

COMP519 Web Programming

Lecture 5: Cascading Style Sheets: Part 1

Handouts

Ullrich Hustadt

Department of Computer Science
School of Electrical Engineering, Electronics, and Computer Science
University of Liverpool

Contents

① Style Sheets

- Principle

- History

- Overview

② Inline Styles

- Overview

- Properties and Values

③ Further Reading

Style Sheets

- HTML5 follows the **separation of concerns** design principle:
a system should be divided into parts with functionality that overlaps as little as possible
~> in HTML5 semantics and presentation are (mostly) separated
- Concerns of presentation are dealt with using **Cascading Style Sheets** (CSS)

CSS: History

- **1996, W3C** CSS Level 1 (CSS1)
Covers properties of fonts, texts, boxes; colours, pseudo classes
Very limited and buggy support in browsers
- **1998, W3C** CSS Level 2 (CSS2)
Covers all of CSS1 plus selectors, media types, box model, tables, bidirectional text, etc
Browser support still patchy
- **2011, W3C** CSS Level 2 Revision 1 (CSS2.1)
Fixes errors, removes features from CSS2 with insufficient browser supports, adds others
- **2011 onwards, W3C** CSS Level 3
Follows a modular specification approach (lots of smaller specifications instead of one very big one)
Covers all of CSS2.1 plus media queries, name spaces, flexible layout, etc

Cascading Style Sheet

- Cascading Style Sheet are called **cascading** since they can be defined at different levels:
 - ① **Inline styles** apply to a single HTML **element**
 - ② **Document style sheets** (aka **embedded styles** aka **internal style sheets**) apply to all elements within the body-element of an HTML document
 - ③ **External style sheets** can be linked and applied to numerous HTML documents
- Lower-level (that is, lower-numbered) style sheets can typically **override** higher-level style sheets
- Style sheets can specify how HTML elements should be presented on **different screen types** and/or in **print**
- Web users can specify **personalised style sheets** that can override the styles associated with an HTML document

This can be useful, for example, to specify larger font sizes for visually impaired users

Cascading Style Sheet

Cascading Style Sheet cover three aspects of style and layout:

- Styling text

Deals with fonts, text colors, list item style, etc

- Styling boxes

Deals with properties of boxes:

- For the purpose of style and layout, all HTML elements are considered to be boxes with the following properties:



- Size of content, padding, border, margin, and box as well as their style (e.g. colour) can be changed
 - Arranging boxes
- Deals with the relative position of boxes to each other

Inline Styles

Using the `style` attribute, one can specify the `presentation style` for a single HTML element:

Within the value of the `style` attribute, list a sequence of *property: value* pairs separated by semi-colons

Example:

To make an `h1` element red with an italic font shape

- set the `font-style` property to *italic* and
- set the `color` property to red

```
<h1 style="font-style: italic; color: red">A Red H1 Heading</h1>
```

A Red H1 Heading

For a complete list of properties and a description of each see

Mozilla and individual contributors: CSS reference. *MDN Web Docs*, 13 October 2017.

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

(accessed 14 October 2017)

CSS Values: Lengths

- One of the most important types of values in CSS are **lengths**
- CSS distinguished between **absolute lengths**, measured in **absolute (length) units**, and **relative lengths**, measured in **relative (length) units**
- **Relative length units** specify a length relative to another length property that is usually under user control
 - the size of fonts (controlled by browser settings)
 - the size of the viewport (controlled by resizing the browser window)
- **Absolute length units** are various multiples of a 'physical measurement', for example, the size of a screen pixel

CSS Values: Lengths

Relative Length Units

em	current font-size
ex	x-height of the current font
ch	width of "0" (zero) in the current font
rem	font-size of the root element (typically 16px)
vw	1% of the width of the viewport
vh	1% of the height of the viewport
vmin	1% of viewport's smaller dimension
vmax	1% of viewport's larger dimension

The [viewport](#) is the user's visible area of a web page, typically determined by the size of the browser window

CSS Values: Lengths

Absolute Length Units

cm 'centimeter'

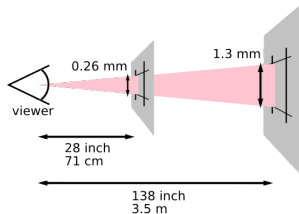
mm 'millimeter'

in 'inch' (1in = 96px = 25.4mm)

px pixel (1px = 1/96 in)

pt point (1pt = 1/72 in)

pc pica (1pc = 12pt)



- A CSS pixel is the visual angle of one pixel on a device with a pixel density of 96dpi and a distance from the reader of 28in (0.2646mm).
- A web client chooses the number of physical pixels that best approximates a CSS pixel
- The multipliers between two absolute length units are always as indicated above

Mobile Devices and Lengths

- The interplay of CSS length units and mobile devices is problematic
 - On mobile devices in portrait orientation, web clients typically
 - render web pages in a **viewport** that is wider than the physical screen
 - then shrink the result to the physical screen
- ~> HTML elements appear to shrink when switching from landscape to portrait orientation
- ~> Media queries might not work as expected

The **viewport meta tag** allows to control this behaviour on web clients

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

CSS Values: Font sizes

CSS allows to specify the size of a font in a number of different ways, including:

- Via a **length**
- Via a **percentage**:
Scales the font relative to the size of the current font
- Via one of the following **keywords**:

Keyword	Typical scaling factor
xx-large	200%
x-large	150%
large	110%
medium	100%
small	80%
x-small	60%
xx-small	55%

CSS Values: Colours

CSS allows to specify colours in a number of different ways, including:

- Via **colour keywords**:

CSS has about 140 names for a wide range of colours, for example:

red, **green**, **blue**, **gold**, **peachpuff**, **snow**, **rebeccapurple**

- Via **hexadecimal notation**: **#RRGGBB**

defines a colour as a mix of the colours red, green and blue

R (red), **G** (green), **B** (blue) are hexadecimal characters (0-9, A-F) representing three numbers between 0 and 255, which in turn, specify the intensity of that colour

Example: #FF000 defines the color red

- Via **functional notation**: **rgb(R,G,B)**

defines a colour as a mix of the colours red, green and blue

R (red), **G** (green), **B** (blue) are either decimal numbers between 0 and 255 or percentages 0% to 100% that again specify the intensity of that colour

Inline Styles: Properties and Values

font-family:	Courier, monospace
font-style:	italic
font-size:	12pt larger
font-weight:	bold
color:	AliceBlue #F0F8FF
background-color:	AliceBlue #F0F8FF
text-decoration:	underline none
text-align:	left right center justify
text-indent:	10pt
vertical-align:	top middle bottom

```
<p style="font-family:Arial,sans-serif; text-align:right">  
  This is a right-justified paragraph in a sans serif font  
  with some <span style="color:green">green text</span>  
  and <a style="color:Red; text-decoration:none; font-size:larger;"  
    href="page01.html">here</a> is a formatted link.</p>
```

This is a right-justified paragraph in a sans serif font with some
green text and here is a formatted link.

Inline Styles: Properties and Values

list-style-type:	disc		circle		square		none
list-style-type:	decimal		lower-alpha		upper-roman		
margin-left:	0.1in		padding-top:		0.1in		
margin-right:	3em		padding-bottom:		3em		
margin:	5%		padding:		5%		

```
<ol style="list-style-type:upper-alpha">  
  <li style="color:red">first thing</li>  
  <li style="color:blue">  
    <span style="color: black">second thing</span></li>  
</ol>
```

A. first thing

B. second thing

Inline Styles: Properties and Values

list-style-type:	disc		circle		square		none
list-style-type:	decimal		lower-alpha		upper-roman		
margin-left:	0.1in		padding-top:		0.1in		
margin-right:	3em		padding-bottom:		3em		
margin:	5%		padding:		5%		

```
<ul style="list-style-type:square; background: Red; padding: 20px;">  
  <li style="background-color:#ff9999; padding: 5px; margin: 5px">  
    one thing</li>  
  <li style="background-color:#ff9999; padding: 5px; margin: 5px">  
    or another</li>  
</ul>
```

- one thing
- or another

Inline Styles: Properties and Values

border-width thin | thick | 5
border-color Blue | #0000FF
border-style dashed | dotted | double | none

```
<table style="border-style:dashed; border-color:blue">  
  <caption style="color: red; font-style: italic;  
    text-decoration: underline">Student data</caption>  
  <tr style="background-color: red"><th>Name</th><th>Age</th></tr>  
  <tr><td>Chris Smith  </td><td>19</td></tr>  
  <tr><td>Pat Jones    </td><td>20</td></tr>  
  <tr><td>Doogie Howser</td><td>15</td></tr>  
</table>
```

Student data

Name	Age
Chris Smith	19
Pat Jones	20
Doogie Howser	15

Revision and Further Reading

Read

- Chapter 11: Introducing Cascading Style Sheets
- Chapter 12: Formatting Text
- Chapter 13: Colors and Backgrounds
- Chapter 14: Thinking Inside the Box

of

J. Niederst Robbins: Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (5th ed).

O'Reilly, 2018.

E-book <https://library.liv.ac.uk/record=b5647021>