

CSS/JavaScript/jQuery

Trainer: Georg Praßl BSc.

➤ Georg Praßl

- Frontend-Developer
- Selbstständiger Trainer
- Berater
- Projektleiter



Die Teilnehmer stellen sich vor

- CSS
- Responsive Webdesign
- JavaScript
- DOM
- jQuery

- Zeiten
- Pausen
- Mittagessen
- Aufenthaltsbereiche

CSS



Cascading Style Sheet

Tag 1

➤ Grundlagen

- CSS-Selektoren
- Box-Model
- Layouttypen
- Schriftgrößen
- CSS-Frameworks

➤ Responsive Webdesign

- Media Queries
- Breakpoints
- Media Types
- Media Features
- CSS Transitions

- Cascading Style Sheets
- Trennung von Formatierung und Inhalt
- Zentrale einheitliche Formatierung
- Positionierung von Elementen
- <http://csszengarden.com/>

„HTML ohne CSS ist wie Pizza Hawaii – Funktioniert zwar ist aber nicht schön“


```
1. h1 {  
2.     color: white;  
3.     background-color: black;  
4. }
```

Selektor

Eigenschaft

Wert

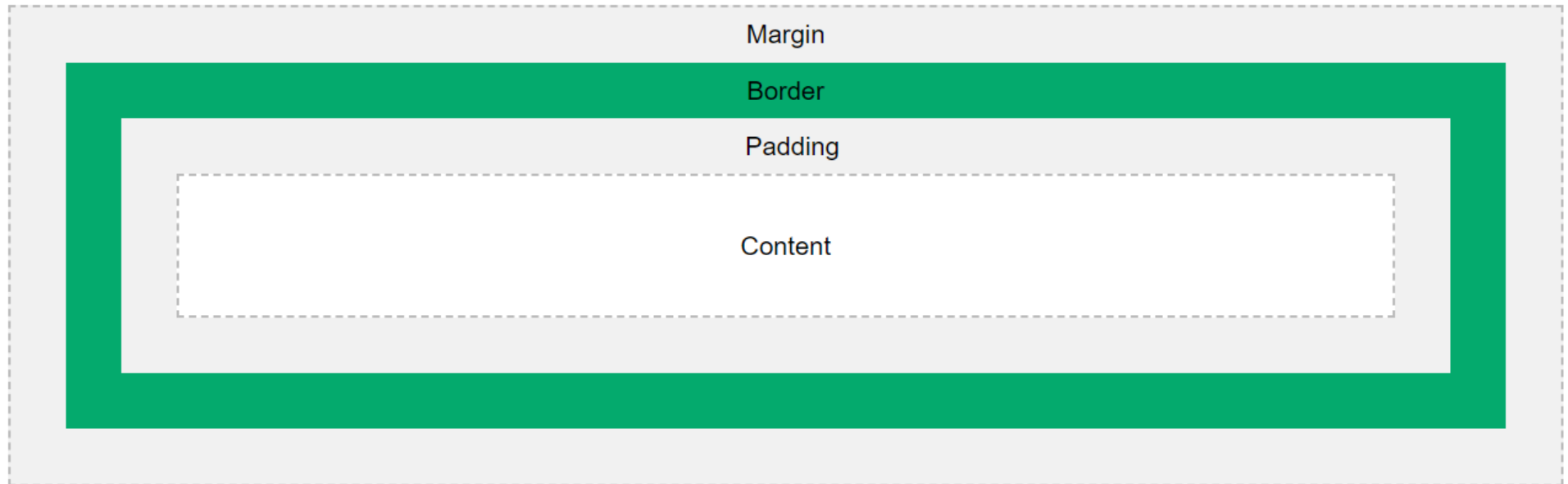
- Einfach Selektoren (h1, #id, .class)
- Kombinierte Selektoren (h1 > div > p)
- Pseudo-Klassen Selektoren (:hover, :link)
- Pseudo-Element Selektoren (::before, :: after)
- Attribute- Selektoren (a[target])

- $\text{div} > \text{li} \rightarrow \text{Kind}$
 - Spricht ein Element F an, wenn es Kind eines Elementes E ist
- $\text{div li} \rightarrow \text{Nachfahren}$
 - Spricht ein Element F an, wenn es Nachfahre eines Elementes E ist
- $\text{div} + \text{li} \rightarrow \text{Nachbarselektor}$
 - Spricht ein Element F an, wenn es direkter Nachbar eines Elementes E ist
- $\text{div} \sim \text{li} \rightarrow \text{Geschwisterselektor}$
 - spricht ein Element F an, falls F und E dasselbe Elternelement haben und F auf E im Elementbaum folgt.

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(<i>language</i>)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects <input> elements with no "required" attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a "readonly" attribute specified
:read-write	input:read-write	Selects <input> elements with no "readonly" attribute
:required	input:required	Selects <input> elements with a "required" attribute specified
:root	root	Selects the document's root element
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid value
:visited	a:visited	Selects all visited links

- `:hover` → Erkennt Mouse Over
- `:nth-child(n)` → Jedes nth Element wird gewählt.
- `:not(selector)` → Wählt jedes Element welches nicht gleich dem Selector ist
- `:valid` → Wählt alle Elemente mit einem gültigen Wert
- `:visited` → z.B. `a:visited` wählt alle bereits geklickten Links

Selector	Example	Example description
<u>::after</u>	p::after	Insert content after every <p> element
<u>::before</u>	p::before	Insert content before every <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of every <p> element
<u>::first-line</u>	p::first-line	Selects the first line of every <p> element
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user



- Normal flow
- Display property
- Flexbox
- Grid
- Floats
- Positioning
- Table layout
- Multiple-column layout

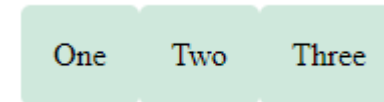

```
1. <p>I love my cat.</p>
2. <ul>
3.   <li>Buy cat food</li>
4.   <li>Exercise</li>
5.   <li>Cheer up friend</li>
6. </ul>
7. <p>The end!</p>
```

I love my cat.

- Buy cat food
- Exercise
- Cheer up friend

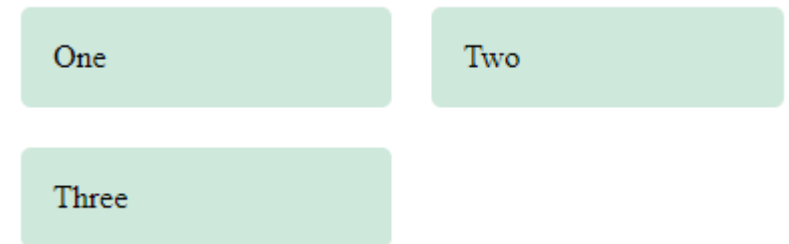
The end!

```
1. <style>
2.   .wrapper {
3.     display: flex;
4.   }
5. </style>
6. <div class="wrapper">
7.   <div class="box1">One</div>
8.   <div class="box2">Two</div>
9.   <div class="box3">Three</div>
10. </div>
```



- Display: Flex → aktiviert Flexbox
- Flex-direction: row/column/row-reverse/column-reverse → Ausrichtung der Flexbox
- Justify-Content: flex-start/flex-end/justify-space-between/justify-space-around/center → Ausrichtung der Elemente
- Align-Items: flex-start/flex-end/justify-space-between/justify-space-around/center → Ausrichtung der Elemente
- Flex: 1 → Auffüllen von Weißflächen
- Flex-wrap: wrap/nowrap/wrap-reverse

```
1. <style>
2.     .wrapper {
3.         display: grid;
4.         grid-template-columns: auto auto;
5.         gap: 20px 20px;
6.     }
7. </style>
8. <div class="wrapper">
9.     <div class="box1">One</div>
10.    <div class="box2">Two</div>
11.    <div class="box3">Three</div>
12.</div>
```



- Display: grid → aktiviert Grid-Layout
- Column-gap: 20px → 20px Leerraum zwischen den Spalten
- Row-gap: 20px → 20px Leerraum zwischen den Reihen
- Gap: 15px 20px → 15px Leerraum zwischen Reihen und 20px zwischen Spalten
- Grid-template-rows: 50px 50px → Zwei Reihen zu je 50px Höhe
- Grid-template-column: 50px 50px 50px → Drei reihen zu je 50px Breite

- Static
 - Element nur im normalen Flow der Website positioniert
- Relative
 - Element ist relative positioniert zur normalen Position → top,left,right,bottom um Element zu positionieren
- Fixed
 - Element ist relative zum Viewport positioniert → Scrollen ändert position nicht
- Absolute
 - Element ist relative zum nächsten Eltern-Element → Eltern-Element muss „relative“ sein sonst ist Body nächster Eltern-Teil
- Sticky
 - Position abhängig von Scrollposition

➤ Table grundsätzlich drei Elemente

- Table
- Th
- Td

➤ CSS-Eigenschaften:

- Border
- Width
- Border-collapse
- Text-align
- Vertical-align

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

- Pixel → 16px
- Em → 1em entspricht normalerweise 16px
 - Wie in Browser eingestellt
 - Um em zu berechnen: $\text{pixel}/16 = \text{em}$
- Vw → viewport width
 - Viewport ist die Browser-Window-Größe
 - $1\text{vw} = 1\%$ der viewport breite → wenn viewport 50cm, 1vw ist 0.5cm

- Tailwind → <https://tailwindcss.com/>
- Bootstrap → <https://getbootstrap.com/>
- Foundation → <https://get.foundation/index.html>
- Material Design for Bootstrap → <https://mdbootstrap.com/freebies/>
- Material Design → <https://material.io/develop/web/getting-started>
- W3schools → <https://www.w3schools.com/w3css/default.asp>

- CSS-Selektoren
- Box-Model
- Layouttypen
- Schriftgrößen
- CSS-Frameworks

- <https://flukeout.github.io/#> → CSS-Game
- <https://cssgridgarden.com/#de> → CSS-Grid-Game
- <https://flexboxfroggy.com/#de> → CSS-Flexbox-Game
- <http://www.hexinvaders.com/> → HEX-Color Game
- <http://www.flexboxdefense.com/> → CSS-Flexbox-Game
- <https://www.w3schools.com/css/default.asp>
- <https://wiki.selfhtml.org/wiki/Startseite>

RESPONSIVE WEBDESIGN



<https://codepen.io/greeny1992/pen/OJvzgpi>

- @media Regel
- Ändert Content der Website bei spezifizierten Breakpoints
- Elemente ändern in der:
 - Größe
 - Farbe
 - Reihenfolge
 - Erscheinung
 - ...

```
➤ /* Extra small devices (phones, 600px and down) */  
   @media only screen and (max-width: 600px) {...}  
  
   /* Small devices (portrait tablets and large phones, 600px and up)  
   */  
   @media only screen and (min-width: 600px) {...}  
  
   /* Medium devices (landscape tablets, 768px and up) */  
   @media only screen and (min-width: 768px) {...}  
  
   /* Large devices (laptops/desktops, 992px and up) */  
   @media only screen and (min-width: 992px) {...}  
  
   /* Extra large devices (large laptops and desktops, 1200px and up)  
   */  
   @media only screen and (min-width: 1200px) {...}
```

- All → Alle Endgeräte
- Print → Für die Druckvorschau
- Handheld → Typischerweise kleine Geräte mit wenig Bandbreite (veraltet)
- Screen → Alle Computer Bildschirme
- Speech → Für Screenreaders

- Werden in Media-Queries verwendet
- Aktuell ~35 Media Features
- Wichtigsten:
 - Min-widht
 - Max-widht
 - Min-height
 - Max-height
 - Hover

- Eigenschaft im CSS um Element dynamisch zu verändern
 - Transition → Shorthand Variation
 - Transition-delay
 - Transition-duration
 - Transition-property
 - Transition-timing-function
- Bsp.: <https://codepen.io/greeny1992/pen/dymJRqO>

- <http://www.cssplay.co.uk/>
- <http://csszengarden.com/>