

# AGENDA

## Curriculum WDD

4100\_07

REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE / QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK

CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT

POSITIONIERUNG / BOX-MODEL / FLEX

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT

BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

EINMITTEN / TABELLEN / IFRAME / FORMULAR

VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST

HELPER TOOLS / WORKFLOW

# WIE LERNE ICH EFFIZIENT?

## Kurzgesagt

- Lege chronologische resp. thematische Ordner auf Deinem Rechner an und lade das SAE-Übungsmaterial kurz vor Unterrichtsstart jeweils auf Deinen Rechner.
- Verwende ein elektronisches Lernjournal (z.B. Evernote, Docs), in welchem Du auch nach Stichworten suchen kannst.
- Über grundlegende Englisch-Kenntnisse musst Du verfügen.
- 10-Finger-Tastaturschreiben ist vorteilhaft.
- Denke an die Kontaktzeit von 25-30h/Woche.  
Nur durch zusätzliches Üben und Lernen wirst Du reüssieren.

# Repetition

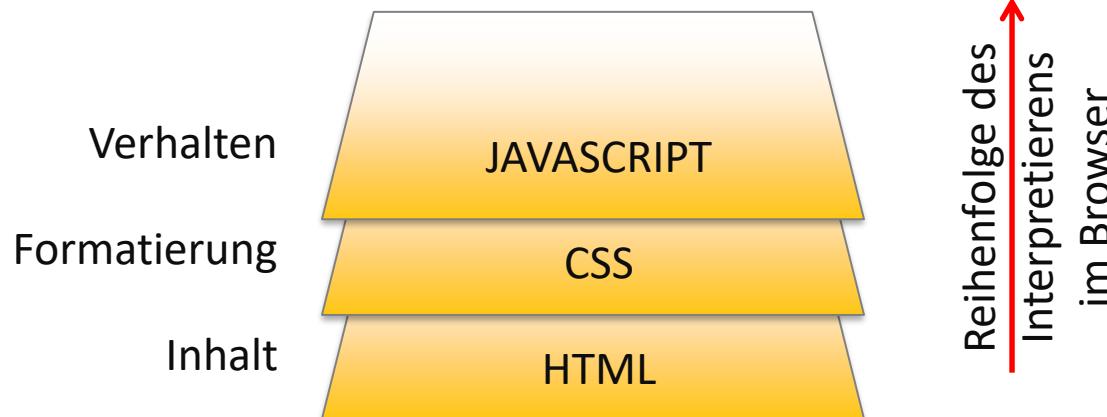
# DER BROWSER KANN 3 SPRACHEN

## Der sprachliche Aufbau einer Webpage client-seitig

Eine Webpage besteht aus 3 logischen Ebenen.

Jede Ebene braucht ihre eigene Sprache.

Der Server schickt Quellcode mit HTML, CSS und Javascript Code-Teile an den Browser, der diese interpretiert (englisch: Rendering).



# SPRACHEN

# WIR UNTERSCHIEDEN

## Auszeichnungssprache vs. Programmiersprache

Auszeichnungssprache (Markup)	Programmiersprache
<ul style="list-style-type: none"><li>• strukturiert und formatiert Inhalte</li></ul>	<ul style="list-style-type: none"><li>• trifft Entscheidungen nach bestimmten Bedingungen</li><li>• funktioniert nach folgendem Prinzip: Eingabe, Verarbeitung, Ausgabe</li></ul>
HTML (Inhalte)  CSS (Gestaltung)	JavaScript:  (clientseitige Scriptsprache für: Verhalten, Ereignisauslöser, Steuerung von Browserfenster, Rechnen, Teile einer Webpage nachladen, Cookie verwalten, Browserverlauf, URL Manipulation, Effekte, Animationen, DOM-Manipulationen von HTML+CSS )

# STATISCHE WEBSITE

# ERLÄUTERUNG

## Eine statische Website ...

- ist zustandslos (kein Gedächtnis, kein Erinnern von Requests)
- ist hardcodiert (Inhalte sind in den HTML-Files abgelegt)
- hat einzelne Webpages, die durch Navigieren im Browser aufgerufen werden können
- jede Webpage kann durch die gleiche zentrale CSS-Datei formatiert werden
- benötigt nicht PHP, da ihre Inhalte NICHT durch eine Datenbank gespiesen wird:  
sie ist somit NICHT dynamisch
- das dauerhafte Abspeichern erfolgt clientseitig:  
Cookies, JS, Sessions, Local Storage
- hat eine Start-Datei: index.html
- ist nur mit HTML/CSS-Kenntnissen verwaltbar

# SYNTAX VON HTML

# SYNTAX

## Start- und Endtag / Attribute / Wohlgeformter Code

Ein Starttag mit zwei Attributen, Texteinschub, sowie Endtag

```
<p id="mycontainer" class="fliessstext" >Ich bin Fliessstext im Paragraph. </p>
```

Attribut

Attribut

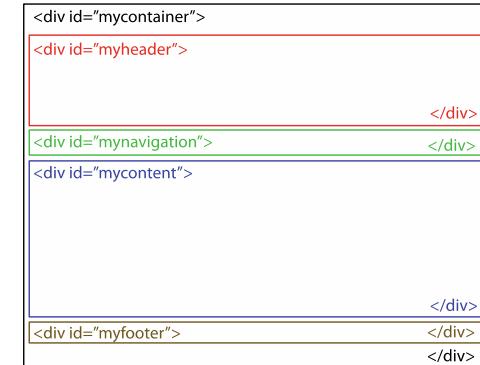
Inhalt

End-Tag

Start-Tag

Unter wohlgeformt versteht man das konsequente Abschliessen und Verschachteln von Tags

```
<section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>
```



# HTML-ELEMENTE – EINE FAMILIE

## Eltern, Kinder, Geschwister, Nachbarn, Nachfahren

< p > Ich bin Text

→ ist Mutter von em, em, del

< em > und zeitweise etwas schräg </ em >

→ ist Nachfahre von p

→ ist Kind von Eltern(p)

→ ist vorausgehender Nachbar von em

im Normalfall normal

→ ist Nachfahre von p

→ ist Kind von Eltern(p)

→ ist Geschwister von em

→ vorausgehender Nachbar von del

und ganz Ende bin

→ ist Nachfahre von p

→ ist Kind von Eltern(p),

→ ist Geschwister von em

→ ist Mutter von em

< del > immer

→ ist Nachfahre von p

→ ist Kind von Eltern(del)

< em > oft </ em > normal.

</ del >

→ Kind: damit sind Elemente der nächsten Einrückungsebene gemeint

→ Geschwister/Nachbarn: auf der gleichen Einrückungsebene

→ Nachfahren: alle Elemente aller Einrückungsebenen

</ p >

# GRUNDGERÜST

# EINE HTML-DATEI

## Aufbau (z.B. index.html)

eröffnendes Tag

```
<!DOCTYPE html>
<html lang="de-CH">
```

suchmaschinenrelevant  
für Reiter im Browser  
für Formatierung  
für Verhalten

Division = Platzhalter

Linkaufruf bei Klick

Heading 1 =  
Überschrift  
Paragraph

schliessendes Tag

```
  <head>
    <meta name="description" content="Die berühmte Maus">
    <title>Titel für den Reiter im Browser</title>
    <link href="css/meine_stile.css" rel="stylesheet" type="text/css"></link>
    <script type="text/javascript"> .....hier JS .... </script>
  </head>
```

```
<body>
  <div id="meine_navi">
    <a href="index.html">Home</a> <br/>
    <a href="galerie.html">Galerie</a>
  </div>
  <div id="inhalt">
    <h1>Über mich</h1>
    <p>Bin Fliesstext im Abschnitt.</p>
  </div>
</body>
```

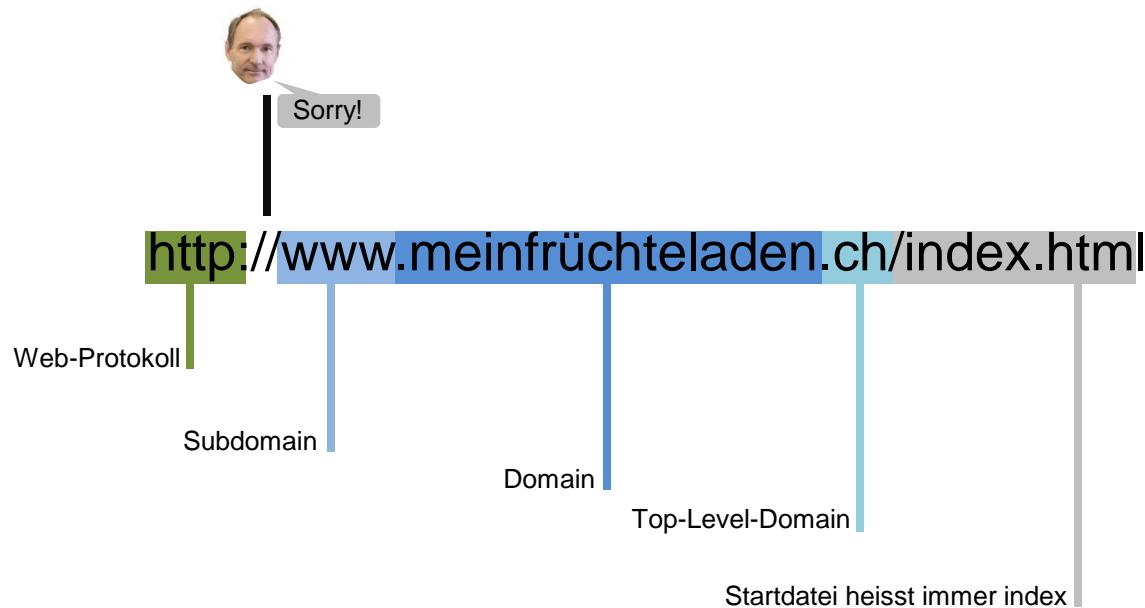
```
</html>
```

nicht sichtbar

sichtbar

# DER AUFRUF DER STARTDATEI

index.html wird standardmässig aufgerufen



# PROJEKT ANLEGEN

## im Brackets

### Navigieren zwischen mehreren html-Dateien

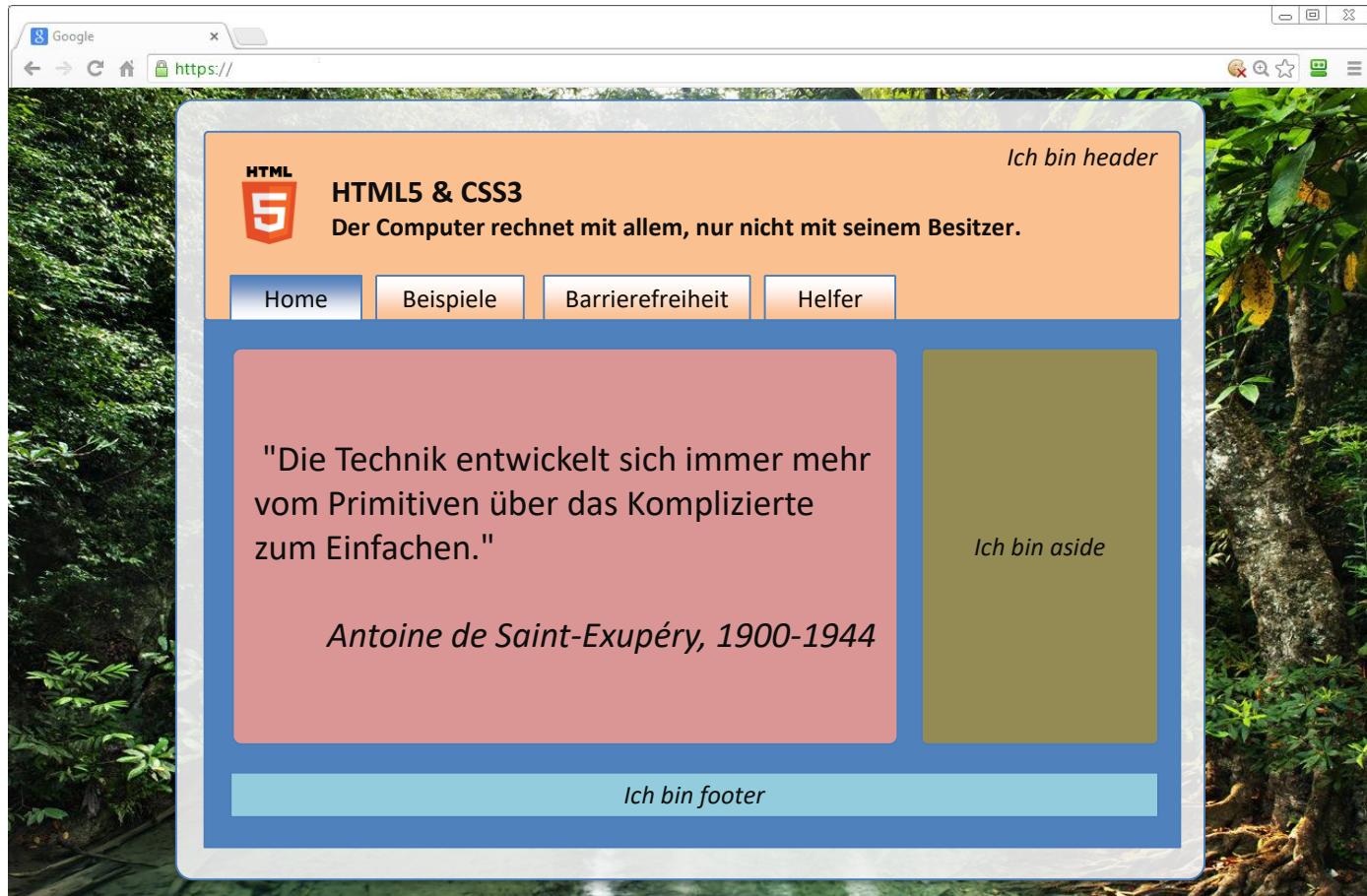
1. Die erste Unterseite erstellen:  
Vervielfältige einmal deine Datei index.html, benenne sie nach dem zweiten Navipunkt, verschiebe sie in den Ordner html und passe Pfade, Title-Tag, Navigation und etwas den Inhaltstext der beiden HTML-Dateien an. Navigieren zwischen den Navipunkten index.html und dem zweiten Navipunkt zwecks Kontrolle.
2. Weitere Unterseiten erstellen:  
Vervielfältige nun die Datei im html-Ordner gemäss den restlichen Navipunkten und passen jeweils noch die Navigation an, so dass nun zwischen allen Webpages mit der Navi navigiert werden kann. Passe den Platzhalter-Text im Main-Bereich etwas an.

### Abfüllen von Inhalten je html-Datei

1. Fülle den Content-Bereich weiter ab, mit Beispielen aus dem Unterricht und berücksichtige dabei immer die Barrierefreiheit.

# ZIEL

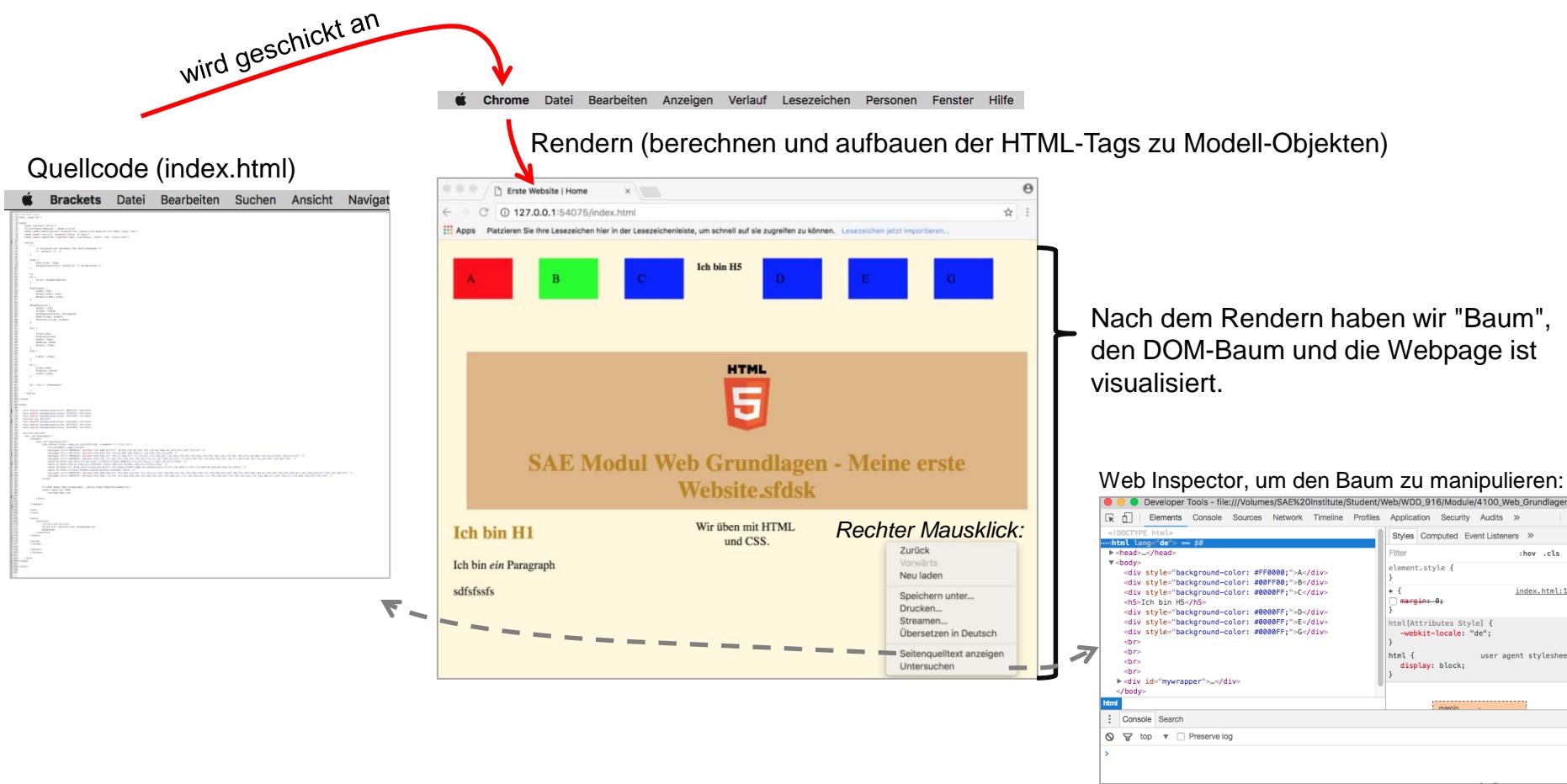
## Statische Website



# QUELLCODE, DOM und Web Inspector

# VOM QUELLCODE ZUM DOM

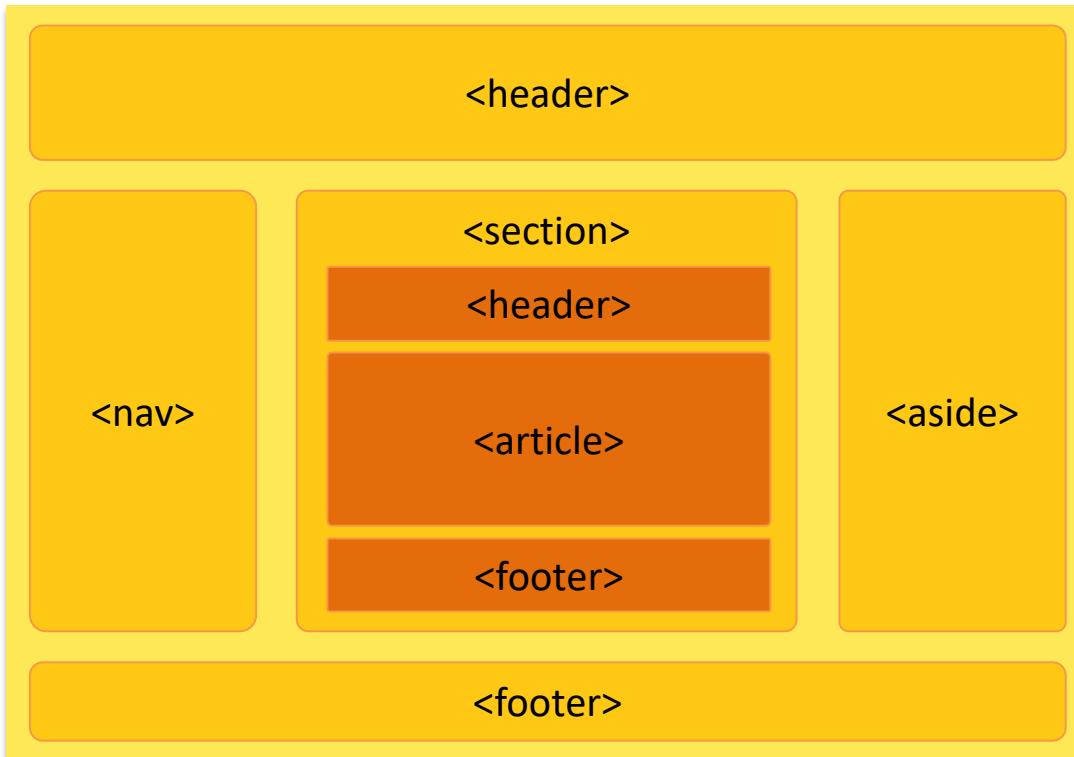
Der Browser liest den Quellcode zum Document Object Model (DOM) ein



# SEMANTIK

# SEMANTISCHE TAGS MIT HTML5

Deren Namen haben einen Bedeutung und dienen den Suchmaschinen und uns zur Orientierung.



Weitere semantische Tags:

<details>

<summary> (innerhalb von <details>)

<figure> zur Umklammerung von Bild + Erklärung

<figcaption> (Bilderkärung)

<main>

<address>

<mark> (hervorgehobener Text)

<time> (Datum/Zeitangaben)

<video>

<audio>

<svg>

<canvas>

# AGENDA

## Curriculum WDD

4100\_07

REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE / QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK

CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT

POSITIONIERUNG / BOX-MODEL / FLEX

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT

BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / EINMITTEN / TABELLEN / IFRAME / FORMULAR

VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST

REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# KLEINE REKAPITULATION ZU



## Repetition ist die Mutter aller Fertigkeiten

1. Was ist ein Browser im Web?  
Ist ein Anzeigeprogramm für Websites mit Navigationsmöglichkeit. To browser heisst stöbern.
2. Programmiersprachen vs. Auszeichnungssprache – wie hältst Du sie auseinander?  
Bei der erstenen werden Kontrollstruktur angelegt und Entscheidungen getroffen, bei der zweitenen werden Inhalte Inhalte strukturiert und formatiert.
3. Statische Website – was verstehst Du darunter?  
Ist datenbankunabhängig (hartcodiert), ist zustandslos,
4. Was verstehst Du unter Syntax beim Code schreiben?  
Eine fest definierte Schreibweise.
5. Aus was kann ein HTML-Tag bestehen?  
Start- und Endtage, Attribute, Inhalt
6. Welche HTML-Tags muss man in HTML5 nicht mehr zwangsläufig mit / schliessen?  
meta, br, img, link, doctype, input,
7. Auf was achtest Du bei der Ladereihenfolge von CSS-Stilvorlagen?  
reset => main => ausgabespezifisch => inline-style
8. Erläutere den Begriff Semantik in Bezug auf HTML  
Weist eine spezfische Bedeutung den Bestandteilen einer Website zu. .

# CSS

# FRONTEND

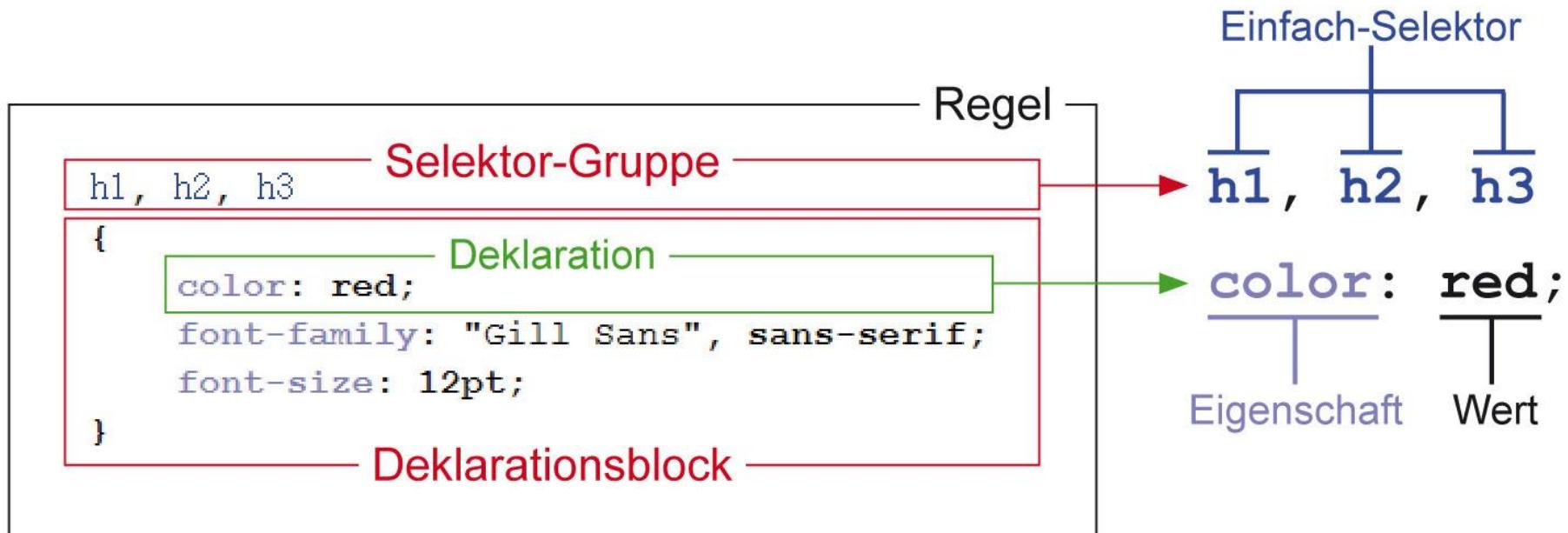
## CSS (Cascading Style Sheet)

- Formatvorlagen ermöglichen die modulare Gestaltung von Websites
- Form und Inhalt werden dadurch getrennt. WICHTIG!!!
- Unter Kaskade versteht man
  - Stilvorlagen werden auf unterschiedenen Ebenen definiert und können überschreiben.
  - Das spezifische Ansprechen eines Elementes
- Die Spezifikation (element, id=, class=, style="", !important) spielt eine wichtige Rolle.
- Die Anweisungen können auf folgende Weisen angelegt werden:
  - über eine separate Datei (z.B. main.css), welche in der HTML-Datei eingebunden wird. (eine separate Datei, die im xxx.html eingebunden wird)
  - innerhalb des HEAD-Tag: <style> .... CSS-Anweisungen ... </style>
  - als Attribut in einem HTML-Tag (inline-style):  
`<p style="color: red;">Ich bin rot.</p>`
- Zentrale, eingebundene Stylesheets ermöglichen schnellere Entwicklung
- Stiles übertragbar auf andere Displaygrößen
- Vererbung und Standardwerte: <https://www.w3.org/TR/CSS2/propidx.html>

# CSS – CASCADING STYLING SHEET

## Defintion

CSS-Regeln sind Stilvorlagen für HTML-Elemente und sind wie folgt aufgebaut:



# CSS EINBINDEN

# CSS EINBINDEMÖGLICHKEITEN

Ziel: Starke Trennung von Inhalt und Gestaltung

Typ	Beispiel
Als externe Datei	<link href="zentrales.css" rel="stylesheet">
Im Head als <style>Tag	<pre>&lt;style&gt;   p {     color: red;     font-size: 12px;   } &lt;/style&gt;</pre>
Inline als style= Attribut	<p style="color:#FF0000;font-size: 12px;">Ich bin rot.</p>

# CSS RESET

# CSS RESET

Damit alle Browser die gleiche Ausgangslage haben

Vor allem Innen- und Aussenabstände von Elementen werden genullt:

```
div {  
    margin: 0;  
    padding: 0;  
}
```

Empfehlenswerte Resets:

- Yahoo
- Meyer Web (in unserem Projekt)

# CSS MEDIA TYPES

# CSS MEDIA TYPES

## Übersicht

Medientypen sprechen verschiedene Ausgabemedien an und können über ein Attribut im HTML-Tag <link ...> oder als @media im CSS definiert werden.

### als HTML

```
<link rel="stylesheet" type="text/css"  
      href="stilvorlagen.css" media="all">
```

oder

```
<link rel="stylesheet" type="text/css"  
      href="stilvorlagen.css" media="screen">
```

oder

```
<link rel="stylesheet" type="text/css"  
      href="stilvorlagen.css" media="print">
```

oder

### als CSS

```
/* all ist Standard */  
@media all {  
    /* hier kommen CSS-Regeln */  
}
```

```
/* Stilvorlagen nur für Bildschirme */  
@media screen {  
    /* hier kommen CSS-Regeln */  
}
```

```
/* Stilvorlagen für Print-Funktionen */  
@media print {  
    /* hier kommen CSS-Regeln */  
}
```

# CSS SELEKTOREN und SPEZIFITÄT

# CSS - NORMALE SELEKTOREN

## Elemente, Klassen, ID's, Attribute

p { }	Alle p-Elemente
p, a { }	Alle a-Elemente und p-Elemente
p a { }	Alle a-Elemente innerhalb von p (alle Nachfahren) ***
.foo { }	Alle Elemente der Klasse 'foo'
div.foo { }	Alle div-Elemente der Klasse 'foo'
div#foo { }	Alle div-Elemente mit der id 'foo'
td div.foo { }	Alle div-Elemente der Klasse 'foo' innerhalb von td, d.h. innerhalb einer Tabellenzelle
a[title] { }	Alle a-Elemente, welche das Attribute-Eigenschaft title enthalten
a[title=klicken] { }	Alle a-Elemente, welche das Attribute title="klicken" enthalten

# CSS – KOMBINATORISCHE SELEKTOREN

## Kinder, Nachbarn, Geschwister

`p > em { }` Child selector : Alle em-Elemente, welche Kinder (erste Einrückungsstufe) von p sind

`h1 + p { }` Adjacent Sibling Selector :  
Das erste p-Elemente, welche als "Vor-"Nachbar h1 hat

`h1 ~ p{ }` General Sibling Selector:  
Alle p-Elemente, welche nachstehende Geschwister von h1 sind

# CSS – PSEUDOKLASSEN

## Spezielle nutzerbeeinflusste Zustände von Elementen

a:hover

a-Element bei Mauskontakt

a:focus

a-Element, wenn es mit fokussiert wird  
(reinklicken, mit Tabulator "reintaben")

a:visited

a-Element nach dem Anklicken

a:active

a-Element beim Mausklicken

a:first-child

das erste a-Element

a:nth-child(odd/even)

das ungerade/gerade a-Element

tr:nth-child(3n + 4)

jede 3. Zeile, zum ersten Mal ab Zeile 4

a:last-child

das letzte a-Element

# CSS PSEUDOELEMENTE

## Spezielle nutzerbeeinflusste Zustände von Elementen

p::first-line {}      Erste Zeile von p bis zum Umbruch durch p's Breite

p::first-letter {}      Erster Buchstabe von p

p::before {}      Inhalte vor p  
meist mit CSS-Regel: content: "\*"; oder content: url(bilder/pfeil.png);

p::after {}      Inhalte nach p  
meist mit CSS-Regel: content: "\*"; oder content: url(bilder/pfeil.png);

# CSS UNTERSCHIED ID UND KLASSEN

## Wichtig!

### Klassennamen, z.B. rot,

```
<style>
    .rot { color: red; }
    .durchgestrichen { text-decoration: line-through; }
</style>
```

### ID-Namen, z.B. statistikliste,

```
<style>
    #hauptnavi { margin: 20px 20px; }
</style>
```

im CSS-Teil

dürfen als Wert des HTML-Attribut *class*= im HTML-Code mehrmals auftauchen:

```
<span class="rot">Ich bin rot</span>
<span class="rot">Ich bin rot</span>
```



dürfen mit anderen Klassennamen verwendet werden:

```
<span class="rot durchgestrichen">
    Ich bin rot und durchstrichen </span>
```



im HTML-Teil

dürfen als Wert des HTML-Attributs *id*= im HTML-Code nur **1 x** auftauchen:

```
<div id="menue"> bla </div>
```



2 x wie hier unten? nein, nein, nein....

```
<div id="menue"> bla </div>
<div id="menue"> bla </div>
```

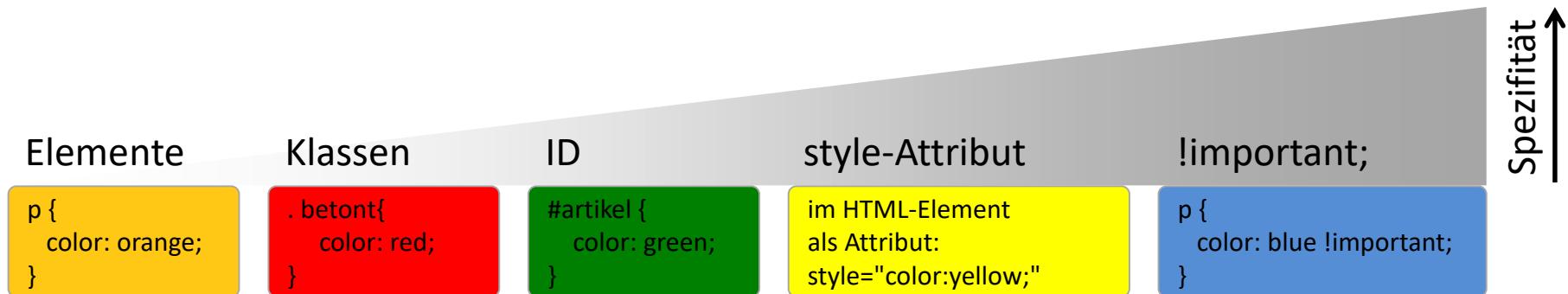


*id="menue" => der Wert "menue" darf nur einmalig vorkommen!*

# CSS SPEZIFITÄT

Kaskadieren heisst Regeln überschreiben,  
von allgemeinen zu sehr spezifischen.

*Spezifität* heisst im CSS: Welche CSS-Regel "gewinnt" am Schluss?



<p id="artikel" class="betont" style="color:yellow;">Welche Farbe hat dieser Zeitungsartikel?</p>

# HTML5SHIV

Da der IE8 HTML5-Tags nicht kennt, werden semantische Tags beim Browser-Rendern gerade wieder geschlossen.

Unbekannte HTML-Tags (wie HTML5) werden in älteren IE-Browser vom CSS nicht mehr richtig angesprochen.

Problem:

im Quellcode

<nav>



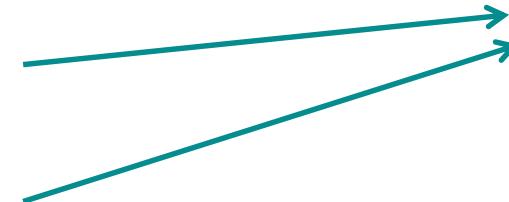
im DOM vom IE 8

</nav>



CSS Regel greift nicht mehr:  
nav {  
 color: red;  
}

</nav>



# CONDITIONAL COMMENTS

DAMIT LÄSST SICH BROWSERSPEZIFISCHER  
HTML-CODE SCHREIBEN

Siehe Dokumentation: <http://www.quirksmode.org/css/condcom.html>

Conditional Comments sehen ähnlich aus wie ein Kommentar-Tag:

```
<!--[if lt IE 9]>
```

```
  <p>Ich werde im IE 8 angezeigt.</p>
```

```
<![endif]-->
```

# MASSEINHEITEN

# MASSEINHEITEN IM CSS

## Absolute und relative Einheiten

### Relative Größen

px	Pixels (relative zur Anzeigedichte des jeweiligen Ausgabemedium, früher lange absolut) z.B. unser Bildschirm hätte eine Auflösung von 96px/inch => 254mm/96px → unformatierter Text innerhalb vom Body-Tag hat 16px.
em	relative Größe zur jeweiligen Schriftgröße vom Element (1em = size of the current font-size of the element itself)
rem	relative Größe zur jeweiligen Schriftgröße von body/html (root) (4em = 4 x size of the current font-size of the body/html-Tag)
%	relativ zur aktuellen Größe oder zum Elternelement
vh	1% of Viewport Height
vw	1% of Viewport Width

# HAUSAUFGABEN

# NEUSTES WISSEN ERARBEITEN

## Hausaufgaben auf nächste Woche

### Auftrag 1 – Marktanteile Browser

Ermittle die gegenwärtigen Marktanteile der Browser nach Schweiz, Europa, Weltweit sowie nach Betriebssystemen.

### Auftrag 2 – Verhältnis von Anfragen über Desktop und Mobile Geräte

Ermittle oben erwähntes Verhältnis in der Schweiz, in Europa und global.

### Auftrag 3 – Demographische Nutzungsdaten

Ermittle oben erwähnte Daten im Hinblick auf Altersklassen nach Browsern. Schweiz/Europa/Global

### Auftrag 4 – HTML5. Finde heraus, ab welcher Version die gängigsten Browser HTML5 generell unterstützen.

Ermittle oben erwähnte Daten im Hinblick auf Altersklassen nach Browsern.

### Auftrag 5 – Doctypes: HTML5, HTML 4.01 und XHTML

Ermittle den aktuellen Stand in Bezug auf ihre Verbreitung.

### Auftrag 6 – Legacy: früher waren Framesets und Tables für ganze Websites verwendet worden.

Zeige ein kleines Frameset (Quellcode und gerendert) und ermittle, warum sie fast nicht mehr anzutreffen sind.

Bitte Ergebnisse auf max. 3 Slides mit Präsentation von max. 3 Minuten

# AGENDA

## Curriculum WDD

4100\_07

REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE /  
QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK  
CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT /  
MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT  
POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT  
BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / EINMITTEN / TABELLEN / IFRAME / FORMULAR  
VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST  
REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# KLEINE REKAPITULATION II ZU



## Repetition ist die Mutter aller Fertigkeiten

1. Die Spezifität bezüglich CSS-Selektoren und Schreibweisen nimmt in folgender Reihenfolge zu:
2. Welche Eindindungsmöglichkeiten gibt es für CSS?
3. Was ist der Unterschied zwischen id= und class= bezüglich CSS und nenne typischen Anwendungsfälle
4. Was sind Pseudoklassen?
5. Was sind Pseudoelemente?
6. Welche Media Types bezüglich CSS gibt es?
7. Welche Masseneinheiten gibt es?
8. Welch Masseneinheiten gibt es in CSS?
9. Ganz generell: welche Eigenschaften werden eher vererbt, welche nicht?

# FARBCODES

# HEX UND RGB CODE FÜR FARBEN

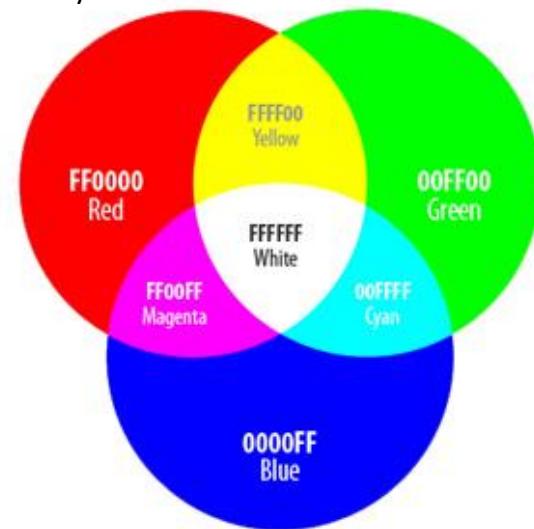
## Möglichkeiten für einen Farbwert im CSS

HEXADEZIMALER CODE<
• color: # <b>000000</b> ; (schwarz)
• color: # <b>FFFFFF</b> ; (weiss)
• jede Stelle hat 16 Abstufungen von: <b>0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F</b>
$\underbrace{16 \times 16}_{256} \times \underbrace{16 \times 16}_{256} \times \underbrace{16 \times 16}_{256} = 16,7 \text{ Mio. Farben}$
• keine Transparenzangaben

RGB CODE
• color: <b>rgba(255, 0, 0, 0.3);</b> 0 – 255 => 256 $\alpha$ -Kanal: 0 (transparent) – 1 (deckend)
• color: <b>rgba(100%, 0%, 0%, 0.3);</b>

zur Nachlese: <https://de.wikipedia.org/wiki/Webfarbe>  
zur Umrechnen: [http://html-color-codes.info/webfarben\\_hexcodes/](http://html-color-codes.info/webfarben_hexcodes/)  
<http://www.colorsphere.com/rgb-color-wheel/>  
in den Graphik-Programmen / Brackets

additives Farbsystem:



# INLINE- UND BLOCKELEMENTE

# INLINE- UND BLOCK-ELEMENTE

## Default-Werte von display: \_\_\_\_\_

Block Elemente
<ul style="list-style-type: none"> <li>Strukturierende Elemente: div, p, form, ul , li, h1, semantische Tags</li> <li>Werden <b>so breit wie möglich</b>, d.h. gleiche Breite wie Eltern (erwirken immer einen Zeilenumbruch)</li> <li>Werden <b>so hoch wie nötig</b>.</li> <li>können padding und margin Eigenschaften haben</li> </ul> <div style="display: flex; justify-content: space-around;"> <div style="background-color: #8B8B45; width: 40px; height: 20px; margin: 5px;"></div> <div style="background-color: #8B8B45; width: 40px; height: 20px; margin: 5px;"></div> </div>

Inline Elemente
<ul style="list-style-type: none"> <li>Normalerweise textbasierte Elemente: span, a, em, b, cite, code, strong, label</li> <li><b>Ingorieren width und height-Angaben</b> (Höhe durch line-height). <b>Kein Absatz</b>.</li> <li>Können auch als "Flusselemente" bezeichnet werden, da sie mitten im (Text)-Fluss auftauchen können.</li> </ul> <p style="font-family: monospace;">Lorem ipsum <b>SBB</b> dolor sit amet, te <i>schräg</i> erat illum inimicus, ex eum discere diceret. Eum cu alienum constituo cotidieque, qui meis <b>utroque</b> tibique eu. Ea <i>Zitat</i> mel ipsum intellegebat, in mei offendit <b>Code-Schnipsel</b>, nec postulant</p>

Inline-Block Elemente
<ul style="list-style-type: none"> <li>Können eine <b>Höhe und Breite</b> haben, jedoch <b>keinen Absatz</b>.</li> <li>Können auch margin und padding enthalten.</li> <li>&lt;img ....&gt; – ein typisches Beispiel.</li> <li>Sind auch Flusselemente, also können mitten im Fluss – z.B. mitten im Text - auftauchen.</li> </ul> <p>Ich bin ein Paragraph und habe hier  ein Bildchen.</p>

# CSS PROPERTY DISPLAY

Du kannst die Default-Werte auch überschreiben:

- Default Values und Vererbung:
  - HTML-Elemente: [http://www.w3schools.com/cssref/css\\_default\\_values.asp](http://www.w3schools.com/cssref/css_default_values.asp)
  - CSS-Eigenschaften: <https://www.w3.org/TR/CSS21/propidx.html>
- Mit CSS kann ich jedoch jedem Element auch explizit sagen, wie es sich verhalten soll (überschreiben):
  - display: inline;
  - display: inline-block;
  - display: block;
- Häufig zum Beispiel bei einem Listen-Element:

```
<ul>
    <li>Navipunkt 1</ul>
    <li>Navipunkt 2</ul>
</ul>
```

# FLOAT

# CSS FLOAT FÜR BLOCKELEMENTE

## Blockelement umfliessen lassen

**Umfluss starten:**

float: none (**default**) | left | right | inherit

**Umfluss stoppen:**

clear: none (**default**) left | right | both | inherit;

**Tipp: mit einer CSS-Klasse den Umfluss stoppen.**

**Auf jenes HTML-Element anwenden, welches wieder im normalen Fluss sein sollte.**

.cl { clear: both; }

*oder auch*

.cl { overflow: auto; }

# AGENDA

## Curriculum WDD

4100\_07

REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE /  
QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK  
CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT /  
MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT  
POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT  
BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / EINMITTEN / TABELLEN / IFRAME / FORMULAR  
VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST  
REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# POSITIONIERUNG

# CSS EIGENSCHAFT POSITION

## Wonach wird ein Element positioniert?

### position: static;

- **default-Wert!**
- im normalen HTML-Fluss
- CSS top, right, bottom, left reagieren nicht, nur margin!

### position: relative;

- Verschiebung, gemessen an der Normalposition oder Anfangsposition des Elements selbst.
- CSS top, right, bottom, left werden aktiviert.
- Aktiviert z-Achse des Elements.

### position: absolute;

- Positionierung, gemessen am Rand des nächsthöheren Vorfahrenelements, also umspannende Element, das die Position relative oder absolute aufweist.
- Das Element wird aus dem HTML-Tag-Fluss ausgelöst.
- Letzter Fallback der Ausrichtung (sozusagen das letzte Eltern-Element) wäre: <body>
- CSS top, left, right, bottom werden aktiviert.
- Aktiviert z-Achse des Elements.

### position: fixed;

- Positionierung, gemessen am "Viewport", d.h. am Browserfenster.
- Bleibt beim Scrollen stehen.
- Das Element wird aus dem HTML-Tag-Fluss ausgelöst.
- CSS top, left, right, bottom werden aktiviert.
- Aktiviert z-Achse des Elements.

„nicht positioniert“

„positioniert“

# CSS POSITION STATIC

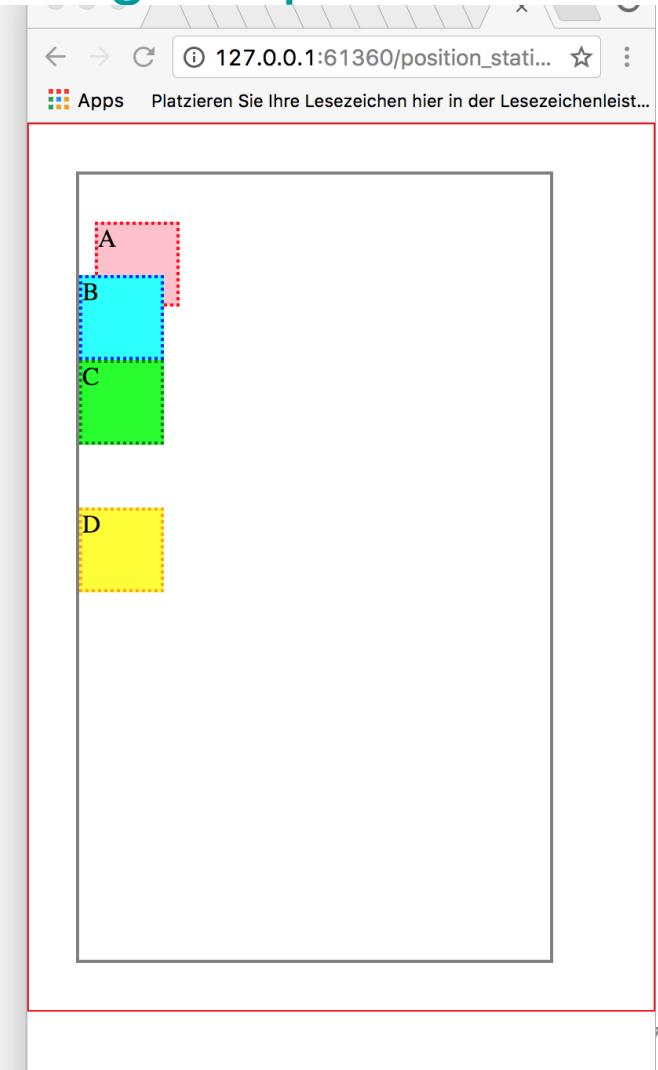
Zum Spicken

## Erlaubte CSS Eigenschaften top/margin-top/z-index

**position: static;**

- **default**
- **margin +/-**
  - z.B. margin-top: 10px; =>  
Element verschiebt sich  
und nachfolgende  
Elemente nach unten.

```
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4      <meta charset="utf-8">
5      <title>Position Static => margin ok</title>
6      <style>
7          * { margin: 0; padding: 0; }
8          body { border: 1px solid red; }
9          div#wrapper { border: 2px solid grey; width:300px; height:500px; margin: 30px; }
10         div#wrapper > div { width: 50px; height: 50px; }
11         div#wrapper div#a { border: 2px dotted red; background: pink; margin-top:30px; margin-left: 10px; }
12         div#wrapper div#b { border: 2px dotted blue; background: cyan; margin-top: -20px; }
13         div#wrapper div#c { border: 2px dotted green; background: lime; margin-top: 0px; }
14         div#wrapper div#d { border: 2px dotted orange; background: yellow; margin-top: 40px; }
15     </style>
16 </head>
17 <body>
18     <div id="wrapper">
19         <div id="a">A</div>
20         <div id="b">B</div>
21         <div id="c">C</div>
22         <div id="d">D</div>
23     </div>
24 </body>
25 </html>
```



# CSS POSITION ABSOLUTE

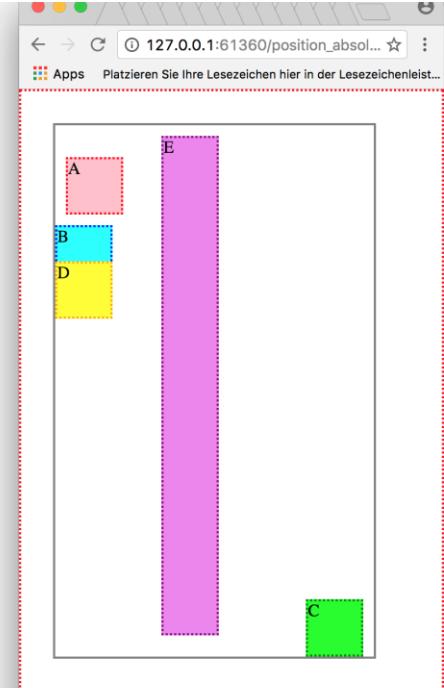
Zum Spicken

## Erlaubte CSS Eigenschaften top/margin-top/z-index

### position: absolute;

- margin +/-
  - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.
- top|bottom|left|right +/-
  - z.B. top: 10px; => Element verschiebt nur sich nach unten.
- z-index +/-
  - z.B. ich kann damit die Elemente anordnen auf der z-Achse. Default ist: auto.
- height/width: auto; und top/bottom/left/right
  - ich kann das Element zum Rand des nächstpositioniereten Elternelements aufziehen .

```
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4      <meta charset="utf-8">
5      <title>Position Absolute mit den Elementen C und E </title>
6      <style>
7          * { margin: 0; padding: 0; }
8          body { border: 2px red dotted; }
9          div#wrapper { border: 2px solid grey; width:300px; height:500px; margin: 30px; position: relative; /* Dieses Elternelement ist NICHT static */ }
10         div#wrapper > div { width: 50px; height: 50px; }
11         div#wrapper div#A { ... }
12         div#wrapper div#B { ... }
13         div#wrapper div#C { border: 2px dotted green; /* BEISPIEL MIT ABSOLUTE - am Elternrand kleben */
14                                background: lime;
15                                position: absolute; /* Distanzen in dieser Regel richten sich nach dem nächsthöheren Element, welches relative, absolute, fixed, und falls jenes Elternelement nicht vorhanden ist, dann body-Tag. Hier ist es der wrapper, der positioniert wurde. */
16                                margin-left: 0px;
17                                margin-top: 0px;
18                                margin-bottom: 0px;
19                                right: 10px;
20                                bottom: 0px;
21                                z-index: 1000; /* z-Achse möglich! */
22         }
23         div#wrapper div#D { ... }
24         div#wrapper div#E { border: 2px dotted purple; /* BEISPIEL MIT ABSOLUTE - aufgezogen */
25                                position: absolute; /* Mit absolute und top/bottom und height:auto kann ich ein DIV auch strecken zum Rand des Elternelements! */
26                                background: violet;
27                                margin-top: 0px;
28                                margin-bottom: 0px;
29                                top: 10px;
30                                bottom: 20px;
31                                left: 100px;
32                                height: auto;
33         }
34     </style>
35 </head>
36 <body>
37     <div id="wrapper">
38         <div id="A">A</div>
39         <div id="B">B</div>
40         <div id="C">C</div>
41         <div id="D">D</div>
42         <div id="E">E</div>
43     </div>
44 </body>
45 </html>
```



# CSS POSITION RELATIVE

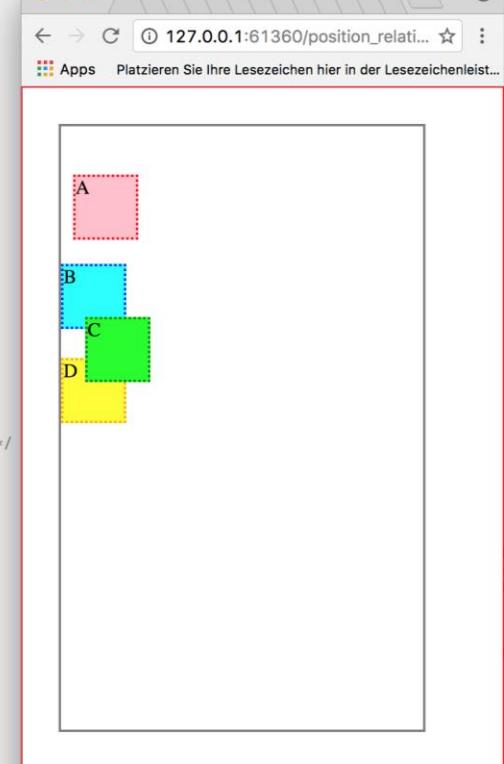
Zum Spicken

## Erlaubte CSS Eigenschaften top/margin-top/z-index

### position: relative;

- margin +/-
  - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.
- top | bottom | left | right +/-
  - z.B. top: 10px; => Element verschiebt nur sich nach unten.
- z-index +/-
  - z.B. ich kann damit die Elemente anordnen auf der z-Achse.  
Default ist: auto.

```
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4      <meta charset="utf-8">
5      <title>Position Relative</title>
6      <style>
7          * { margin: 0; padding: 0; }
8          body { border: 1px red solid; }
9          div#wrapper {
10              border: 2px solid grey;
11              width:300px;
12              height:500px;
13              margin: 30px;
14          }
15          div#wrapper > div {
16              width: 50px;
17              height: 50px;
18              position: relative; /* RELATIVE für alle DIV-Kinder */
19          }
20          div#wrapper div#A {
21              border: 2px dotted red;
22              background: pink;
23              margin-left: 10px; /* margin => verschiebt sich */
24              margin-top:30px; /* und nachfolgende Elemente */
25              margin-bottom: 0px;
26              top: 10px; /* top/bottom/left/right => Verschiebung */
27              bottom: 0px; /* des Elements selber. Nachfolgende
28                           Elemente bleiben unbeeindruckt. */
29          }
30          div#wrapper div#B {
31              border: 2px dotted blue;
32              background: cyan;
33              margin-top: 10px;
34              margin-bottom: 0px;
35              top: 20px;
36              bottom: 0px;
37          }
38          div#wrapper div#C {
39              border: 2px dotted green;
40              background: lime;
41              margin-left: 20px;
42              top: 10px;
43              bottom: 0px;
44              z-index: 1000; /* z-Achse möglich! */
45          }
46          div#wrapper div#D {
47              border: 2px dotted orange;
48              background: yellow;
49              margin-top: -20px;
50              top: 10px;
51      }
52      </style>
53  </head>
54  <body>
55      <div id="wrapper">
56          <div id="A">A</div>
57          <div id="B">B</div>
58          <div id="C">C</div>
59          <div id="D">D</div>
60      </div>
61  </body>
62  </html>
```



# CSS POSITION FIXED

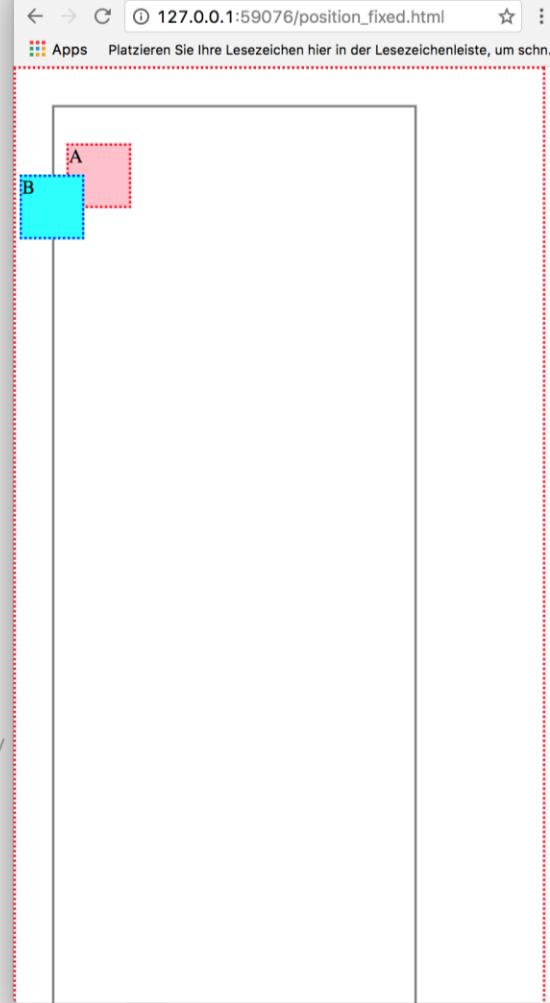
Zum Spicken

## Erlaubte CSS Eigenschaften top/margin-top/z-index

### position: fixed;

- margin +/-
  - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.
- top|bottom|left|right +/-
  - z.B. top: 10px; => Element verschiebt nur sich nach unten.
- z-index +/-
  - z.B. kann ich damit die Elemente anordnen auf der z-Achse. Default ist: auto.
- height/width: auto; und top/bottom/left/right
  - z.B. kann ich damit das Element zum Rand des nächstpositioniereten Elternelements aufziehen .

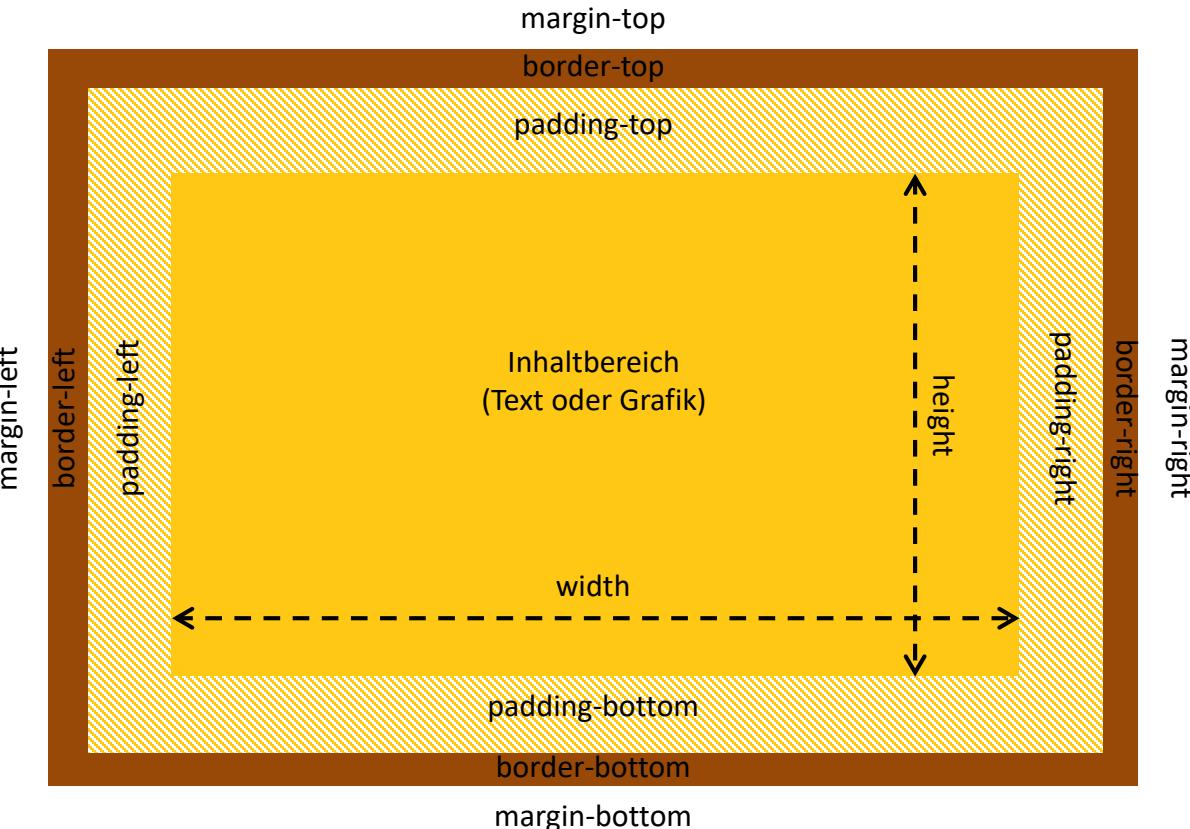
```
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4      <meta charset="utf-8">
5      <title>Position FIXED - ab dem Viewport gemessen - siehe DIV B</title>
6      <style>
7          * { margin: 0; padding: 0; }
8          body { border: 2px solid grey; width:300px; height:450px; margin: 30px; }
9          div#wrapper > div { width: 50px; height: 50px; }
10         div#wrapper div#a {
11             border: 2px dotted red; background: pink; margin-left: 10px; margin-top:30px; margin-bottom: 0px; top: 10px; bottom: 0px; /* z-Anordnung: DIV A wird dann über DIV B gelegt: */ /* position: relative; z-index: 100000; */
12         }
13         div#wrapper div#b {
14             border: 2px dotted blue; background: cyan; margin-top: 10px; margin-bottom: 0px; /* Scroll nach unten im Browser für den Effekt */ position: fixed; /* fixed + top/left/bottom/right */ top: 80px; left: 5px; z-index: auto; /* default, könnt auch Nummer vergeben */
15             /* auch hier könnten ich die Höhe wieder max. ausdehnen: */ /*bottom: 20px; height: auto; */
16     }
17     </style>
18 </head>
19 <body>
20     <div id="wrapper">
21         <div id="A">A</div>
22         <div id="B">B</div>
23     </div>
24 </body>
25 </html>
```



# BOX-MODEL

# BOX-SIZING

## CONTENT-BOX VS. BORDER-BOX

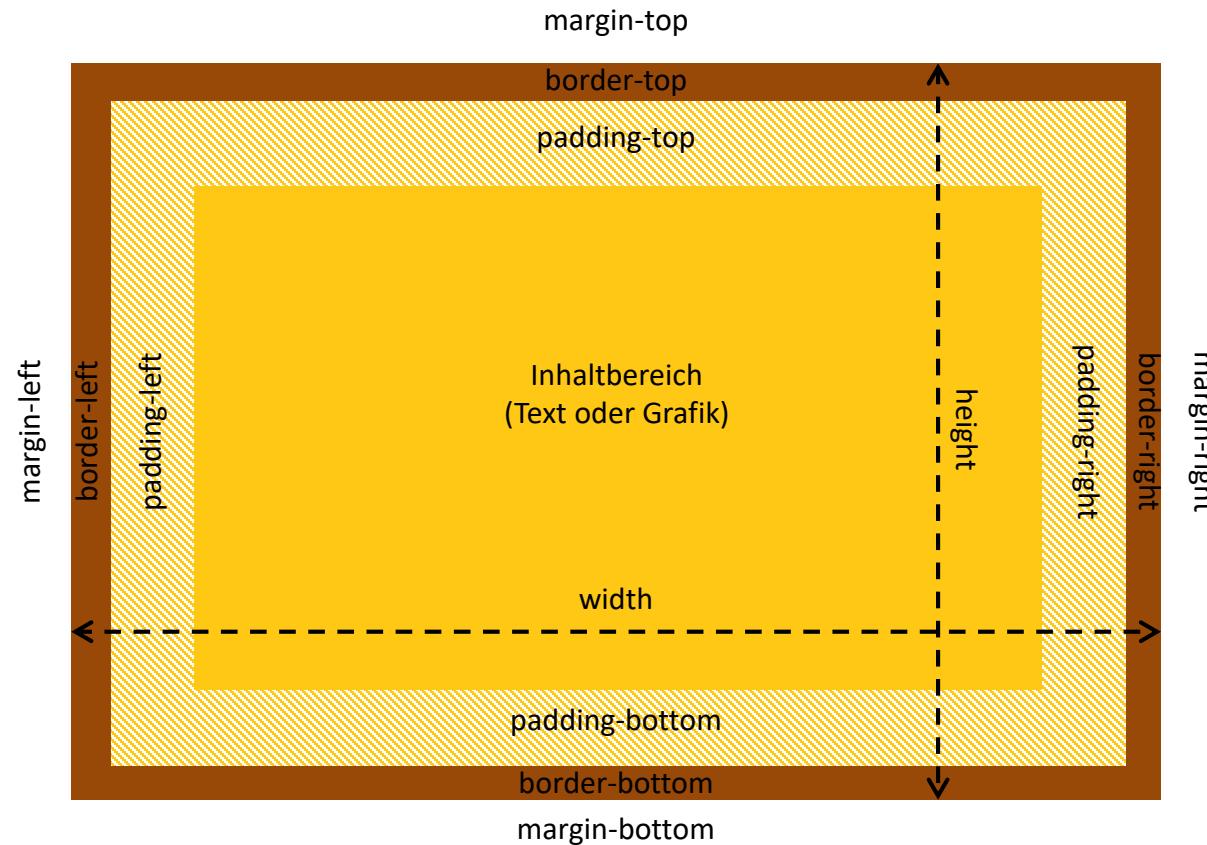


### Eigenschaften

- im CSS:  
**box-sizing: content-box;**
- Standard-Wert (default)**
- wird nicht vererbt

# BOX-SIZING

## CONTENT-BOX VS. BORDER-BOX



### Eigenschaften

- im CSS:  
**box-sizing: border-box;**
  - wird nicht vererbt
  - mehrere Boxen mit dieser CSS-Eigenschaften lassen sich in Höhe und Breite fehlerfrei mit **%-Angaben** bestimmen:
- 
- width: 80%; width: 20%;
- Der Internet Explorer hatte mit diesem Ansatz eine Vorreiterrolle und wollte dies ohne spezielle CSS-Anweisung als Standard setzen.

# BOX-SIZING: BORDER-BOX; GO FOR IT!

Fallback für aus vorgefertigten Elementen/Features mit content-box, die von externen Frameworks stammen

## Einfache Notation im CSS

\* = diese Notation bezieht sich auf alle HTML-Elemente – rechenintensiv!

```
* {  
    box-sizing: border-box;  
}
```



## Bessere, flexiblere Notation

```
body {  
    box-sizing: border-box;  
    /* box-sizing wird eigentlich nicht vererbt, aber : */  
}  
  
*, *:before, *:after {  
    box-sizing: inherit;  
}
```

Erkläre warum!

Die flexiblere Notation erlaubt es, für ein spezifisches Element xyz sowie dessen Nachfahren das Box-Model z.B. auf content-box umzustellen. Die Geschwister-Elemente von xyz sind davon nicht betroffen.

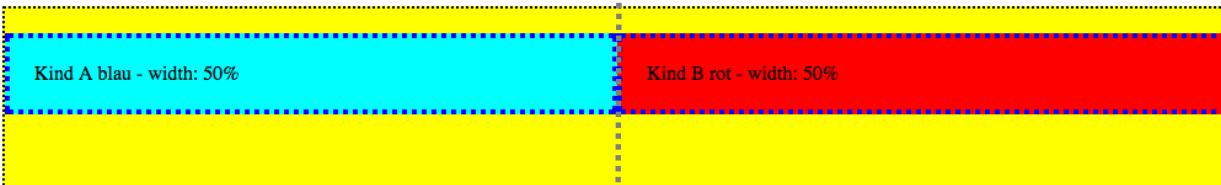
# ÜBUNG BOX-SIZING / VERERBUNG

Siehe auch Demo-Datei.

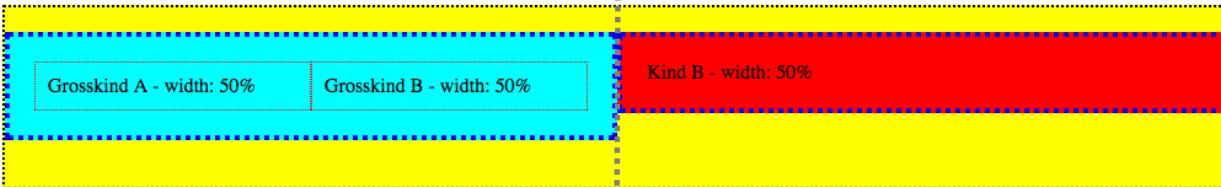
Welches Box-Modell  
wurde angewandt?

Beispiel 1: - die Mama hat Zwillinge mit Padding+Border bekommen:

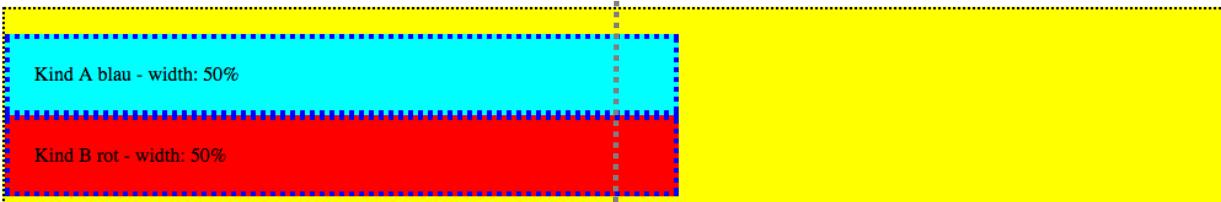
fifty-fifty...



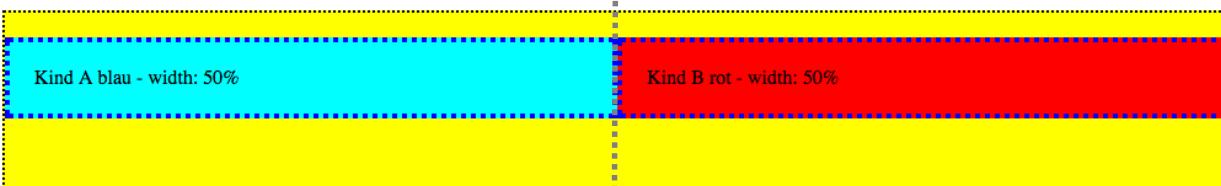
Beispiel 2: - die Grosskinder haben auch padding+border



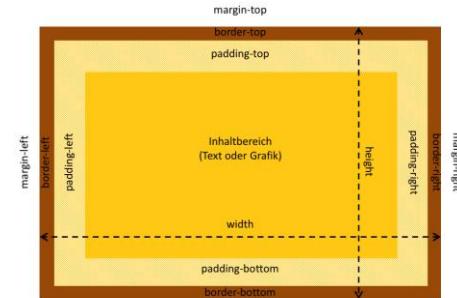
Beispiel 3: - wir heben mit der CSS-Klasse .contentBox das Border-Box-Modell auf.  
Box-sizing: content-box wird auch auf die Kinder vererbt.  
Wir sehen: mit width:50% haben die Zwillinge horizontal nebeneinander nicht mehr Platz.



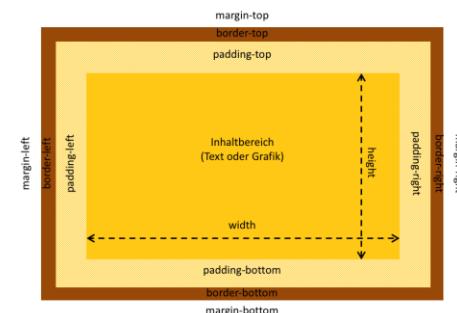
Beispiel 4: - ich funktioniere nach der "alten" Vererbung (border-box):



## Border-box (1)



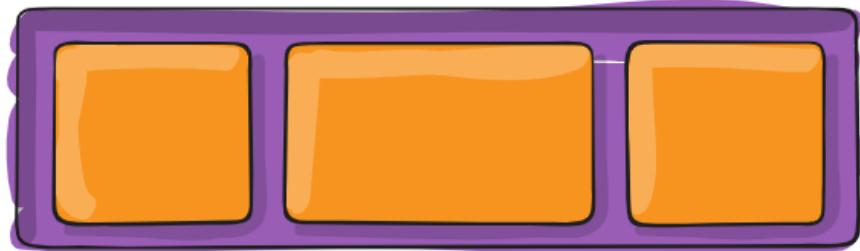
## Content-box (2)



# FLEX

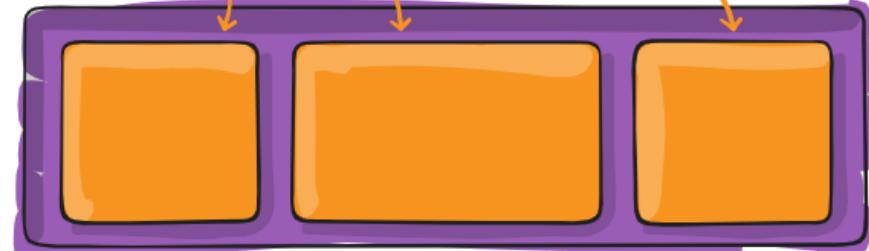
## Easy way of creating layouts

container ↗



**Properties for the Parent  
(flex container)**

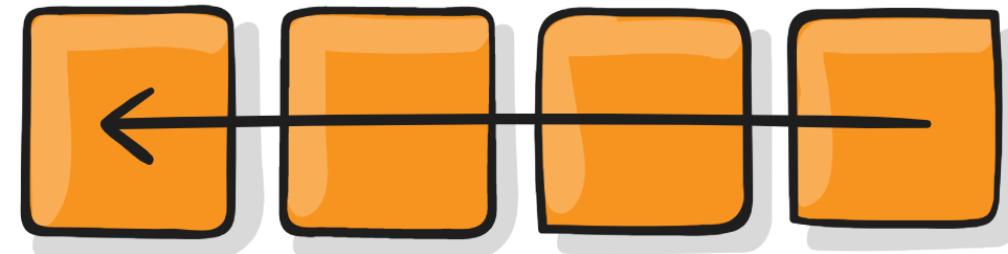
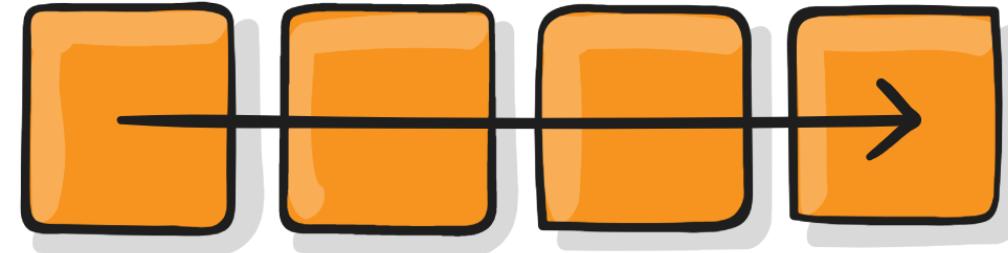
items ↗ ↗ ↗



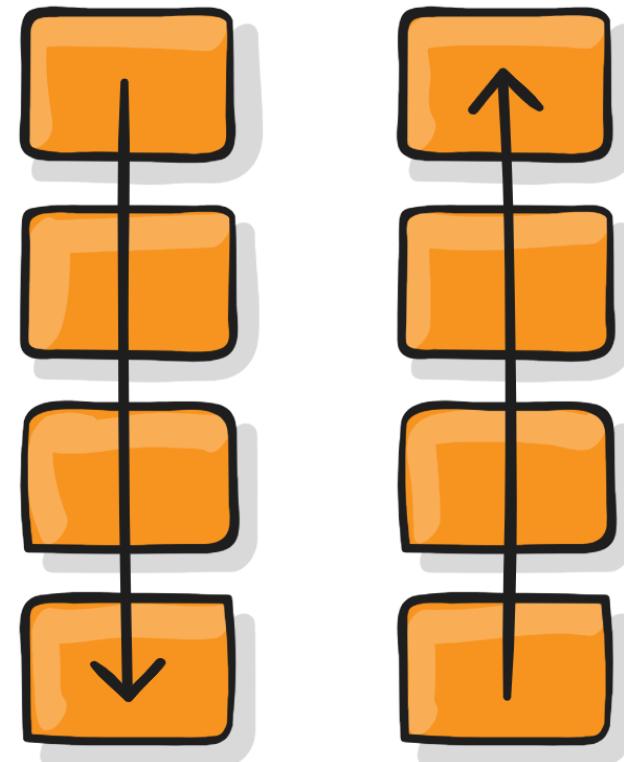
**Properties for the Children  
(flex items)**

# FLEX

## flex-direction



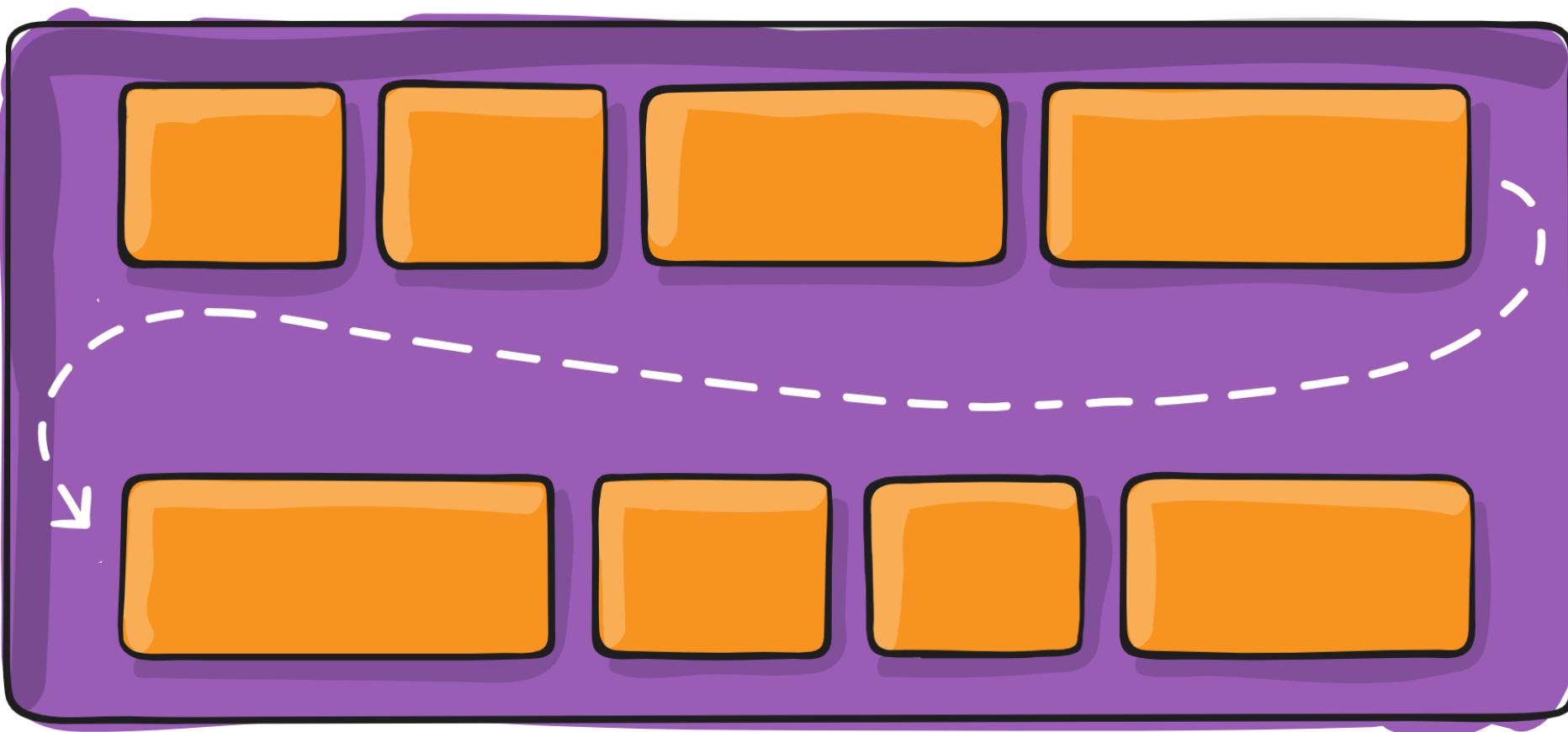
row (default)



column

# FLEX

## flex-wrap



# FLEX

## justify-content

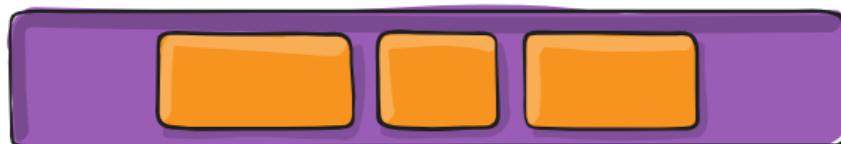
flex-start



flex-end



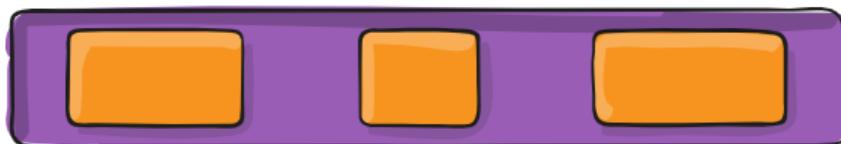
center



space-between



space-around



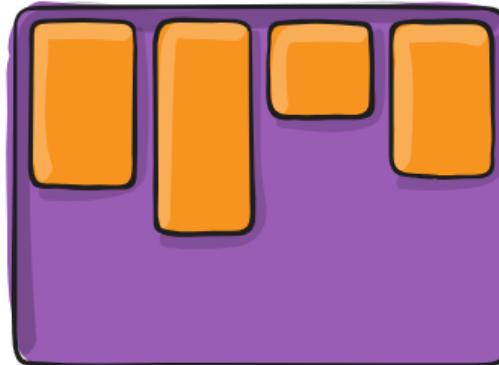
space-evenly



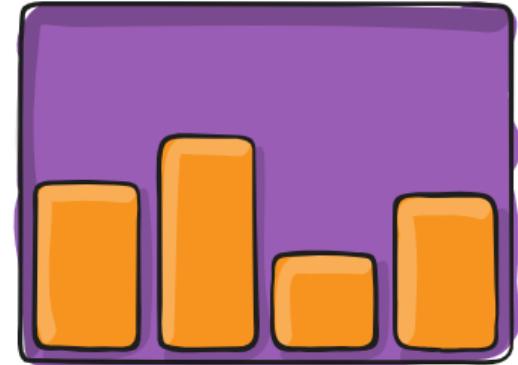
# FLEX

## align-items

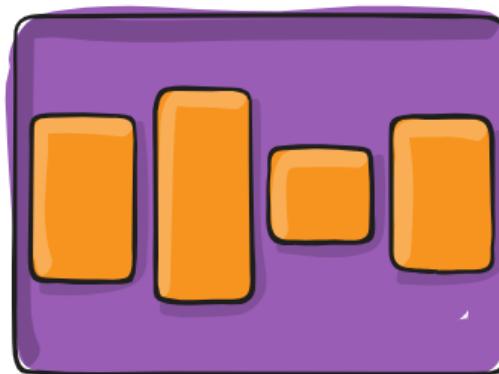
flex-start



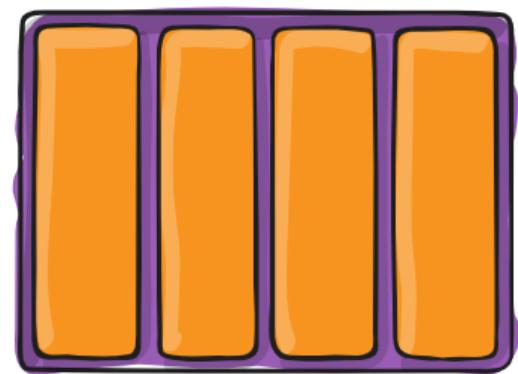
flex-end



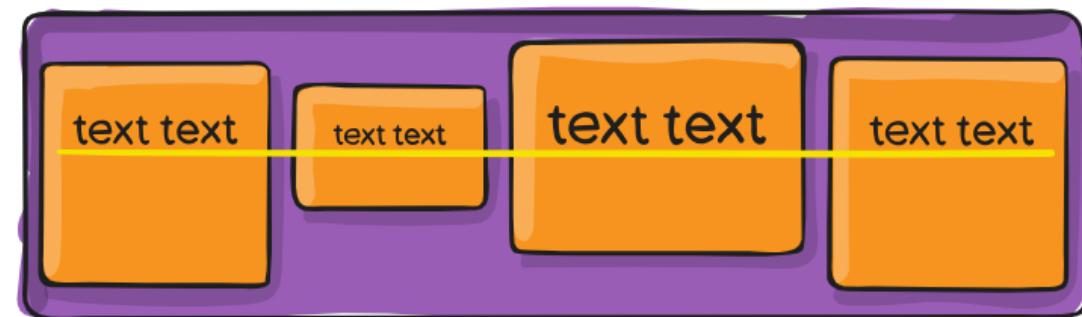
center



stretch



baseline



# EINMITTEN

# EINMITTEN VON ELEMENTEN

Wir können horizontal und/oder vertikal einmitten.  
Teste vor allem ab IE9!

- Es gibt moderne und altmodische Lösungen.  
Sei dir bewusst, dass das moderne CSS transform: translate (x-Achse, y-Achse) zur Zeit noch viele Internet Explorer-Anhänger ausschliesst!
- Wie teste ich für ältere Versionen des Internet Explorers?
  - Du hast einen PC:  
In neueren Internet-Explorer kannst du im "Web Inspector" ältere Versionen vom IE emulieren und somit testen.
  - Du hast einen Mac:  
ein Notbehelf zum Testen ist: <http://browsershots.org/>
- Siehe Ordner einmitten auf dem Server für Beispiele

# EINMITTEN VON ELEMENTEN

## Zusammenfassung der CSS-Einmittungstechniken

- Zentriert (aber auch nur horizontal oder vertikal möglich) ab IE 11:
  - das Kind-Element hat:  
**position: relative; top: 50%; left: 50% ;**  
**und Korrektur: transform: translate(-50%, -50%);**
- Horizontal – für alle Browser:
  - für INLINE Kind-Elemente:  
**text-align: center;** im Eltern-Element
  - für BLOCK Kind-Elemente:  
**width: \_\_px; margin: 0 auto;** im Kind-Element
- Vertikal – für alle Browser:
  - egal ob INLINE oder BLOCK Kind-Element:  
**display: table-cell; vertical-align: middle;** im Eltern-Element

# AGENDA

## Curriculum WDD

4100\_07

REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE /  
QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK  
CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT /  
MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT  
POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT  
BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / EINMITTEN / TABELLEN / IFRAME / FORMULAR  
VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST  
REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# NAVIGATION

## Der Hyperlink – der Wegbereiter des Web

```
<a href="http://sbb.ch" target="_blank" title="Externer Link zu SBB">  
  Auf Klick öffne ich die SBB-Seite in einem neuen Tag  
</a>
```



*Die Seite öffnet sich wie?*

target= \_self (default) | \_blank | \_parent | \_top\framename

im gleichen Tab

neuer Tab

im Parent-Frame

im Body des Fenster resp. Framename

*Wenn der Mauszeiger darüber stehen bleibt, dann erscheint ein Text:*  
title="Externer Link zu SBB"

# FARBVERLÄUFE

# FARBVERLÄUFE (GRADIENT) MIT CSS

Früher Mühsal, heute kinderleicht



background-image: linear-gradient(direction, color1 stop1, color2 stop2, etc...)

# KNACKNÜSSE

## Flaggen Tricolor mit CSS linear-gradient



versuche ein DIV so zu stylen  
unter Menüpunkt Beispiele

# FONTS

# SCHRIFTSATZ

## Charset utf-8 – für die meisten europäischen Sprachen

Zeichensatz-Angabe im Head:

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

Nachlese:

<https://wiki.selfhtml.org/wiki/Zeichenkodierung/Webserver>

Spezialzeichen

[http://www.w3schools.com/html/html\\_charset.asp](http://www.w3schools.com/html/html_charset.asp)

# FORMATE FÜR SCHRIFTEN

## Einbettungsmöglichkeiten

*CSS-Regel:*

```
font-family: 'Lato', 'Verdana', 'Arial', sans-serif;
```

Webfonts / eigene Schriften	Systemschriften beim Client
<ul style="list-style-type: none"><li>• mit ' _____ ' ansprechbar, also 'Lato'</li><li>• ab Hosting Service (z.B. Google Fonts)<ul style="list-style-type: none"><li>• einfach zum Implementieren: @import</li><li>• zusätzliche "online"-Abhängigkeit</li></ul></li><li>• ab eigenem Webspace:<ul style="list-style-type: none"><li>• aufwändiger einzubinden</li><li>• folgende Formate sind verbreitet:<ul style="list-style-type: none"><li>• CHECK caniuse.com</li></ul></li></ul></li></ul>	<ul style="list-style-type: none"><li>• Angabe der Schriftenfamilien</li><li>• Fallback (z.B. sans-serif)</li><li>• Einfach zum Implementieren</li><li>• Eventuell nicht authentisch mit dem Screendesign</li></ul>

# SCHRIFTGRÖSSEN PIXEL UND REM

## Umrechnen

Grösse in Pixel	<b>16 Pixel / 16 = 1rem (body-Tag)</b>
12px	0.75rem
14px	0.875rem
<b>16px</b>	<b>1rem</b>
18px	1.125rem
20px	1.25rem
24px	1.5rem
32px	2rem

# WEBFONTS VON GOOGLE

## ab dem eigenen Webspace

1. Lade die Schriftarten von Google Font runter
2. Konvertiere jene Schriftarten mit einem der folgenden Tools in verschiedene Formate (woff und woff2):
  - <https://www.fontsquirrel.com/>
  - <https://www.web-font-generator.com/>
  - <https://everythingfonts.com/>
3. Binde es ein mit @font-face  
<http://css-tricks.com/snippets/css/using-font-face/>
4. Mach ein Beispiel, indem du einen Text so formatierst.
5. Überprüfe mit dem Browser-Addon "What Font", welche Font effektiv angezeigt wird.

# TEXT FORMATTING

# CSS EIGENSCHAFTEN FÜR FONTS

## Die wichtigsten für Textformatierung

- font-family: Schrift1, Schrift2, sans-serif;
- font-size: px | rem | em | %
- color: farbname | hexacode | rgba
- text-decoration: none | underline | overline | line-through
- text-shadow: offset-x offset-y blur-radius color-code;  
(mehrere Schatten kommaseparierte Wertketten)
- text-align: left | right | center | justify
- vertical-align: top | bottom | middle
- text-transform: uppercase | lowercase | capitalize
- text-indent: : px | rem | em | %
- letter-spacing: : px | rem | em (auch negative Werte)
- line-height: 0 bis ~ (Fliesskommazahl, px, %)
- word-spacing: px | rem | em | % (auch negative Werte)

# KNACKNÜSSE

Kreis mit Rahmen – mit box-shadow



# KNACKNÜSSE

## CSS Schrifteffekte

Grell

CSS3 für Webdesigner

Abkantung (engl.: Emboss)

CSS3 für Webdesigner

Relief (engl. Bevel)

CSS3 für Webdesigner

# ICON FONTS

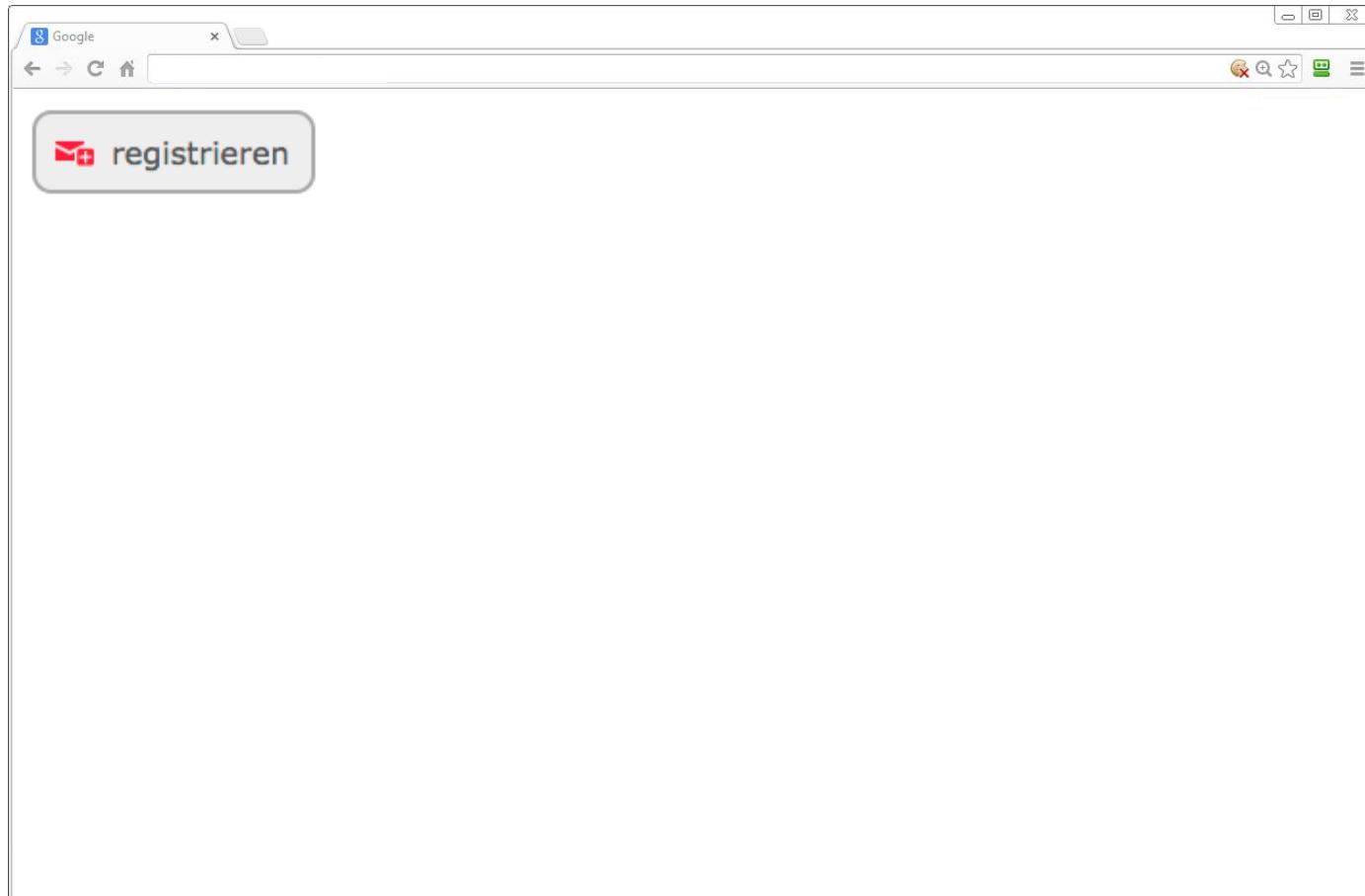
# ICONS ALS SCHRIFTSATZ mit CSS formatierbar!

Anbieter	Beschreibung
icofont.com	Simple icons
fontawesome.io	Complex icon collection

→ Oftmals muss vertikaler Versatz angepasst werden.  
Vergewissere Dich auch beim Zoomen im Browser!

# BEISPIEL MIT GENERICON

Versuche nachzubauen und zoome im Browser.



# AGENDA

## Curriculum WDD

4100\_07

REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE /  
QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK  
CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT /  
MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT  
POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT  
BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / EINMITTEN / TABELLEN / IFRAME / FORMULAR  
VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

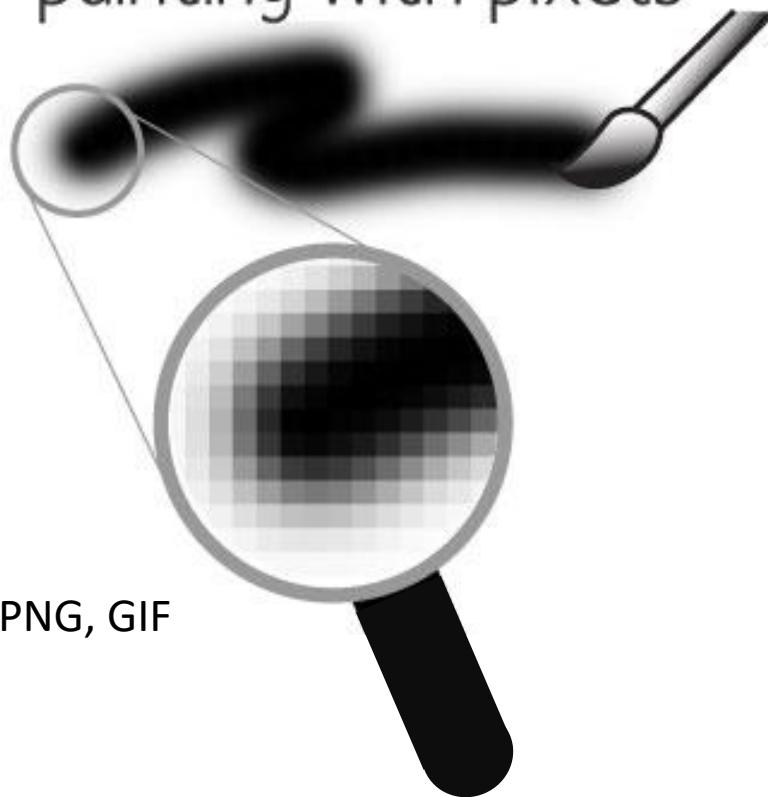
HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST  
REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# BILDER

# FRONTEND

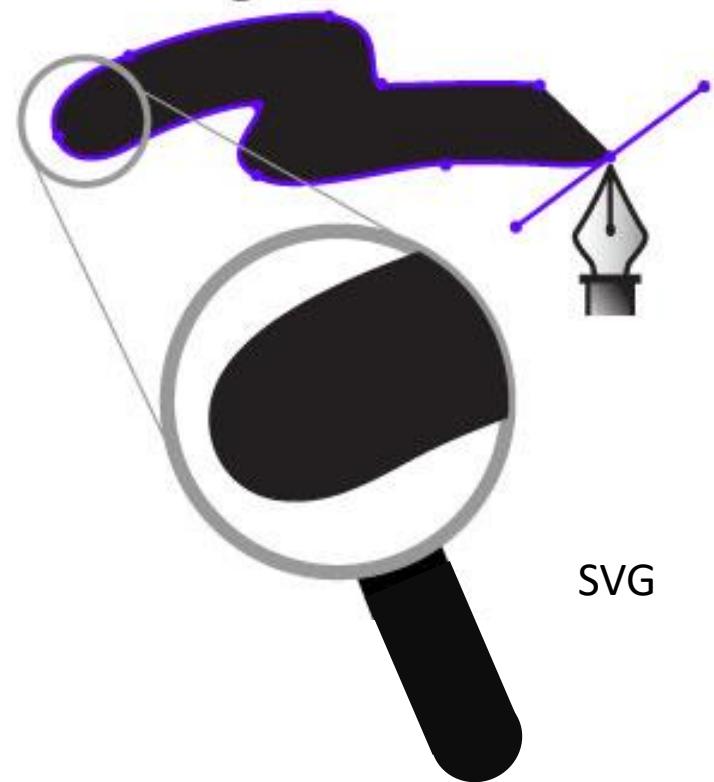
## Pixel vs. Vektor

painting with pixels



JPG, PNG, GIF

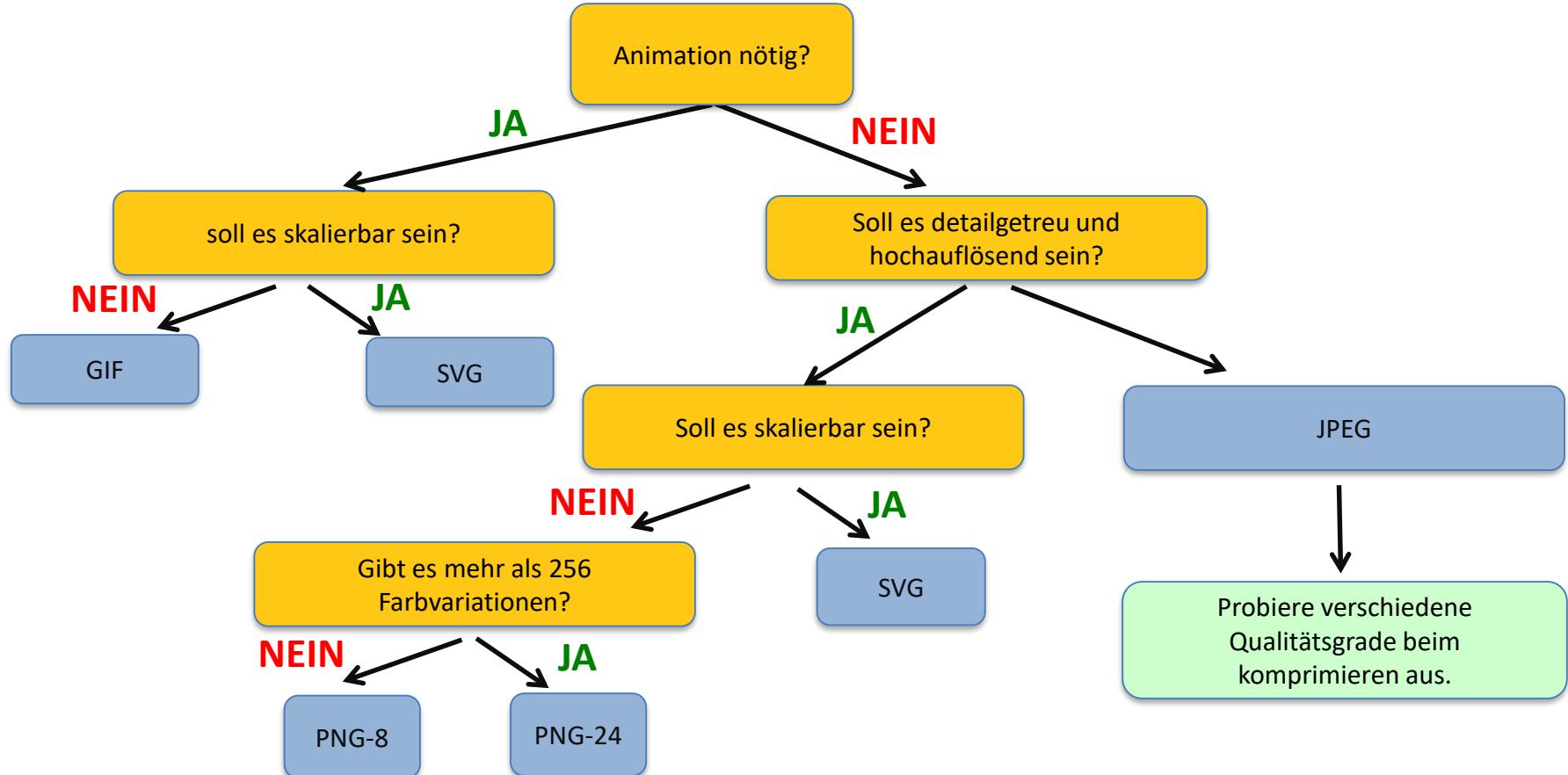
drawing with vectors



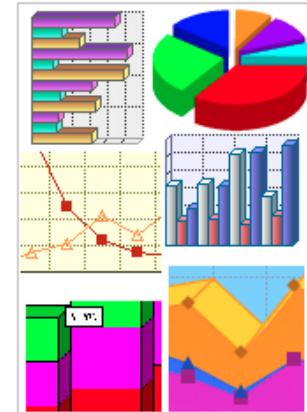
SVG

# WELCHES BILDFORMAT?

Wir unterscheiden grob zwischen Pixel- und Vektorbilder



# Bildformate



wird durch eine  
Animation  
vergrössert  
skaliert

# Bildformate

Kürzel	Name	Vergleich zur Originalgrösse	Kompressionstyp	RGB	CMYK	Indexiert	Verwendungszweck	Eigentümer	Transparenz Unterstützung	Charakter
jpg	Joint Picture Expert Group	Es wird immer komprimiert, aber variable je nach Komprimierungstiefe	JPG, verlustbehaftet	Ja, 16,7 Mio. Farben	Ja, 16,7 Mio. Farben	nein	Bei photorealistischen Bilder im Internet, Druck	Frei	nein	Verlustbehaftet, spart Speicherplatz, 24-bit Farbtiefe, interlacing möglich, zunehmende Verschlechterung bei mehrmaligem Abspeichern mit Komprimierung, sehr verbreitet
gif	Graphics Interchange Format	Immer zu etwa 3:1	LZW (Lempel-Ziv-Welch) (lizenziert)	nein	nein	8-bit, 256 Farben max.	Animationen mit oder ohne Tweening für Internet, für Bilder mit geringer Farbtiefe wie Logos, Buttons, etc..	Compuserve, nicht lizenziert	Möglich, aber kein Alphakanal da keine Transparentabstufung möglich (also entweder oder Entscheidung)	Verlustfrei, sehr verbreitet
png	Portable Network Graphics	Weniger komprimierbar als JPG, jedoch verlustfrei		24-bit Farbtiefe (mit Alphakanal: 32-bit) 16,7 Mio. Farben	24-bit Farbtiefe (mit Alphakanal: 32-bit) 16,7 Mio. Farben	nein	Internet	Frei	Alpha-Kanal	Plattformunabhängig, verlustfreies komprimieren, Gamma Korrektur, Interlacing möglich
svg	Scalable Vector Graphics	Verlustfrei, 20%-30% der Originalgrosse	SVGZ				Vektorgraphiken, Logos, Fonts, ist skalierbar. Unterstützt Alphakanal, Clipping paths, Filter Effekten und Transformationen	Frei	Alpha-Kanal	Ist als Text in ein XML Datei gespeichert. Ist skalierbar

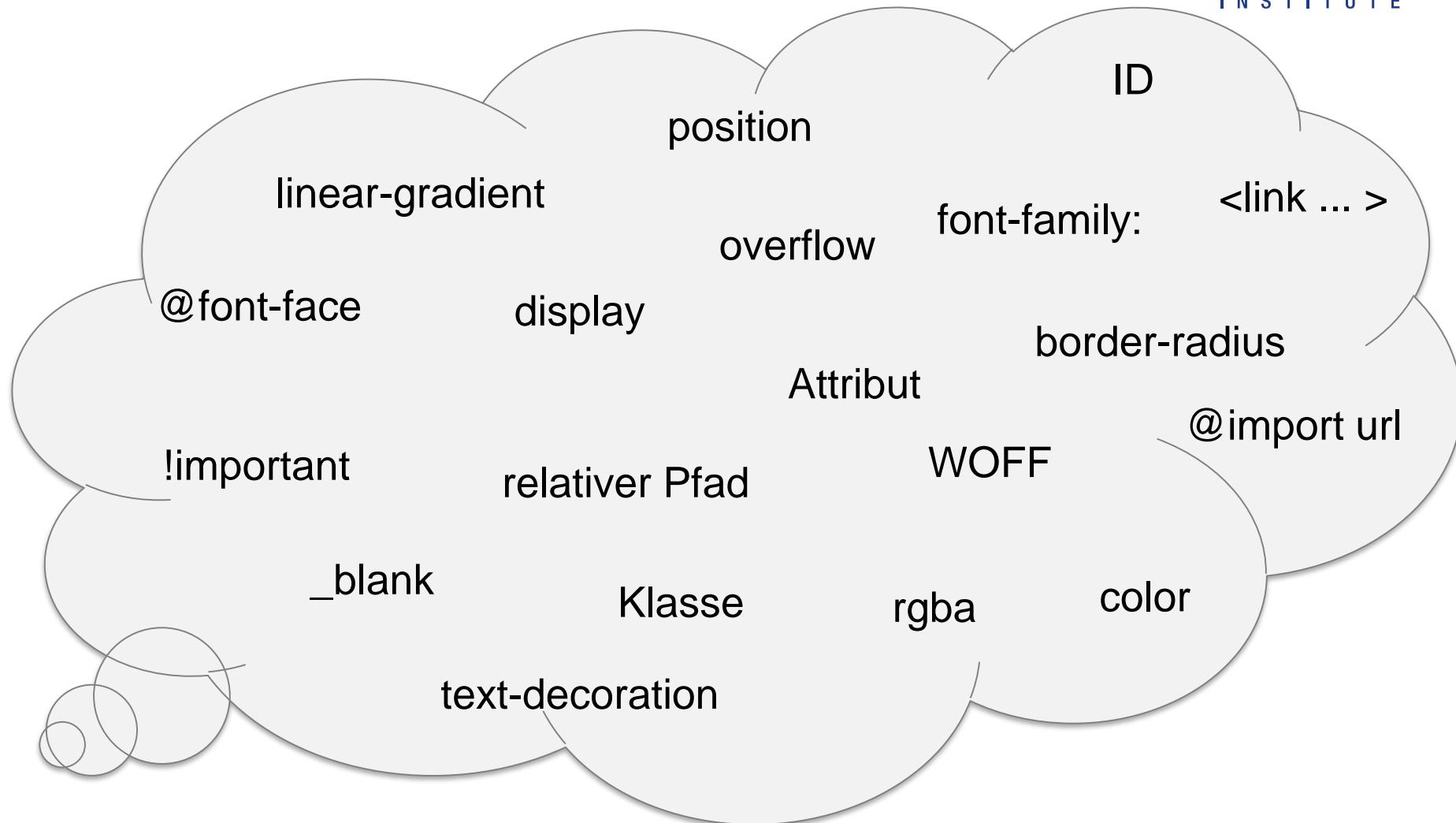
# KNACKNÜSSE

## Spielereien mit CSS box-shadow

Bilderstapel



# REPETITION



# HINTERGRUND

# FENSTER AUSFÜLLEND MIT CSS

Früher mühsam, heute kinderleicht

```
body {  
    background-color: #999;  
    min-height: 100%;  
    background-image: url('extragross.jpg');  
    background-position: left top;  
    background-repeat: no-repeat;  
    background-size: cover;  
}
```

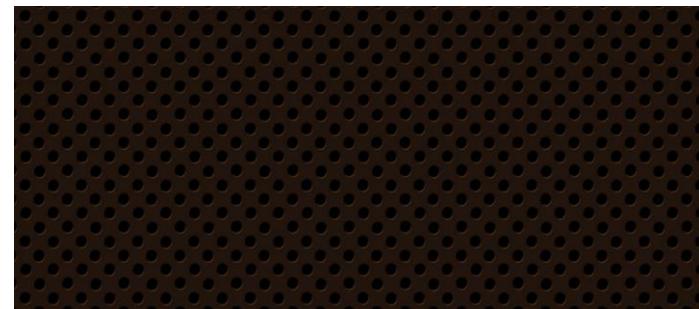
# BILDER – MIT HTML ODER CSS?

## HTML <img> vs. CSS background-image: url()

	Vorteile	Nachteile
CSS background-image: url();	<ul style="list-style-type: none"> <li>mit background-size:cover =&gt; Bild ist bildschirmfüllend</li> <li>man kann auf einfache Weise noch Text drauf platzieren</li> <li>Werden nur geladen, wenn benötigt (:hover)</li> <li>Gilt als Vorlage für alle Webpages</li> <li>Eher dekorativen Charakter</li> <li>CSS Sprites – viele Icons auf 1 Bild =&gt; 1 Request</li> </ul>	<ul style="list-style-type: none"> <li>Bilder müssen noch bei :hover nachgeladen werden =&gt; verzögerte Anzeige.</li> <li>schwerer editierbar für Kunden, wenn Bildpfad und Bildname nicht über ein CMS editierbar gemacht werden.</li> <li>werden nicht gedruckt</li> </ul>
HTML <img src="">	<ul style="list-style-type: none"> <li>Passt thematisch besser zum Inhalt</li> <li>Bilder sind für Kunden besser zugänglich.</li> <li>Leichter animierbar und mit Effekten ausstattbar</li> <li>mit rechter Maustaste direkt abspeicherbar</li> <li>können von Suchmaschinen indexiert werden</li> <li>können alternativen Text mit Attribut alt für den Screenreader zur Verfügung stellen</li> <li>Mitskalierbar mit Schriftgrösse im Browser</li> </ul>	<ul style="list-style-type: none"> <li>sind leichter "klaubar"</li> <li>für jedes Bild werden einzelne Requests getätigt – das summiert sich</li> <li>die Bilder müssen zum Content passen</li> </ul>

# KNACKNÜSSE

Verkachelung mit Hintergrundbild (Tessellation)

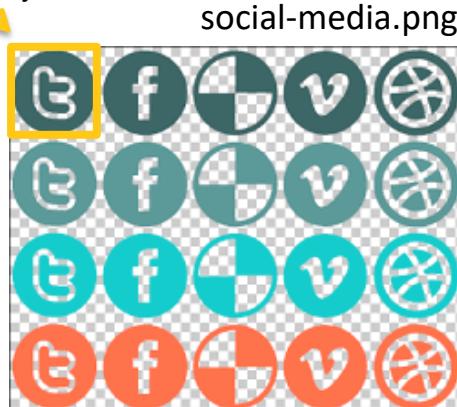


# CSS SPRITES

# CSS SPRITES

viele ähnliche Icons auf einem Bild – siehe Ordner

```
#meinDiv {
    background: url(social-media.png) no-repeat;
    width: 40px;
    height: 40px;
}
```



## Eigenschaften

- nur 1 Request für verschiedene Bilder
- als background-image im CSS definierbar
- Ausschnittfenster mit width und height zu definieren

```
#meinDiv { background-position: 0px 0px; }
```

```
#meinDiv: hover { background-position: 0px -40px; }
```

```
#meinDiv: visited { background-position: 0px -80px; }
```

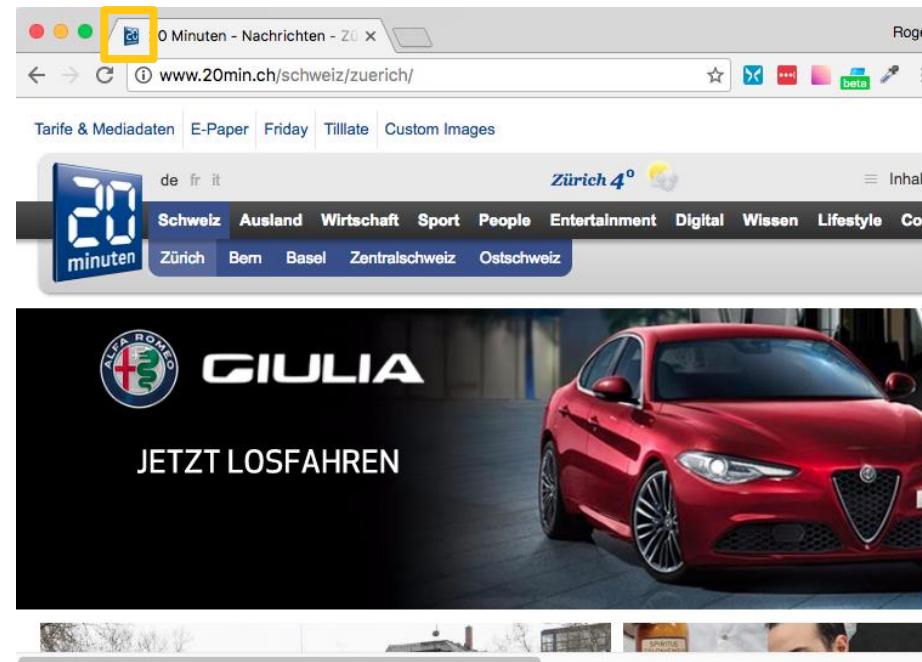
```
#meinDiv: active { background-position: 0px -120px; }
```

Weiteres Beispiel live: [http://www.myswitzerland.com//img/sprites/boxed\\_icons.png](http://www.myswitzerland.com//img/sprites/boxed_icons.png)

# FAVICON

# FAVICON

Ein PNG-Bild oder .ico-Bild. von 32x32 Pixel



- Wird über den `<link ...>`-Tag im Head-Bereich eingebaut:  
[http://www.w3schools.com/tags/att\\_link\\_sizes.asp](http://www.w3schools.com/tags/att_link_sizes.asp)
- Es gibt auch zahlreiche Favicon-Generatoren im Web,  
z.B. <http://www.favicon.cc/>

# HAUSAUFGABEN

# AGENDA

## Curriculum WDD

4100\_07

ENTWICKLUNGSUMGEBUNG / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / QUELLCODE, DOM und WEB INSPECTOR / SEMANTIK

CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT

POSITIONIERUNG / BOX-MODEL / FLEX

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT

BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

TABELLEN / IFRAME / FORMULAR

VIDEO/AUDIO / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST

REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# TABELLEN

# TABELLEN SPARSAM BENUTZEN

## Nur für tabellarische Daten

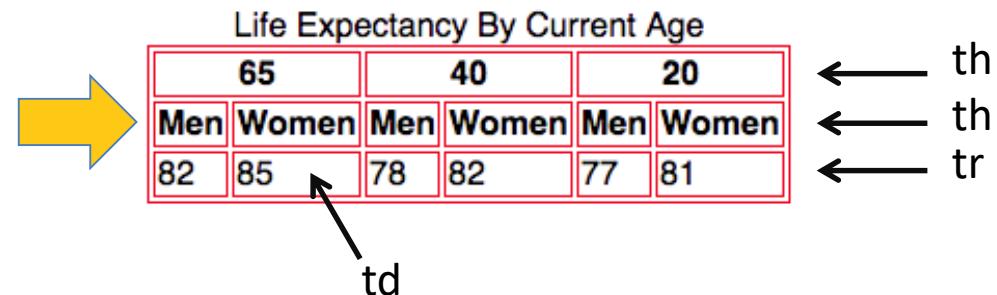
- Früher wurden Webpages mit riesigen Tabellen gemacht.  
Der blanke Graus für Screenreader respektive Sehgeschwäche.
- Screenreader lesen Reihe für Reihe (<tr>)
- Kleine Tabellen für Daten-Reihen und HTML-Emails sind okay – siehe auch im Ordner:

```

<style>
  table, tr, th, td {
    border: 1px solid red;
  }
  table {
    /* Damit verschmelzen die Ränder: */
    /* border-collapse: collapse; */
  }
</style>
<table>
  <caption>Life Expectancy By Current Age</caption>
  <tr>
    <th colspan="2">65</th>
    <th colspan="2">40</th>
    <th colspan="2">20</th>
  </tr>
  <tr>
    <th>Men</th>
    <th>Women</th>
    <th>Men</th>
    <th>Women</th>
    <th>Men</th>
    <th>Women</th>
  </tr>
  <tr>
    <td>82</td>
    <td>85</td>
    <td>78</td>
    <td>82</td>
    <td>77</td>
    <td>81</td>
  </tr>
</table>

```

Life Expectancy By Current Age



65		40		20	
Men	Women	Men	Women	Men	Women
82	85	78	82	77	81

td

**Legende:**

table = Tabelle  
 caption = Überschrift  
 tr = table row  
 th = table head  
 td = table data

# TABELLEN SPARSAM BENUTZEN

## Nur für tabellarische Daten

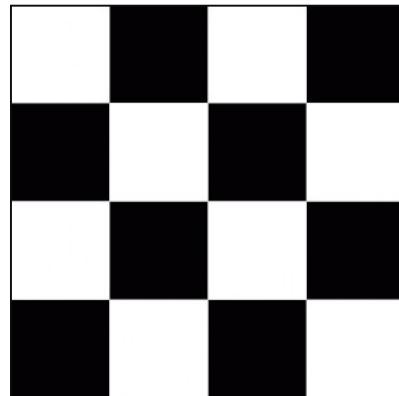
- Baue nach mit Online-Hilfen.
  - HTML: table, th, tr, td, colspan, rowspan, caption
  - CSS: border-collapse, odd/even

Number	First Name
1	Eve
2	John
3	Adam
4	Jill

Time Table					
Hours	Mon	Tue	Wed	Thu	Fri
	Science	Maths	Science	Maths	Arts
	Social	History	English	Social	Sports
	Lunch				
	Science	Maths	Science	Maths	Project
	Social	History	English	Social	

# KNACKNÜSSE

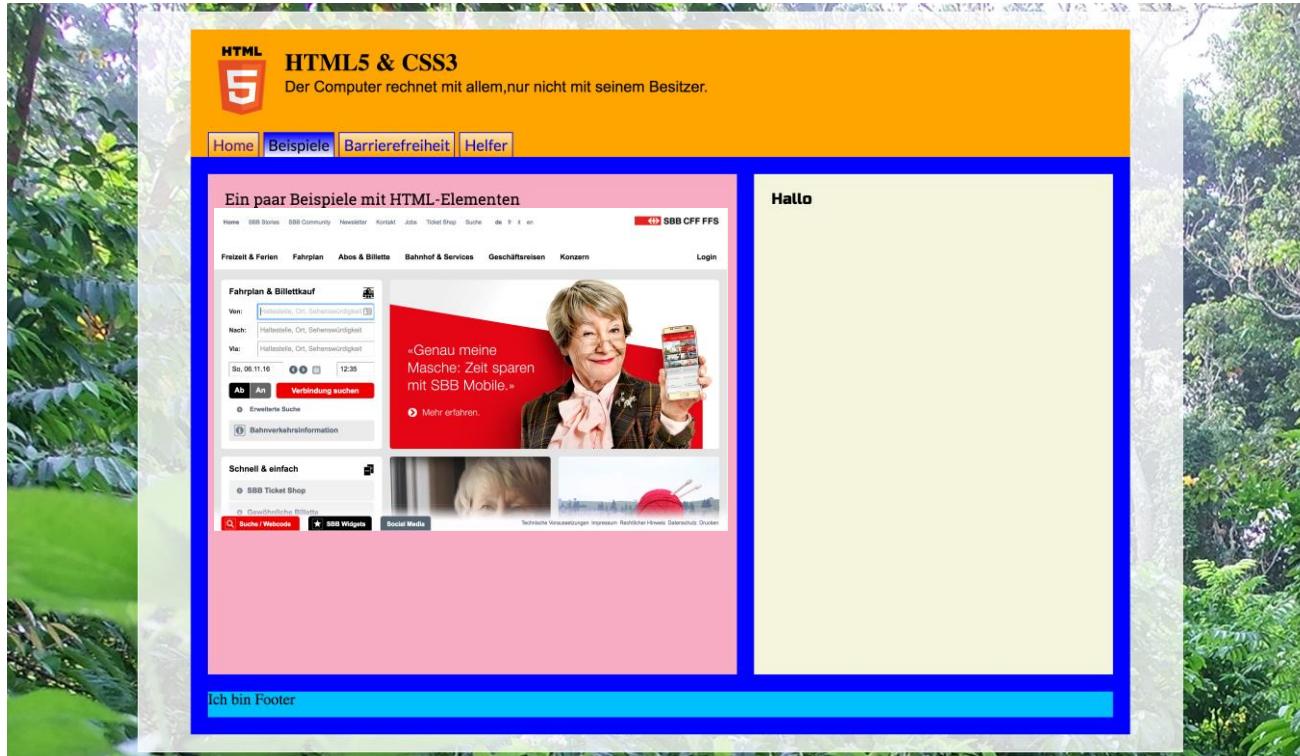
Schachbrett – mit <table> und CSS-Selektor odd/even



# IFRAME

# IFRAME

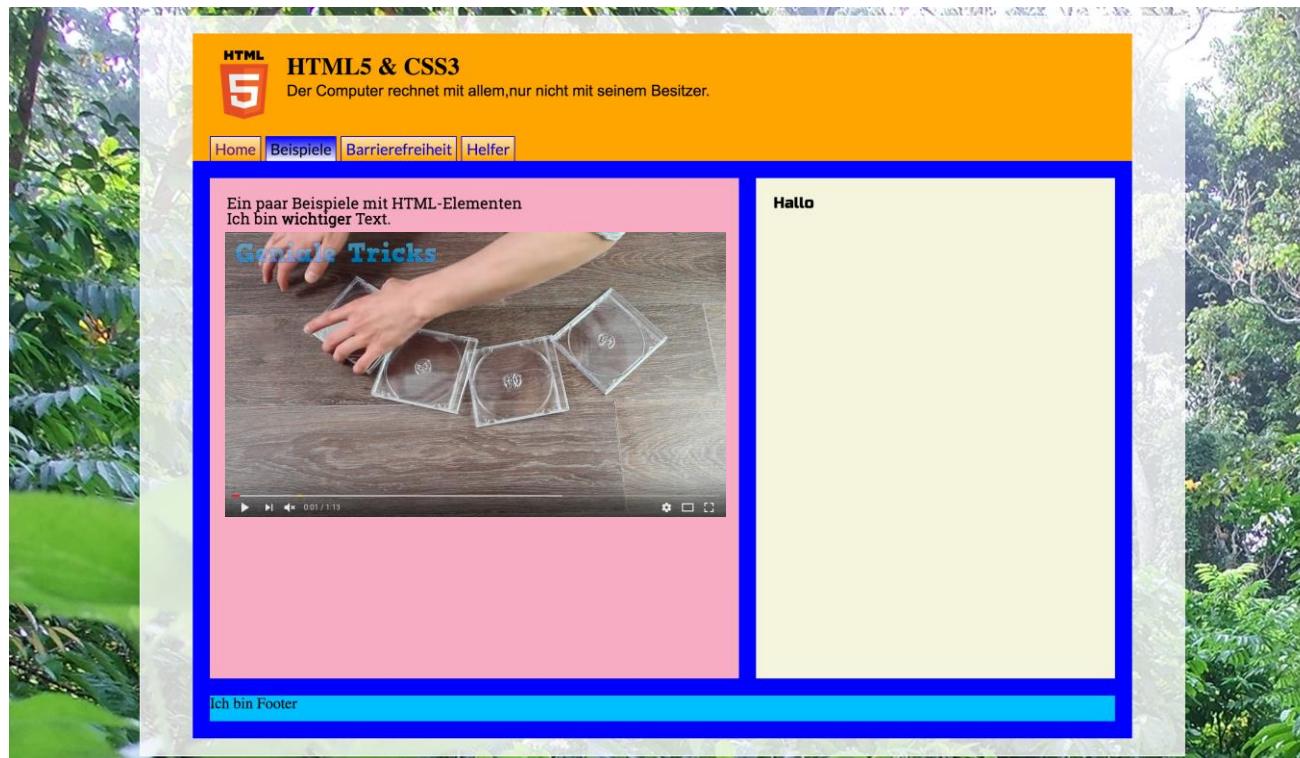
## eine fremde Website in die Webpage einspeisen



# IFRAME

Viele Drittdienste werden so eingebunden.

- Merke: Google wird den Inhalt vom iFrame NICHT indexieren!
- Versuche, ein Youtube-Video in deine Website einzubinden – mit flexiblen Größen.



# FORMULAR

# FORMULARFELDER

## Auch mit CSS können wir die schön machen

- Alle Form-Elemente sollten ein <label ...>-Tag haben – für Screenreader und erweiterte Klickfläche, die in den Cursor in das Eingabefeld platziert.
- Alle Form-Elemente sind innerhalb von <form> hier sind input-Felder </form>
- Diese Felder lassen sich typisieren:

email, tel, url, radio, checkbox, date, time, color, submit, reset

→ damit die richtige Tastatur auf den mobilen Geräten eingeblendet wird.

### To do:

- Baue das besprochene Formular in deine Website ein und style es etwas.
- Mit CSS cursor:pointer blendest du ein Mauszeiger ein.
- Baue noch :focus ein für Felder, um eine spezielle Hintergrundfarbe im Feld anzuzeigen, wenn man reinklickt.
- Versuche mit dem Tabulator durchzutabben, und verbessere allenfalls die Tab-Führung durch tabindex="number".

# WICHTIGE FELDER

Schreibe seitlich vom Formular,  
um welche Art von HTML-Tags es sich handelt.

**HTML/CSS FORM**

**Vorname:** Dein Vorname

**Nachname:** Dein Nachname

**E-Mail:** Deine E-Mail

**Passwort:** Dein Passwort

**Telefon:** Deine Tel-Nummer

**Website:** Deine Website

**Männlich**  **Weiblich**

Besitze ein Velo  Besitze ein Auto

Wie viele Unfälle  
hatten sie schon?

Welche Automarke  
finden Sie toll?

Ihr Geburtsdatum:

Ihre Lieblingsfarbe:

Wähle eine Zeit:

Wähle deinen  
Browser:

**Dein Kommentar:** Dein Kommentar

# WICHTIGE FELDER

Schreibe seitlich vom Formular, um welches HTML-Tag und den Typ dazu es sich handelt.

<label for="vorname">Vorname</label>  
etc...

HTML/CSS FORM	
Vorname:	<input type="text"/> <input type="text"
Nachname:	<input type="text"/> <input type="text"
E-Mail:	<input type="email"/> <input type="email"
Passwort:	<input type="password"/> <input type="password"
Telefon:	<input type="tel"/> <input type="tel"
Website:	<input type="url"/> <input type="url"
Männlich <input checked="" type="radio"/> Weiblich <input type="radio"/>	<input type="radio" name="identisch"
Besitze ein Velo <input type="checkbox"/> Besitze ein Auto <input type="checkbox"/>	<input type="checkbox"
Wie viele Unfälle hatten sie schon? <input type="number"/>	<input type="number" min="1" max="5"
Welche Automarke finden Sie toll? <input type="selection"/>	<selection> mit einigen <option>
Ihr Geburtsdatum: <input type="date"/>	<input type="date"
Ihre Lieblingsfarbe: <input type="color"/>	<input type="color"
Wähle eine Zeit: <input type="time"/>	<input type="time"
Wähle deinen Browser: <input type="list"/>	<input list="browsers" und <datalist id="browsers" für Type ahead Funktion
Dein Kommentar: <input type="text"/>	<textarea>
<input type="reset"/> Zurücksetzen	<input type="reset" und <input type="submit" 120
<input type="submit"/> Absenden	

# AGENDA

## Curriculum WDD

4100\_07

ENTWICKLUNGSUMGEBUNG / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / QUELLCODE, DOM und WEB INSPECTOR / SEMANTIK

CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT

POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT

BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / TABELLEN / IFRAME / FORMULAR

VIDEO/AUDIO / BARRIEREFREIHEIT / WAS MOECHTE ICH REPETIERT HABEN? / IM SELBSTSTUDIUM

4100\_11

HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN /  
SUBNAVIGATION / ZWISCHENTEST

REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDESIGN ÜBERSETZEN /  
WISSENSQUIZ

# VIDEO/AUDIO

# AUDIO PLAYER

## Die nativen Audio Player der gängisten Browser



Opera (Win)



Chrome (OS X/Win)



Chrome (Android 4.\*)



Safari (OS X)



Safari (iOS)



Firefox (Mac)



IE9+



# AUDIO DATEIEN

Verbreitete Formate – browserabhängig.

Browser	MP3	OGG (Vorbis)	MP4	WebM
Firefox	✓	✓	✓	✓
Chrome	✓	✓	✓	✓
Safari	✓	✗	✓	✗
IE9 und neuer	✓	✗	✓	✗✓

[https://en.wikipedia.org/wiki/HTML5\\_Audio](https://en.wikipedia.org/wiki/HTML5_Audio)

[http://www.w3schools.com/tags/ref\\_av\\_dom.asp](http://www.w3schools.com/tags/ref_av_dom.asp)

# AUDIO DATEIEN EINBINDEN

Anstelle des Plug-Ins Flash Players (.swf) können Browser nun einen eigenen Player zur Verfügung stellen.

```
<audio id="song" controls preload="metadata">
    <source src="elvis.mp3" type='audio/mpeg; codecs="mp3"'>
    <source src="elvis.oga" type='audio/ogg; codecs="vorbis"'>
    usw.
    <!-- add your fallback solution here -->
    Dein Browser unterstützt keines unserer Formate. Lade es runter:
    <a href="elvis.mp3">MP3-Datei runterladen </a>
</audio>
```

- Man kann auch ein eigenes Steuerungspanel basteln mit HTML-Elementen und JavaScript. Oder einen vorgefertigten Player aus dem Web.
- Siehe auch:
  - <http://html5doctor.com/html5-audio-the-state-of-play/>
  - [http://www.w3schools.com/tags/ref\\_av\\_dom.asp](http://www.w3schools.com/tags/ref_av_dom.asp)

# VIDEO FORMATE

Verbreitete Formate – browserabhängig.



Browser	MP4 (H.264)	WebM (VP8)	OGG (Theora)	WebM (VP9)
Firefox	✓	✓	✓	✓
Chrome	✓	✓	✓	✓
Safari	✓	✗	✗	✗
IE9 und neuer	✓	✗	✗	✗

[https://en.wikipedia.org/wiki/HTML5\\_video](https://en.wikipedia.org/wiki/HTML5_video)

[http://www.w3schools.com/tags/ref\\_av\\_dom.asp](http://www.w3schools.com/tags/ref_av_dom.asp)

# VIDEO DATEI EINBINDEN

## Einbetten von Audiofiles

```
<video width="320" height="240" controls autoplay>
  <source src="mein_video.mp4" type="video/mp4">
  <source src="mein_video.ogg" type="video/ogg">
  usw...
  <!-- Add your fallback solution here -->
  Dein Browser unterstützt keines der Videoformate.
  Lade die Datei runter:
  <a href="mein_video.mp4">MP4-Datei runterladen </a>
</video>
```

- Bei der Video-Datei unterscheiden wir zwischen Container (Dateiformat) und Codec, die Komprimierungsart.
- Alternative ist das Videohosting (einbinden über iFrame):
  - youtube (Nachteil: Werbung)
  - vimeo
- Man kann auch ein eigenes Steuerungspanel basteln mit HTML-Elementen und JavaScript.

Siehe auch:

[http://www.w3schools.com/tags/ref\\_av\\_dom.asp](http://www.w3schools.com/tags/ref_av_dom.asp)

# BARRIEREFREIHEIT

# BRAIN STORMING

## WIR VERSETZEN UNS IN LEUTE ...

- mit Sinneseinschränkungen
- motorischen Einschränkungen

In welchen Bereichen treffen wir Massnahmen zur Zugänglichkeit an?

Lichtsignal mit Piepsen / Rampen / Gehwegmarkierungen / Geländer / Ansagen in Zug / Lautstärke / Analoge und Digitale Kommunikationswege aufrechterhalten / Übersichtlich und intuitive Anordnung von Inhalten / Grosse, kontrastreiche Schriften / Eindeutige, international leicht verständliche Piktogramme / Braille / Gebärdensprache / kurzer Mausweg / räumliche Zugänglichkeit / Sprachen / Sitzplätze / Support / Ausschilderungen / Farbe der Busstange / Wecker mit Licht / Televison mit Untertiteln

# BARRIEREFREIHEIT

## Definition

Leichte, einfache Zugänglichkeit für alle.  
Rücksichtnahme auf Leute mit Einschränkungen.

Im Web insbesondere:

- Sehgeschwächte und/oder Blinde
- Schwerhörige und/oder Taube
- Beweglichkeitseingeschränkte



Bundes- und bundesnahe Websites müssen per Gesetz barrierefrei sein

Zur Nachlese: Es gibt von W3C Empfehlungen:

- <https://www.w3.org/TR/wai-aria/>
- <https://www.w3.org/WAI/intro/aria>
- <https://www.w3.org/Translations/WCAG20-de/>

Zum Prüfen:

- <http://www.cynthiasays.com/>
- CSS ausschalten

# HILFSMITTEL

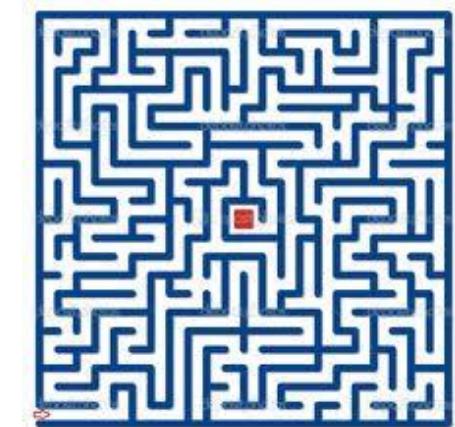
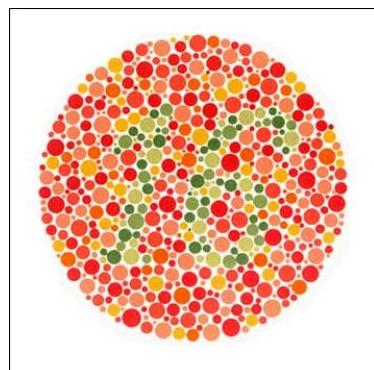
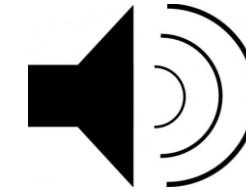
## Gerätetechnik



# BARRIEREFREIHEIT

## Die Einschränkungen sind vielschichtig im Web

Verwirrende Navigation, Bilder, Schriftgrößen, Farbkontraste,  
Nachlademöglichkeit von Inhalt ohne Page Reload



# BARRIEREFREIHEIT

## An dies musst du denken

- Durchtabben logisch? Hilf dir sonst mit Attribut tab-index="Nummer"
- Alt- und title-Attribut für Bilder
- Title-Attribut für Links
- Untertitel für Videos nötig?
- Transkription für Videos nötig?
- Div mit Skip Links zu Navi und Content, welches offscreen ist:

```
<div id="access-links">
  <a href="#skip-to-content">zum Inhalt</a>
  <a href="#primary-nav">zur Hauptnavigation</a>
</div>
```
- und ➔

# BARRIEREFREIHEIT

## <img>-Tag mit alt- und title-Attribut

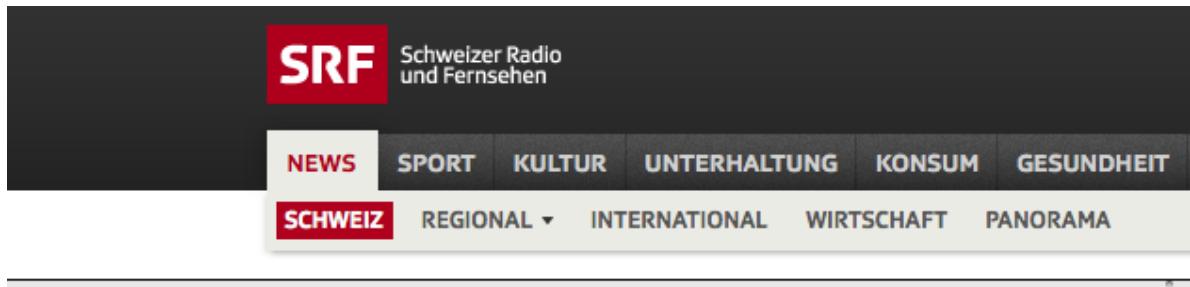


```
<span>Heute </span>

<span>6°/14°C </span>
```

# BARRIEREFREIHEIT

## Aktiver Navigationspunkt mit zusätzlichem Screenreader-Text



The screenshot shows the SRF website's header and a portion of its page content. The header includes the SRF logo and navigation links for NEWS, SPORT, KULTUR, UNTERHALTUNG, KONSUM, and GESUNDHEIT. Below the main navigation, there are regional links for SCHWEIZ (REGIONAL, INTERNATIONAL, WIRTSCHAFT, PANORAMA). A developer tools window is overlaid, showing the HTML structure of the active navigation item. The active item is an [News](/news) link under the SCHWEIZ category. This link contains a , aktiver Navigationspunkt element, which is highlighted in blue. To the right of the code, a CSS style block defines the .offscreen class with properties: left: -10000em; position: absolute; width: 1px; }.

```
en < a < li.active < ul.nav < div#primary-nav < nav < div.navbar < div.header < div#page-h
  ▼<ul class="nav">
    ▼<li class="active">
      <a href="/news">
        News
        <span class="offscreen">, aktiver Navigationspunkt</span>
      </a>
      ▶<ul class="nav">
        </li>
      ▶<li>
      ▶<li>
      ▶<li>
      ▶<li>
```

.offscreen {  
 left: -10000em;  
 position: absolute;  
 width: 1px;  
}

display:none nicht verwenden!

# BARRIEREFREIHEIT

## HILFSMITTEL

Zum Prüfen:

- <http://www.cynthiasays.com/>
- W3C Validator
- Wave Add-on für Browser
- <https://www.wuhcag.com/wcag-checklist/>

CSS ausschalten: im Web Developer Toolbar von Firefox z.B. oder im Code:

- Verstehe ich meine Website immer noch?
- Könnte ich sie mit verbundenen Augen und Screenreader auch noch bedienen?

# BARRIEREFREIHEIT

## CHECKLISTE

- Alt-Tag bei img
- Title-Tag bei img, a
- Skip-Navi mit CSS Klasse (offscreen)
- Hinweis auf aktiven Navigationspunkt durch nachstehendes <span class=„offscreen“>
- Durchtabbar (tabindex), falls nötig
- Accesskeys, falls nötig
- Farbkontraste
- Anreihung ähnlicher Inhalte durch <li> statt <div>
- Label-Tags bei Formularen
- Ich deaktiviere alle meine CSS-Files => Verstehe ich es immer noch?
- Screenreader aktivieren und mal schauen, ob man zurecht kommt.

# WAS REPETIEREN?

# HAUSAUFGABEN

# AGENDA

## Curriculum WDD

4100\_07

ENTWICKLUNGSUMGEBUNG / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / QUELLCODE, DOM und WEB INSPECTOR / SEMANTIK

CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT

POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT

BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / TABELLEN / IFRAME / FORMULAR

VIDEO/AUDIO / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REPETITION / HELPER TOOLS / TEST II

TESTBESPRECHUNG / WORKFLOW / SCREENDESIGN ÜBERSETZEN

# GEDÄCHTNISLÜCKE

## HTML CSS Theorie

Accessibility

@font-face

inline-Element

transform:translate(\_\_%, \_\_%)

iFrame

Retina

position: absolute

device-pixel-ratio

background-position

table

@media

vertical-align

text-align

min-resolution

:hover

table-cell

block-Element

transform:translate

box-sizing

label

display

margin: 0 auto;

.offscreen

border-box

alt

# HAUSAUFGABEN

# **REKAPITULIEREN**

# **FLOAT / CLEAR**

# REKAPITULIEREN POSITIONIEREN

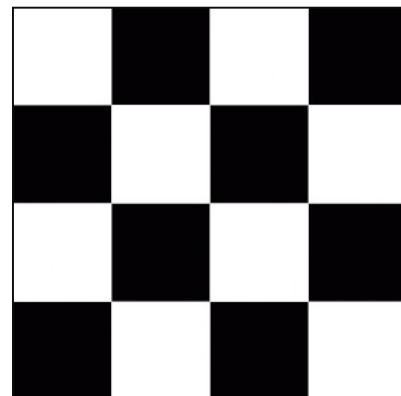
# ZWISCHENTEST

## 40 Minuten

# HAUSAUFGABEN

# KNACKNÜSSE

Schachbrett – mit <table> und CSS-Selektor odd/even



# WICHTIGE FELDER

Schreibe seitlich vom Formular, um welches HTML-Tag und den Typ dazu es sich handelt.

<label for="vorname">Vorname</label>  
etc...

HTML/CSS FORM	
Vorname:	<input type="text"/> <input type="text"
Nachname:	<input type="text"/> <input type="text"
E-Mail:	<input type="email"/> <input type="email"
Passwort:	<input type="password"/> <input type="password"
Telefon:	<input type="tel"/> <input type="tel"
Website:	<input type="url"/> <input type="url"
Männlich <input checked="" type="radio"/> Weiblich <input type="radio"/>	<input type="radio" name="identisch"
Besitze ein Velo <input type="checkbox"/> Besitze ein Auto <input type="checkbox"/>	<input type="checkbox"
Wie viele Unfälle hatten sie schon? <input type="number"/>	<input type="number" min="1" max="5"
Welche Automarke finden Sie toll? <input type="selection"/>	<selection> mit einigen <option>
Ihr Geburtsdatum: <input type="date"/>	<input type="date"
Ihre Lieblingsfarbe: <input type="color"/>	<input type="color"
Wähle eine Zeit: <input type="time"/>	<input type="time"
Wähle deinen Browser: <input type="list"/>	<input list="browsers" und <datalist id="browsers" für Type ahead Funktion
Dein Kommentar: <input type="text"/>	<textarea>
<input type="reset"/> Zurücksetzen	<input type="reset" und <input type="submit" 148
<input type="submit"/> Absenden	

# CSS SELEKTOREN PRAXISBEISPIELE

# CSS - NORMALE SELEKTOREN

## Elemente, Klassen, ID's, Attribute

p { }	Alle p-Elemente
p, a { }	Alle a-Elemente und p-Elemente
p a { }	Alle a-Elemente innerhalb von p (alle Nachfahren)
.foo { }	Alle Elemente der Klasse 'foo'
div.foo { }	Alle div-Elemente der Klasse 'foo'
div#foo { }	Alle div-Elemente mit der id 'foo'
td div.foo { }	Alle div-Elemente der Klasse 'foo' innerhalb von td, d.h. innerhalb einer Tabellenzelle
a[title] { }	Alle e-Elemente, welche das Attribute-Eigenschaft title enthalten
a[title=klicken] { }	Alle e-Elemente, welche das Attribute title="klicken" enthalten

# CSS – KOMBINATORISCHE SELEKTOREN

## Kinder, Nachbarn, Geschwister

`p > em { }` Alle em-Elemente, welche Kinder (erste Einrückungsstufe) von p sind

`h1 + p { }` Das erste p-Elemente, welche als "Vor-"Nachbar h1 hat

`h1 ~ p { }` Alle p-Elemente, welche nachstehende Geschwister von h1 sind

# CSS – PSEUDOKLASSEN

## Spezielle nutzerbeeinflusste Zustände von Elementen

a:hover

a-Element bei Mauskontakt

a:focus

a-Element, wenn es mit fokussiert wird  
(reinklicken, mit Tabulator "reintaben")

a:visited

a-Element nach dem Anklicken

a:active

a-Element beim Mausklicken

a:first-child

das erste a-Element

a:nth-child(odd/even)

das ungerade/gerade a-Element

tr:nth-child(3n + 4)

jede 3. Zeile, zum ersten Mal ab Zeile 4

tr:last-child

das letzte tr-Element

# CSS PSEUDOELEMENTE

## Spezielle nutzerbeeinflusste Zustände von Elementen

p::first-line {}      Erste Zeile von p bis zum Umbruch durch p's Breite

p::first-letter {}      Erster Buchstabe von p

p::before {}      Inhalte vor p  
meist mit CSS-Regel: content: "\*"; oder content: url(bilder/pfeil.png);

p::after {}      Inhalte nach p  
meist mit CSS-Regel: content: "\*"; oder content: url(bilder/pfeil.png);

# BESTANDES- AUFNAHME:

WER MÖCHTE WAS  
REKAPITULIERN?

# REKAPITULIREN EINZELNER THEMEN

## SCHAUE DIR DIE AGENDA AN

- Schreib auf die Karten deinen Namen und je ein Thema auf, worin dich unsicher fühlst:
  - Rot: (sehr unsicher)
  - Gelb: (unsicher)
  - Grün: (geht so)

# ZWISCHENTEST

## 40 Minuten

# AGENDA

## Curriculum WDD

4100\_07

ENTWICKLUNGSUMGEBUNG / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / QUELLCODE, DOM und WEB INSPECTOR / SEMANTIK

CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN

4100\_08

FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT

POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN

4100\_09

NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT

BILDER / HINTERGRUND / CSS SPRITES / FAVICON

4100\_10

HAUSAUFGABEN / TABELLEN / IFRAME / FORMULAR

VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT

4100\_11

HAUSAUFGABEN / REPETITION / HELPER TOOLS / TEST II

TESTBESPRECHUNG / WORKFLOW / SCREENDESIGN ÜBERSETZEN / FUSSBALL FÜR HIRN

# AUTOPREFIXES

## für unterschiedliche CSS-Entwicklungsstadien der Browser

Spezielle Präfixe, welche man für neu entwickelte CSS-Eigenschaften voranstellt:

- webkit- (Chrome, Safari, neuere Versionen von Opera.)
- moz- (Firefox)
- o- (Old versions of Opera)
- ms- (Internet Explorer)

Nutze die Erweiterung Autoprefixer in Brackets!

zum Beispiel:

```
-webkit-transform: translate(-50%, -50%);  
transform: translate(-50%, -50%);
```

Wo nachschauen?

- <http://pleeease.io>
- <http://shouldiprefix.com>
- <http://canluse.com>

Oder mit einem JavaScript-Framework automatisieren

- leider ;( nur möglich, wenn nicht mit @import gearbeitet wird:
- prefix-free

# EINMITTEN VON ELEMENTEN

## Zusammenfassung der CSS-Einmittungstechniken

- Zentriert (aber auch nur horizontal oder vertikal möglich) ab IE 11:
  - das Kind-Element hat:  
**position: relative; top: 50%; left: 50% ;**  
und Korrektur: **transform: translate(-50%,50%);**
- Horizontal – für alle Browser:
  - für INLINE Kind-Elemente:  
**text-align: center;** im Eltern-Element
  - für BLOCK Kind-Elemente:  
**width: \_\_px; margin: 0 auto;** im Kind-Element
- Vertikal – für alle Browser:
  - egal ob INLINE oder BLOCK Kind-Element:  
**display: table-cell; vertical-align: middle;** im Eltern-Element

# REKAPITULIEREN

# BARRIEREFREIHEIT

# BARRIEREFREIHEIT

## An dies musst du denken

- Durchtabben logisch? Hilf dir sonst mit Attribut tab-index="Nummer"
- Alt- und title-Attribut für Bilder
- Title-Attribut für Links
- Untertitel für Videos nötig?
- Transkription für Videos nötig?
- Div mit Skip Links zu Navi und Content, welches offscreen ist:

```
<div id="access-links">
  <a href="#skip-to-content">zum Inhalt</a>
  <a href="#primary-nav">zur Hauptnavigation</a>
</div>
```
- und ➔

# BARRIEREFREIHEIT

## CHECKLISTE

- Alt-Tag bei img
- Title-Tag bei img, a
- Skip-Navi mit CSS Klasse (offscreen)
- Hinweis auf aktiven Navigationspunkt durch nachstehendes <span class=„offscreen“>
- Durchtabbar (tabindex), falls nötig
- Accesskeys, falls nötig
- Farbkontraste
- Anreihung ähnlicher Inhalte durch <li> statt <div>
- Label-Tags bei Formularen
- Ich deaktiviere alle meine CSS-Files => Verstehe ich es immer noch?
- Screenreader aktivieren und mal schauen, ob man zurecht kommt.

# MASSNAHMEN IM CODE

## Kleiner Aufwand, grosse Wirkung

- alt-Attribut für <img>
- title-Attribut title für Links (a hrefs) und Bilder
- offscreen-Elemente wie z.B. Skip-Menü und Orientierungs-Textbausteine, z.B. ", aktiver Navigationspunkt"
- accesskey Attribute für Tastenkombinationen (alt+Buchstabe)
- tab-index für nummerbasiertes Durchtabben mit Tabulator (tab-index="10")
- Schriftgrösse
- Farbkontraste

# SUBNAVIGATION

# SUBNAVIGATION

## Verschachtelung von Listen <ul>

```
/* SUBNAVIGATION: */
#subnavi {
  clear:both;
  position:absolute;
  z-index:1000;
  top:27px;
  left: -9999px;
  /* ab IE 10: */
  /* 1. Wert: Animationsdauer / 2. Wert:
  Verweildauer nach der Animation */
  /* transition: 0s 5s; */

}

#hauptnavi2:hover #subnavi{
  left: 0px;
  /* ab IE 10: */
  /* transition: 0s; */
}
```

```
<nav>
  <!-- ul ist eine ungeordnete Liste -->
  <ul id="mynavi" >
    <li class="aktivstatus">Home</li>
    <li id="hauptnavi2" style="position: relative; "
      <a href="html/beispiele.html" title="Link zu Webpage Beispiele">Beispiele</a>

      <ul id="subnavi">
        <li style="width:100px; "><a href="#"> Beispiel A</a></li>
        <li style="width:100px; "><a href="#">Beispiel B</a></li>
        <li style="width:100px; "><a href="#">Beispiel C</a></li>
      </ul>

    </li>
    <li><a href="html/barrierefreiheit.html" title="Link zu Webpage Barrierefreiheit">Barrierefreiheit</a></li>
    <li><a href="html/helper.html" title="Link zu Webpage Helper">Helper</a></li>
  </ul>
</nav>
```



# CONDITIONAL COMMENTS IM HTML

## Wegweisung für einen bestimmten Browser

<http://www.quirksmode.org/css/condcom.html>

```
<!--[if IE 8]>
According to the conditional comment this is IE 8<br />
<![endif]-->
<!--[if IE 9]>
According to the conditional comment this is IE 9<br />
<![endif]-->
<!--[if gte IE 8]>
According to the conditional comment this is IE 8 or higher<br />
<![endif]-->
<!--[if lt IE 9]>
According to the conditional comment this is IE lower than 9<br />
<![endif]-->
```

# HELPER TOOLS

# AUFTAG IN MURMELGRUPPEN

Trage alle Hilfsmittel auf das A3-Mind Map Blatt zusammen und rotiere das Blatt

- Bezuglich Brackets / Erweiterungen
- Bezuglich Online Tools/Websites
- Bezuglich Browser

# SAMMELSURIUM VON HILFSMITTELN

## Browser

Finch.io für chrome

Yslow für chrome

Css peepers für chrome

Dimensions für chrome

Color picker

Web developer

Wappalizer

Ghostery

# SAMMELSURIUM VON HILFSMITTELN

## Online Tools / Websites



Nibbler

Tinypng

Learnlayout

Pintab.co

Reddit.com/r/web\_dev

Stackoverflow

Dribbble

Behance

Awwwards

Screensiz.es

Freecodecamp.org

# SAMMELSURIUM VON HILFSMITTELN

## Brackets & Code Editors



Teletype (atom)

Minimap

Htmlskeleton

Lorem Ipsum

# SAMMELSURIUM VON HILFSMITTELN

## Browser

Allgemein:

- [Conditional Comments](#) für browserspezifische Avisierung

für Chrome:

- PerfectPixel
- What font
- Page Ruler
- Window resize
- Web Inspector
- Wave Toolbar für Accessibility
- Web Developer Toolbar
  - Tilt
  - Cache killer
- Bookmarklets
- Color picker

für Firefox:

- Web Developer Toolbar

# SAMMELSURIUM VON HILFSMITTELN

## Online Tools / Websites

### Validatoren

- <http://w3c.github.io/developers/tools/>
- <http://validator.w3.org>
- <http://validator.w3.org/checklink>
- <https://jigsaw.w3.org/css-validator/>

### Dokumentation

- <http://selfhtml.org/>
- <http://screensiz.es/>
- <http://canluse.com>
- <http://www.quirksmode.org/css/condcom.html>
- <https://developer.mozilla.org/de/>
- Stackoverflow
- <https://websitelaunchchecklist.com/>

### CSS Prefixes

- <http://pleeease.io>
- <http://shouldiprefix.com>
- <http://canluse.com>

### CSS Tools

- <http://css3generator.com> - zum Lernen/Spielen/Ausprobieren, für Bequeme, für on-the-fly Webdesigner
- <http://css3gen.com> - für Buttons – Schritt für Schritt
- <http://css-tricks.com> - Sammelsurium von CSS-Tricks und Hilfestellung
- Unsplash

### Accessibility

- <http://cynthiasays.com>
- <https://www.w3.org/WAI/intro/wcag.php>
- <http://wave.webaim.org/>

### Features und Snippets

- <http://codepen.io/>
- Google Fonts
- Codepen
- Browserstack
- Github (Repository/Versionierung)

### Übungen:

- <http://www.w3schools.com/html/exercise.asp> (für HTML, gratis)
- <http://www.w3schools.com/css/exercise.asp> (für CSS, gratis)
- <https://www.codecademy.com/learn/web> (gratis)
- <https://jsfiddle.net/> – HTML, CSS , JS und Resultatefenster in einem.

### Tutorials

- <https://www.youtube.com/watch?v=1A1D2bCcBX8> (gratis)
- <https://www.codeschool.com/pricing> (kostenpflichtig)
- <https://www.video2brain.com/de/> (resp. Lynda) (kostenpflichtig)

### Community

- Smashing Magazin
- Team Treehouse
- Reddit
- Webdeveloper.com

# SAMMELSURIUM VON HILFSMITTELN

## Brackets

Erweiterungen:

- Autoprefixer
- Beautify.io (für wohlgeformten Code)
- Brackets Icon
- Bracket Git (Versionierung)
- Emmet (mit Tastenkürzel werden dir HTML-Tags geschrieben)
- Live Preview for Responsive
- W3C Validation
- Quick Search
- Indent Guides (vertikale Hilfslinien für zum Einrücken des Codes)
- Code Outline
- Live Preview for Responsive
- TODO

**Vergiss nicht, deren Aktivierungseinstellung im defaultPreferences.json zu suchen und ins brackets.json zu kopieren und dabei zu aktivieren (Aktivierung (meistens xxx.enabled = true/false)). Danach Brackets neu starten.**

# PRÜFUNGS- VORGABEN