developer.mozilla.org/de/>

Mediaevent – CSS, HTML und JS mit {stil}

<https://www.mediaevent.de/>

openstreetmap

<https://www.openstreetmap.org>

Peter Kropff Tutorials

<https://www.peterkropff.de/>

pixabay Bilderplattform

<https://pixabay.com/de/>

php.net – Benutzerhandbuch

<https://www.php.net>

PHP-Einfach.de Tutorials

<https://www.php-einfach.de>

selfhtml.org – Die Energie des Verstehens

<https://selfhtml.org/>

stackoverflow – developers empowering

<https://stackoverflow.com/>

W3C CSS Validation Service

<http://jigsaw.w3.org/css-validator>

W3C Markup Validation Service

<https://validator.w3.org>

w3schools.com – Web Developer Site

<https://www.w3schools.com/>

Wikimedia Commons

<https://commons.wikimedia.org>

Wikipedia – die freie Enzyklopädie

<https://de.wikipedia.org>

## Literaturverzeichnis

---

Bauer, Joachim, Lob der Schule, Heyne Verlag, Hamburg 2008

Bernhart, Gottfried, Haberl, Klaus-Peter, Sachadonig, Günter, Einführung in die Personalverrechnung, Manz Verlag Schulbuch GmbH, Buch-Nr. 136189, Wien 2010

Brückner, Michael, Werbebriefe in Textbausteinen, Wirtschaftsverlag Carl Ueberreuter, Wien/Frankfurt 2002

Doralt (Hrsg.), Werner, Kodex Schulgesetze, LexisNexis Verlag, Wien 2010

Esteras, Remacha, Infortech – English for computer users, Cambridge University Press 2008

Fürst, Maria, Trinks, Jürgen, Philosophie, ÖVB Pädagogischer Verlag, SchBNr. 3380, ISBN 3-215-10085-1

Gaisbacher, Johann, Pongratz, Hanns Jörg (Hrsg.), Persönlichkeiten stärken, Leykam Graz, 2012

Glattauer, Niki, Der engagierte Lehrer und seine Feinde, Verlag Carl Ueberreuter, Wien 2010

Goleman, Daniel, Kaufman, Paul, Ray, Michael, Kreativität entdecken, Carl Hanser Verlag, München-Wien 1997

Goleman, Daniel, Soziale Intelligenz, Knaur, München 2008

Grbenic, Stefan A., Grbenic, Stefan, Einführung in das Steuer- und Beihilfenrecht, Manz Verlag Schulbuch GmbH, Buch-Nr. 136192, Wien 2009

Greimel, Bettina, Evaluation österreichischer Übungsfirmen, StudienVerlag GesmbH, Wien-Innsbruck 1998

Gruschka, Andreas, Verstehen lehren, Reclam Verlag, Stuttgart 2011

Gudjons, Herbert, Spielbuch Interaktions-Eziehung, Julius Klinkhardt, Bad Heilbrunn 2003

Haberl, Klaus-Peter, u. a., Rechnungswesen & Controlling HAK 2 bis 5, Manz Verlag Schulbuch GmbH, Buch-Nr. 120788, Buch-Nr. 125416, Buch-Nr. 131026, Buch-Nr. 136200, Wien 2009

Haberl, Klaus-Peter, u. a., Wirtschaftliches Rechnen HAK 1, Manz Verlag Schulbuch GmbH, Buch-Nr. 136198, Wien 2008

Hackenberg, Heide, Kommunikationsdesign Akquisition und Kalkulation, Verlag Hermann Schmidt, Mainz 2002

Hauser, Tobias, Wenz, Christian, PHP 7 und MySQL, Rheinwerk Verlag Bonn, 2019

Heckhausen, Heinz, Hoffnung und Furcht in der Leistungsmotivation, Verlag Anton Hain, Meisenheim am Glan, 1963

Hesse, Jürgen, Schrader, Hans Christian Schrader, Testtrainig Logik, Eichborn Verlag, Frankfurt am Main, 1999

Hüffel, Clemens, Reiter (Hrsg.), Anton, Handbuch Neue Medien, CDA Verlag, Perg 2008

Jacobsen, Jens, Website Konzeption, Addison-Wesley Verlag, München 2009

Jarosch-Frötscher, Carla, u. a., Businessstraining, Übungsfirma, Case Studies V HAK, Trauner Verlag, Linz 2008

Jarz, Thorsten, Grundlagen der Netzwerktechnik, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2011

Jarz, Thorsten, VB.NET Eine Einführung ins Programmieren, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2010

Jarz, Thorsten, Visual Basic für Applications in Excel, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2009

Jarz, Thorsten, Windows 7, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2010

Jarz, Thorsten, Windows 8, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2013

Jarz, Thorsten, Windows Server 2008, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2011

Jungmayer, J. R., Reproduktions und Druck Technik, Leykam-Verlag, Graz ISBN 3701112444

Klafki, Wolfgang, Neue Studien zur Bildungstheorie und Didktik, Beltz Verlag, Basel 1985

Kölbl, Doris, Hutz, Gabriela, Lehrerinnenarbeit – heute und morgen, StudienVerlag GsmBH, Innsbruck 1997

Kopeinigg, Christine, Textdesign 2007 Informations- und Officemanagement 1/I, Wien 2007

Kopeinigg, Christine, Textdesign und Publishing, Informations- und Officemanagement 1/I, Wien 2011

Laube, Michael, Einstieg in SQL, Rheinwerk Verlag Bonn, 2019

Lauffer, Dora, Der Weg zum erfolgreichen Unterricht, Weishaupt Verlag, Graz 1996

Liessmann, Konrad Paul, Theorie der Unbildung, Piper Verlag, München-Wien 2011,

Mitschka, Ruth, Die Klasse als Team, 1997

Mitschka, Ruth, So lernt man lernen, 1983

Moriz, Werner, Unterrichtswissenschaften, Books on Demand GmbH, Norderstedt 2007

Pesendorfer Robert u. a. Informationsmanagement Office 2010 I/1, Trauner-Verlag Linz, 2013

Pesendorfer Robert u. a. Informationsmanagement Office 2010 II/2, Trauner-Verlag Linz, 2013

Pesendorfer Robert u. a. Informationsmanagement Office 2010 III/3, Trauner-Verlag Linz, 2013

Peterßen, Wilhelm H., Kleines Methoden-Lexikon, Oldenbourg 2009

- Pflugfelder, Michael, Normativität und Professionalität in der Übungsfirma, Grin Verlag, Norderstedt 2007
- Posch, Peter, 9 x Partizipation – Praxisbeispiele aus der Schule, Verlag Carl Ueberreuter, Wien 2006
- Poser, Hans, Wissenschaftstheorie, Reclam Verlag, Stuttgart 2001
- Punkenhofer, Yvonne, Das Funktionieren des Übungsfirmenmarktes, VDM Verlag Dr. Müller, Saarbrücken 2011
- Schaub, Horst, Zenke, Karl, Wörterbuch Pädagogik, DTV München 2007
- Schischkoff, Georgi, Philosophisches Wörterbuch, Kröner Verlag, Stuttgart 1991.
- Schmidt, Siegfried J., Handbuch Werbung, LIT Verlag, Münster 2004
- Schneider, Wilfried, u. a., Betriebswirtschaft HAK 1 bis 5, Manz Verlag Schulbuch GmbH, Buch-Nr. 136141, Buch-Nr. 120806, Buch-Nr. 125368, Buch-Nr. 131041, Buch-Nr. 136143, Wien 2008
- Schubert, Sigrid, Schwill, Andreas, Didaktik der Informatik, Spektrum Verlag Heidelberg 2011
- Schulz-Reiss, Christine, Nachgefragt: Philosophie, Basiswissen zum Mitreden, Loewe Verlag GmbH, Bindlach 2005
- Seeborn, Joachim, Kompakt-Lexikon Werbepaxis, Verlag Gabler GmbH, Wiesbaden 2001
- Specht, Werner, Thonhauser Josef (Hrsg.), Schulqualität, Innsbruck 1996
- Stephan, Ingrid, Büroprozesse in der Übungsfirma, Bildungsverlag EINS, Köln 2011
- Stettner, Marko, Manipulation und Pädagogik, Leykam/Pädagogischer Verlag, Graz 1973
- Stumpf, Hildegard, Die wichtigsten Pädagogen, Marix Verlag GmbH, Wiesbaden 2007
- Theis, Thomas, Einstieg in PHP7 und MySQL 5.6, Rheinwerk Verlag Bonn, 2016
- Unterweger, Walter, Haberl, Klaus-Peter, Hauer, Thomas, Handbuch CRW Winline, Manz Verlag Schulbuch GmbH, Buch-Nr. 136214, Wien 2008
- Wabnegger, Erwin, Feindbild Lehrer, EDITION VA BENE, Wien-Klosterneuburg 2002
- Werneck, Tom, Die raffiniertesten Denkspiel, VPM Verlagsunion Pabel Moewig KG, Rastatt
- Whitehead, Alfred North, Wissenschaft und moderne Welt, Suhrkamp Taschenbuch Verlag, Frankfurt am Main 1984
- Winter, Felix, Grundlagen der Schulpädagogik Band 49: Leistungsbewertung, Schneider Verlag Hohengehren, 2012
- Winterhoff, Michael, Thielen, Isabel, Persönlichkeiten statt Tyrannen, Wilhelm Goldmann Verlag, München 2011