

Text Technology SS 17: CSS - Part 2

Björn Rudzewitz

Tübingen University

July 10, 2017

- Pseudo-Classes
- Units

1 Box Model

- Fonts
- Text

2 Cascade Model

Pseudo-Class Selectors

- pseudo-classes behave similarly to classes
- but: dynamically assigned by the browser
- example: a link that changes its color after having been visited

Pseudo-Class Selectors

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

- changes the font color of all visited links to red

Pseudo-Class Selectors

pseudo-class	explanation
:visited	visited link
:focus	focused element (selected input box/clicked element/...)
:hover	element with mouse over it
:active	current active element (e.g. link during click)

more pseudo-classes exist; see e.g. https://www.w3schools.com/css/css_pseudo_classes.asp

Pseudo-Class Selectors

:first-child

```
span:first-child {font-weight: bold;}
```

- selects boldface font for all spans that are the first child of another element

Pseudo-Class Selectors

```
p:first-letter {font-size: 150%;}  
p:first-line {color: green;}
```

- increases the size of the first letter of a p (enclosed in a virtual element)
- only applicable to block-level elements, only certain properties available

Pseudo-Element Selectors

```
q::before {  
  content: "’";  
  color: blue;  
}  
q::after {  
  content: "’";  
  color: red;  
}
```

- insert quotation marks around a quote

source: <https://developer.mozilla.org/de/docs/Web/CSS/::before>

- numbers: int (1) or real (-1.5)
- percentage: [real]% (relative size)
- color¹: blue, #009933, rgb(0, 153, 51)
- length: [number]unit; unit: in, cm, mm, pt, pc, em, ex, px
- URL: @import url(http://example.com/ex.css);

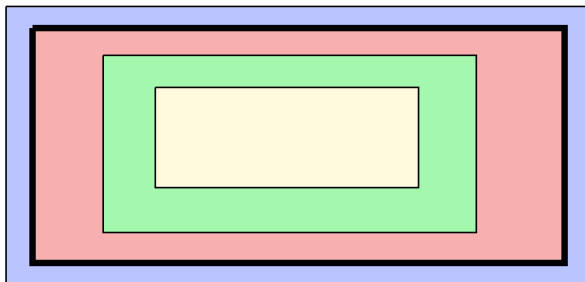
¹https://www.w3schools.com/colors/colors_picker.asp

- most browsers have built-in CSS editors
- can be accessed via *Right-Click* → *Inspect element* or similar
- for older browsers CSS editing plugins exist, like Firebug

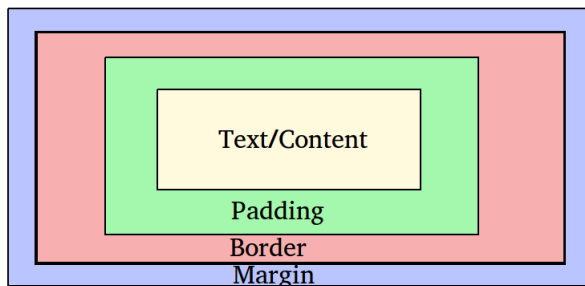
properties:

- color: foreground color
- background-color
- border-color

CSS Box Model



CSS Box Model



Element box:

- content: text/images/widgets/...
- padding: transparent box around content
- border: line (possibly with different styles), encloses content/padding
- margin: transparent box around everything else

- padding, border, margin properties can be set individually and for each direction (top, bottom, left, right)
- property patterns: margin-left:, border-top:, padding:
- values: number units (positive or negative)

- inspect an element in a browser (Firefox/Chrome/...)
- change to the box model tab

border style:

- dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden

```
p.solid {border-style: solid;}
```

https://www.w3schools.com/css/css_border.asp

CSS Box Model

border style:

- border-width:
- border-color:

collapsible properties:

```
p.reds {border-bottom: 4px dashed red;}
```

https://www.w3schools.com/css/css_border.asp

Display Property

- type of *element box* is changed
- *many* values exist, for example:
 - none: don't display
 - inline
 - block: display as block element
 - inline-block

```
p.bl {display: inline}
```

demo: https://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=inline

Float

- changes where elements are floating
- ability to float them to the right or left
- elements after floating elements flow around it², unless *clear:both*; is set on following element

a full explanation of the float concept lies beyond the scope of this seminar

²see https://www.w3schools.com/cssref/pr_class_float.asp

Position Property

- type of positioning
- values: static, relative, absolute, fixed, inherit

```
/* absolute positioning disregarding other elements*/  
p.abs {  
    position: absolute;  
    left: 200px;  
    top: 300px;  
}
```

- CSS defines 5 font families:
 - Serif, Sans-Serif, Monospace, Cursive, Fantasy
- **font family**: combination of **font faces**
- Example: *Times* font family consists of TimesRegular, TimesItalic, TimesBold, ...
- declarations of font families are usually augmented with specific font names (see next slide)

```
p {  
    font-family: Serif;  
}  
  
p.a {  
    font-family: "Times New Roman";  
}  
  
p.b {  
    font-family: "Times New Roman", Georgia, Serif;  
}
```

⇒ providing a specific font family and more generic “backup” font family can increase cross-system compatibility

- **font-weight**: if high, text appears in boldface, if low with light characters

```
div.heavy {  
    font-weight: 800;  
}  
div.light {  
    font-weight: 100;  
}  
div.bold {  
    font-weight: bold;  
}
```


Fonts

- relative and absolute sizes
- possible values: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, [length], [percentage], initial, inherit

```
div.big{  
    font-size: 3em;  
}  
div.other{  
    font-size: 6px;  
}  
div.rel{  
    font-size: 60%;  
}
```

- font styles (italic/oblique) and variants (small-caps) for slight alternations

```
span.it{  
    font-style: italic;  
}  
span.sc{  
    font-variant: small-caps;  
}
```

- text indentation: indent first line of a text block

```
div.ind{  
    text-indent: 5em;  
}
```

- text-align: to where are the lines aligned
- values: left, center, right, justify, inherit
- justify: centered with each line of same size

```
p{  
    text-align: center;  
}
```

- line-height: space between lines
- vertical-align property: font alignment³

```
p{  
    line-height: 20px;  
}
```

³https://www.w3schools.com/cssref/playit.asp?filename=playcss_vertical-align&preval=sub

- `text-transform`: control capitalization
- values: uppercase, lowercase, capitalize, none, inherit
- *capitalize*: make first letter capital

```
p.lc{  
  text-transform: lowercase;  
}
```

- text-decoration: add specific visual effects to text
- values: underline, overline, line-through

```
p.striked{  
  text-decoration: line-through;  
}
```

- text-wrap: handle whitespace
- default behavior (value “normal”): collapse adjacent whitespaces, ignore linebreaks
- values: normal, nowrap, pre, pre-line, pre-wrap
- nowrap: preserve whitespace, new lines only by `
`
- pre: like nowrap, but also linefeed

```
p.ws{  
    text-wrap: nowrap;  
}
```


- cascade: a way for hierarcically applying styles and resolving conflicts in rules
- allows to define general and specific rules
- e.g. having a general style sheet for a web page collection, and a specific style sheet for one page for overriding certain parts

3 combined information sources for rules:

- 1 specificity
- 2 inheritance
- 3 cascade

- more specific rules are preferred over more general rules
- specificity of a rule can be represented with a quadruple $\langle a, b, c, d \rangle$ with $a, b, c, d \in \mathbb{N}_0$

$\langle a, b, c, d \rangle$ with $a, b, c, d \in \mathbb{N}_0$

- a: inline style declaration
- b: id declaration
- c: class/attribute declaration
- d: element and pseudo-element declaration

for each occurrence of a declaration, increase the value at the position by 1

Cascade

Specificity

```
h1 {color: red;}  
body h1 {color: green;}
```

```
h2.grape {color: purple;}  
h2 {color: silver;}
```

```
html > body table tr[id="totals"] td ul > li {color: maroon;}  
li#answer {color: navy;}
```

source: Meyer, E. A. (2006). CSS: The Definitive Guide: The Definitive Guide. "O'Reilly Media, Inc.". (Chapter 3.1)

Cascade

Specificity

```
h1 {color: red;} /* 0,0,0,1 */
body h1 {color: green;} /* 0,0,0,2 (winner)*/

h2.grape {color: purple;} /* 0,0,1,1 (winner) */
h2 {color: silver;} /* 0,0,0,1 */

html > body table tr[id="totals"] td ul > li {color: maroon;}
/* 0,0,1,7 */
li#answer {color: navy;} /* 0,1,0,1 (winner) *
```

source: Meyer, E. A. (2006). CSS: The Definitive Guide: The Definitive Guide. "O'Reilly Media, Inc.". (Chapter 3.1)

special cases:

- * selector has a specificity of $\langle 0, 0, 0, 0 \rangle$
- inline declarations always win over declarations in stylesheets (highest digit 1)
- rules with keyword *!important* always win over non-important rules:

```
.code{  
    color: grey !important;  
}
```

- inheritance: child elements inherit properties from parents
- CSS properties are passed on to descendants
- not all properties are inherited (e.g. border)

- inherited values have no specificity
- i.e. a universal selector `*` with specificity $\langle 0, 0, 0, 0 \rangle$ wins over inherited properties
- for each CSS property you can look up in the reference whether it is inherited

Cascade

The Cascade

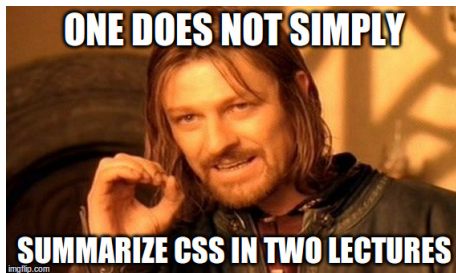
- cascade is a mechanism to combine the style sheet information
- combines multiple style sheets/declarations, inheritance, and specificity

cascade process:

- ① get all declarations for element
- ② sort by weight (important/unimportant), sort by origin (user agent/application)
- ③ sort by specificity
- ④ sort by order (later wins)

Disclaimer

- it's not possible to cover all CSS details, properties, values, ... that exist in this seminar
- you have been presented an excerpt from the huge field of CSS
- feel encouraged to explore and learn more about CSS



source: created with imgflip.com

Further Reading

Good step-by-step tutorial for learning CSS:

<https://www.w3schools.com/css/default.asp>

parts of the slides are inspired by this tutorial (in addition to the references listed below)

CSS Reference: <https://www.w3schools.com/cssref/default.asp>

Elizabeth Castro and Bruce Hyslop. *HTML and CSS: Visual Quickstart Guide*. Pearson Education, 2013.

Eric A Meyer. *Cascading style sheets: The definitive guide*. " O'Reilly Media, Inc." , 2004.