

# 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
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 / SCREENDSIGN 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 ersteren werden Kontrollstruktur angelegt und Entscheidungen getroffen, bei der zweiten werden 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 Endtag, 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 spezifische Bedeutung den Bestandteilen einer Website zu. .

# CSS

# FRONTEND

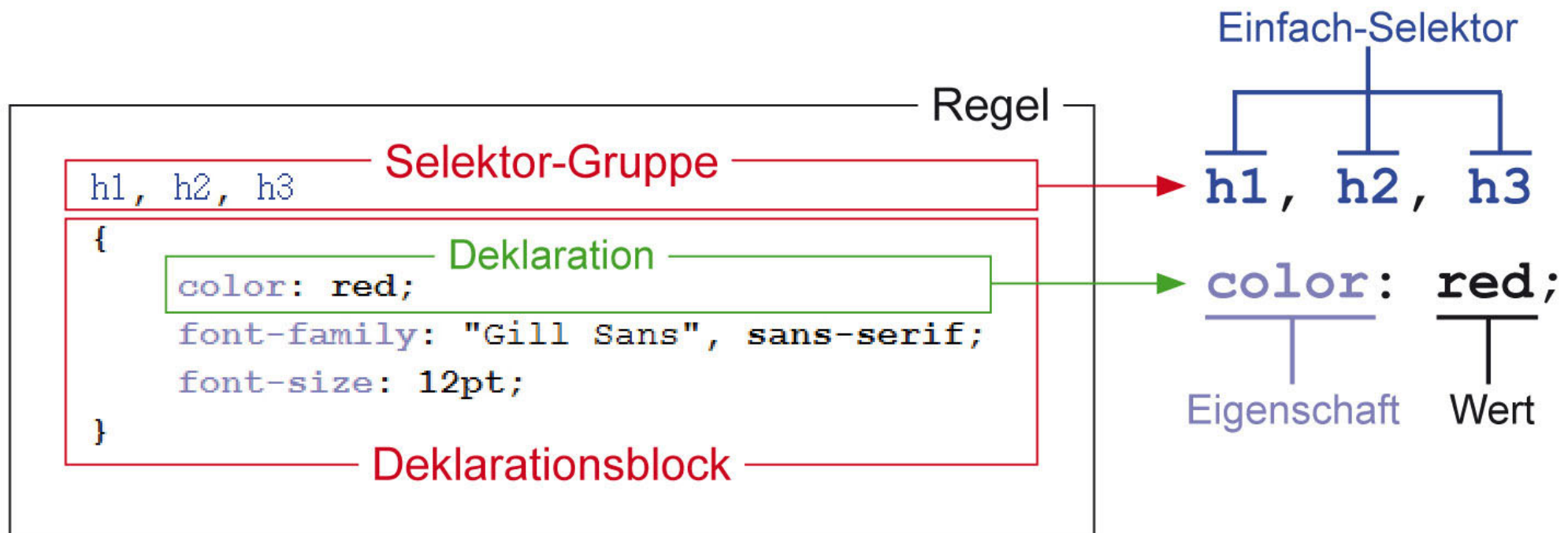
## CSS (Cascading Style Sheet)

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

# 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">
```

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

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

oder

oder

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

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

## Kinder, Nachbarn, Geschwister

<code>p &gt; em { }</code>	Child selector : Alle em-Elemente, welche Kinder (erste Einrückungsstufe) von p sind
<code>h1 + p { }</code>	Adjacent Sibling Selector : Das erste p-Elemente, welche als "Vor-"Nachbar h1 hat
<code>h1 ~ p{ }</code>	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!

im CSS-Teil

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 HTML-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>
```



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>
```

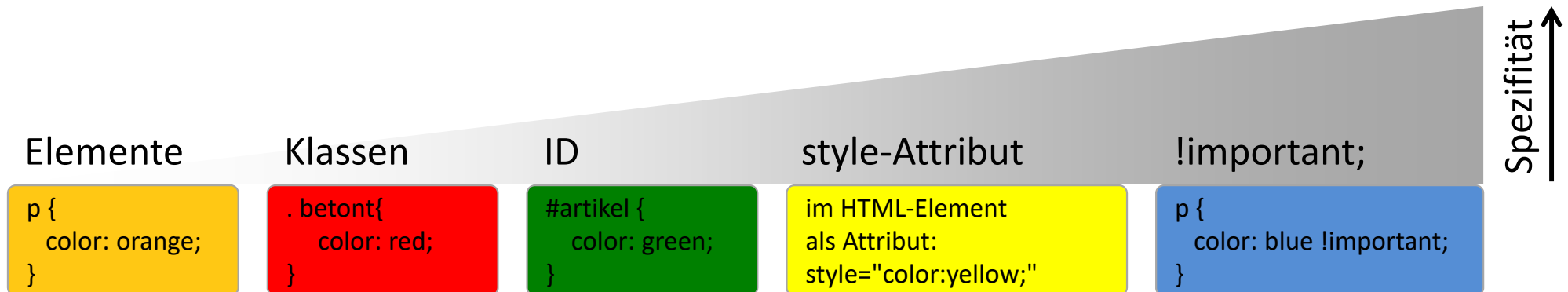


`id="menue"` => der Wert "menue" darf nur einmalig vorkommen!

# 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>`



`</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