

# CSS

## Cascading Style Sheets



*Lektion 01*



# Lernziele / Agenda

- **CSS Regeln**
- **Selektoren**
- **Notationen**
- **Einheiten**
- **Text Eigenschaften**
- **Rahmen**
- **Farben**
- **Positionierung**

- CSS ist die Abkürzung für *Cascading Stylesheets*, eine Formatierungssprache.
- Mit Hilfe von CSS können Webseitenentwickler den Inhalt Ihrer Webseiten von Ihrer Darstellung trennen.
- Die CSS-Regel teilen dem Browser mit, wie bestimmte HTML-Elemente formatiert werden sollen (z.B. Größe, Farbe, Abstand,...)

# CSS-Regeln

- Eine CSS-Regel besteht aus dem zu formatierenden HTML-Element (Selektor) und seiner Formatierung (Deklaration).
  - ♦ **Selektor:** Der Teil einer CSS-Regel, der definiert, welches HTML-Element formatiert werden soll.
  - ♦ **Deklaration:** Der Teil einer CSS-Regel, der die Formatierung enthält. Besteht aus **CSS-Eigenschaft** und **Wert**, getrennt durch einen Doppelpunkt.
  - ♦ Selektor1 [, Selektor2,...]  
{CSS-Eigenschaft1: Wert1; CSS-Eigenschaft2: Wert2; [...]}
  - ♦ Bsp.:  
*h1* {font-family: Arial; color: gray;}

# Selektoren

- Universal-Selektor
- Typ-Selektor
- Class-Selektor (Klassen)
- ID-Selektor

# Universal-Selektor

Universal-Selektoren verknüpfen jedes vorhandene Element mit Stylesheetangaben.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Universal-Selektor</title>
    <style type="text/css">
      *{font-size:20px; color:#ff0000;}
    </style>
  </head>
  <body>
    <h1>Ein "h1"-Tag mit roter Schrift</h1>
    <h2>Ein "h2"-Tag mit roter Schrift</h2>
    <h3>Ein "h3"-Tag mit roter Schrift</h3>
    <p>Ein "p"-Tag mit roter Schrift</p>
  </body>
</html>
```

# Type-Selektor

Typ-Selektoren verknüpfen einen Elementtyp, also ein HTML-Tag mit einer Formatierung.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Type-Selektor</title>
    <style type="text/css">
      h1 { font-size:14px; font-style:italic; }
      p {color:#ff0000;}
    </style>
  </head>
  <body>
    <h1>Eine kursive &Uuml;berschrift</h1>
    <p>Ein roter Text</p>
  </body>
</html>
```

# Class-Selektor

Klassen sind Selektoren mit beliebigen Namen.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Class-Selektor</title>
    <style type="text/css">
      p {font-size:14px; font-style:italic;}
      p.gerahmt {border:1px solid #aaaaaa;}
    </style>
  </head>
  <body>
    <p>Ein kursiver Absatz</p>
    <p class="gerahmt">Ein kursiver Absatz
mit Rahmen</p>
  </body>
</html>
```



# ID-Selektor

ID-Selektoren verknüpfen Elemente über das ID-Attribut mit einer Formatierung.

```
<!DOCTYPE html >
<html>
  <head>
    <title>Irgendein Titel</title>
    <style type="text/css">
      div { background-color:#efefef; border:1px solid #aaaaaa;}
      div#a { position:absolute; top:100px;
        left:100px; width:120px; height:80px; }
      #b { position:absolute; top:150px;
        left:300px; width:120px; height:80px; }
    </style>
  </head>
  <body>
    <div id="a">100 Pixel von oben und 100 Pixel von links</div>
    <div id="b">150 Pixel von oben und 300 Pixel von links</div>
  </body>
</html>
```

## Verschiedene Möglichkeiten der Notation:

### ♦ **Inline-Notation**

- HTML-Tags formatieren
- Abschnitte eines Dokuments mit `<span>` formatieren.

### ♦ **Zentrale Notation:**

Stylesheets zentral im Dokument am Dateianfang (head) deklarieren

### ♦ **Ausgelagerte Notation:**

Stylesheets in eine externen Datei (\*.css) auslagern.

# Inline-Notation

Die Formatierungsangaben stehen im HTML-Element selbst.

Beispiel:

```
<h1 style="font-family:Arial; font-style:italic;  
font-weight:bold; font-size:18pt;">Überschrift</h1>
```

```
<p style="font-family:Arial; font-style:italic;  
font-size:11pt;">Hier steht ein kursiver Text</p>
```

```
<h2 style="font-family:Arial; font-size:12pt;  
font-weight:bold;">  
Hier kommt eine Unter-Überschrift</h2>
```

Nachteil: Inline-Notation wird sehr schnell sehr unübersichtlich.

# Element-unabhängige Formatierung

`<span>` ist der HTML-Tag `<div>` sehr ähnlich. Die Formatierung kann auf zwei verschiedene Arten erfolgen:

```
<span style= "Deklarationen">...</span>
```

```
<span style="background-color:#efefef;">Text</span>
```

Mit `<span style=" ">` legen Sie die CSS-Formatierung direkt am jeweiligen Element fest.

```
<span class= "Klassenname"> ... </span>
```

```
<span class="gruen" >subspecies</span>
```

```
span.gruen {  
    color: #00FF00;  
}
```

Der Markierung wird eine Klasse, die bestimmte Formatierungen enthält, zugewiesen.

# Zentrale-Notation

Formatierungen für HTML-Elemente oder Gruppen von Elementen werden am Dateianfang notiert.

Beispiel:

```
<head>
  <title> CSS </title>
  <style type="text/css">
    <!-- p{background-color:#6699cc; color:#0033cc}
         h2{background-color:#e6dacf;color:#806040}
    --> </style>
</head>
```

# Ausgelagerte-Notation

- Oft besteht ein Webauftritt aus mehreren Webseiten.
- Stylesheets werden in einem separaten Stylesheet Dokument (\*.css) gespeichert und alle Webseiten werden mit dem Stylesheet Dokument verknüpft.
- Ein HTML-Dokument wird mit einem externen Stylesheet mit Hilfe eines <link>-Tags verknüpft:



```
<link rel="stylesheet" type="text/css" href="Verzeichnis/Dateiname">
```

```
<link rel="stylesheet" type="text/css" href="./Dateiname">
```

# Ausgelagerte-Notation

## Beispiel: Datei „styles.css“:

```
body {  
    font-family:Arial;  
    font-size:1cm;}  
  
h1{  
    font-style:italic;  
    font-weight:bold;  
    font-size:1.5cm;}  
  
h2{  
    font-weight:bold;  
    font-size:1.3cm;}  
  
p{  
    font-style:italic;}
```

## Datei „index.html“

```
<html>  
  
<head>  
  
<title>Titel der Seite</title>  
  
<link rel="stylesheet"  
      type="text/css"  
      href="styles.css">  
  
</head>  
  
<body>  
  
<h1>Überschrift</h1>  
  
<p>Hier steht ein kursiver  
    Text</p>  
  
<h2>Hier kommt eine Unter-  
    Überschrift</h2>  
  
</body>  
  
</html>
```

# Wirkung

1. Inline

2. Head

3. extern





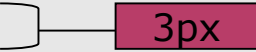


# Text mit CSS stylen

CSS-Eigenschaft	Werte/Beispiele	Beschreibung
font-family	Arial Arial Black Comics Sans ...	Name der Schriftart bzw. Schriftfamilie.
font-size	12px 150% 1.2em 14pt	Schriftgröße
font-style	normal italic (kursiv) oblique (kursiv)	Schriftschnitt
color	blue #0000ff 0,0,255	

# Text mit CSS stylen

CSS-Eigenschaft	Werte/Beispiele	Beschreibung
font-variant	normal small-caps	Kapitälchen
font-weight	normal bold	Gewicht
text-align	left right center justify	Ausrichtung
text-decoration	none underline line-through overline blink	Texteffekte

# Text mit CSS stylen

CSS-Eigenschaft	Werte/Beispiele		Beschreibung
text-indent	Wir wünschen Ihnen einen schönen Urlaub.		Einzug der Erste Zeile
text-transform	none capitalize (jeder erster Buchstabe groß) lowercase (Kleinschreibung) uppercase (Großschreibung)		Groß-/Kleinschreibung erzwingen
line-height	3px	HTL  Spengergasse	Zeilen Abstand
word-spacing	3px	HTL Spengergasse 	Wortabstand
letter-spacing	3px	H T L 	Zeichenabstand

# Schriftfamilien angeben

- Bei der Festlegung einer Schriftart, wird dem Browser mitgeteilt, welche Schriftart verwenden werden soll. Dieser Anweisung kann der Browser nur dann erfüllen, wenn der Schriftart auf dem System des Benutzers installiert ist, sonst greift er auf dem Standardschriftart ( Times New Roman) zurück.
- Mit der CSS-Eigenschaft font-family kann zusätzlich Alternative Schriftarten angegeben werden.

```
body{  
    font-family: Verdana, Geneva, Arial,  
    sans-serif;}
```

# Schriftfamilien definieren

## Schriftfamilien: Schriftarten mit gemeinsamen Eigenschaften

fontfamily	Beschreibung	Beispiel
serif	Für Druck geeignet. Für Webseiten weniger lesbar.	Times New Roman Georgia
sans-serif	Serifenlose Schriften haben keine Abschlussstriche an den Buchstaben und sind geeigneter für Text auf dem Bildschirm	Arial Trebuchet MS Verdana
monospace	Die Zeichen haben eine konstante Breite.	Courier Lucida Console
cursive	Sehen wie Handschrift aus. Eindruck von Verspieltheit.	Comic Sans MS Monotype Corsiva
fantasy	Sind nicht auf alle Systeme verfügbar	Chiller Impact

# Rahmen mit CSS definieren

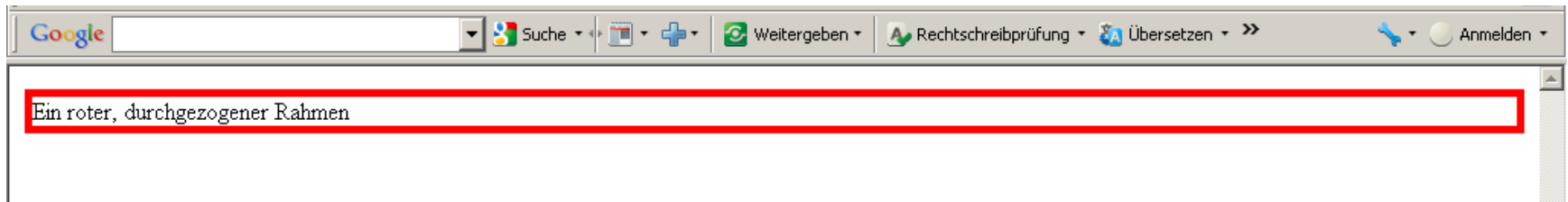
- Die Rahmen-Eigenschaften legt die drei Angaben Breite (*border-width*), Farbe (*border-color*) und Art (*border-style*) für den gesamten Rahmen fest.

Kurzform:

```
Selektor { border: border-weight-wert border-style-wert  
border-color-wert }
```

Beispiel:

```
<p style="border: 5px solid #ff0000;"> Ein roter,  
durchgezogener Rahmen</p>
```



<http://www.css4you.de/borderproperty.html>

# Farben mit CSS definieren

Eigenschaften: **background-color** für Hintergrundfarbe und **color** für Vordergrundfarbe

Beispiel:

```
<p style="color:#0000ff; font-size:18px;" > Ein  
Text in blau </p>
```

```
<table width="500">  
<tr>  
<td style="background-color:#efefef;" > Eine  
Zelle mit grauer Hintergrundfarbe  
</td>  
</tr>  
</table>
```

# Farbangaben

- Farbnamen

```
<p>Text in <span style="color:yellow">gelb</span></p>
```

- Hexadezimale Angabe von RGB-Werten

```
<p>Text in <span style="color:#ff0000">rot</span></p>
```

- Dezimale Angabe von RGB-Werten in CSS

```
<p style="color:rgb(255,0,0)">Text in rot</p>
```

Oder

```
<span style="color:rgb(100%,0,0)">rot</span></p>
```



# Farben

Name	Wert/Farbe
Aqua	#00FFFF
Black	#000000
Blue	#0000FF
Fuchsia	#FF00FF
Gray	#808080
Green	#008000
Lime	#00FF00
Maroon	#800000
Navy	#000080
Olive	#808000
Orange	#FFA500
Purple	#800080
Red	#FF0000
Silver	#C0C0C0
Teal	#008080
White	#FFFFFF
Yellow	#FFFF00

Websichere Farben: <http://www.css4you.de/refcolor216.html>

# Hyperlinks formatieren

- Es gibt vier verschiedene Hyperlinkszustände, die unterschiedlich formatiert werden können.
  - Noch nicht besuchte Hyperlinks (a: link)
  - Links, über denen der Cursor schwebt (a: hover)
  - Gerade angeklickte Hyperlinks (a: active)
  - Besuchte Hyperlinks (a: visited)

# Hyperlinks formatieren

```
a:link {
    color: #ff6600;
    font-weight:bold;
    text-decoration: none;
    padding left: 3px;
    padding right: 3px; }
a:visited {
    font-weight:bold;
    color: #5e7630;
    text-decoration: none;
    padding-left: 3px;
    padding-right: 3px; }
a:hover {
    color: #ffffff;
    background-color: #ff6600;
    text-decoration:underline;
}
a:active {
    color: #FF9933;
}
```

<http://www.echoecho.com/csslinks.htm>

# CSS Einheiten I

Einheit	Typ	Beschreibung
cm	absolut	Zentimeter
em	relativ	Relative Angabe in Bezug auf den höchsten Buchstaben im Element
ex	relativ	Relative Angabe in Bezug auf den Buchstaben x im Element
in	absolut	Zoll (Inch = 2,54 cm)
mm	absolut	Millimeter
pc	absolut	Pica (1 Pica = 12 Punkt)
pt	absolut	Punkt (1 Punkt = 0,351/0,353; <small>US-System Point/Pica</small> )
px	absolut	Pixel
%	relativ	Prozent

[https://wiki.selfhtml.org/wiki/CSS/Wertetypen/Zahlen,\\_Maße\\_und\\_Maßeinheiten](https://wiki.selfhtml.org/wiki/CSS/Wertetypen/Zahlen,_Maße_und_Maßeinheiten)

- **Absolute Werte: bleiben unverändert**
  - ♦ **Pixel (px):** Die einzelnen nebeneinander stehenden Quadrate die Werte zu Helligkeit und Farbe erhalten. Die Größe eines Pixels ändert sich je nach Auflösung.
  - ♦ **Punkt (pt):** Ein typografisches Maß = 1,72 in (bzw. 0,353 mm)
  - ♦ **Pica (pc):** Ein typografisches Maß. Bsp.: Die Schriftgröße 12 pt = 1/6 in (bzw. 4,23 mm) groß = 1 Pica
  - ♦ **Inch (in):** Ein englisches Längenmaß = 25,4 mm
  - ♦ **Millimeter (mm) und Zentimeter (cm)**

- **Relative Werte: sind skalierbar**
  - ♦ **Ems (em):** x-fache Größe des übergeordneten Elements ( $1\text{ em}=100\%$ ). Bsp.: Wenn die Schriftgröße 12 pt definiert ist, dann ist 1 em gleich 12 pt.
  - ♦ **Exes (ex):** x-face Höhe des Buchstabens „x“ des übergeordneten Elements ( $1\text{ ex}=100\%$ ).
  - ♦ **Prozent (%):** Prozentwert gegenüber der Größe des übergeordneten Elements

# CSS Einheiten IIII

Einheit	Beispiele
cm	font-size:1cm; oder top:3.2cm;
em	font-size:1.2em; oder line-height:1.5em;
ex	font-size:1.3ex
in	border-width:0.1in; oder margin-left:1in;
mm	margin-bottom:10mm; oder width:70mm;
pc	font-size:1pc; oder line-height:1.2pc;
pt	font-size:12pt; oder line-height:14pt;
px	border-width:3px; oder margin-right:60px;
%	line-height:120%

# CSS Positionierung:

- Elemente positionieren Allgemein:

- Notation:

Position: die Positions-Anweisung;



# CSS Positionierung:

- Die wesentlichen Basiseigenschaften sind:
- „**static**“ Entspricht keiner Positionierung. In der Praxis unbedeutend.
- „**relative**“ Ein mit „relative“ definiertes Element ändert damit seine Position in Relation zu seiner normalen Lage im Elementfluss.
- „**absolut**“ Ein mit „absolut“ definiertes Element ändert seine Position in Relation zu seinem Elternelement (das „nächsthöhere“ Element, in dem es eingebettet ist). Geht beim Scrollen mit.
- „**fixed**“ Wie bei „absolut“, nur dass das Element beim Scrollen unveränderbar an seiner Position bleibt.

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>

# CSS Positionierung:

- „**top**“ legt den Abstand oberhalb des Elements fest.
- „**bottom**“ legt den Abstand unterhalb des Elements fest.
- „**left**“ legt den Abstand links vom Element fest.
- „**right**“ legt den Abstand links vom Element fest.
  
- Zu beachten: „top“, „bottom“, „left“, „right“ können unterschiedliche Bezugspunkte haben.
- In Kombination mit „position:relative;“ ist das der Abstand zu der Normposition des Elements im Elementfluss.
- Bei „position:absolute;“ und „position:fixed;“ sind es die Abstände innerhalb eines Elternelements, in dem sich das positionierte Element befindet.

**Vielen Dank für  
Ihre  
Aufmerksamkeit!**