COMP519 Web Programming

Lecture 7: Cascading Style Sheets: Part 3
Handouts

Ullrich Hustadt

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

Contents

1. Document Style Sheets
   - Attribute Selectors
   - Pseudo Classes
   - Pseudo Elements

2. External Style Sheets
   - Overview
   - Media Attribute and Adaptive Design

3. Further Reading

Attribute Selectors

- It is possible to specify selectors based on the presence or value of an attribute in HTML elements:

Selector	Example	Selects
`[`*attr*`]`	`[title]`	All elements with a `title` attribute
`[`*attr*`=`*val*`]`	`[target=_blank]`	All elements with `target="_blank"`

Attribute Selectors

- It is possible to specify selectors based on the presence or value of an attribute in HTML elements:

Selector	Example	Selects
[*attr*|=*val*]	[lang|=en]	All elements whose lang attribute value begins with the word "en" (delim " and -)
[*attr*^=*val*]	[href^="https"]	All elements whose href attribute value begins with the string "https"

Examples:	p[class|=red]	matches	<p class="red">
		and	<p class="red-bold">
		but not	<p href="red.bold">
	p[class^=red]	matches	<p class="red">
		and	<p class="red-bold">
		and	<p class="red.bold">

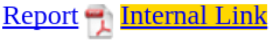
Attribute Selectors

- It is possible to specify selectors based on the presence or value of an attribute in HTML elements:

Selector	Example	Selects
[*attr*$=*val*]	[href$=".pdf"]	All elements whose href attribute value ends with the string ".pdf"
[*attr**=*val*]	[href*="csc"]	All elements whose href attribute value contains the substring "csc"
[*attr*∼=*val*]	[class∼=alert]	All elements whose class attribute value contains the word "alert"

Attribute Selectors

```
<!DOCTYPE html>
<html lang="en-GB">
 <head>
  <title>Attribute Selectors</title>
  <style type="text/css">
    a[href$=".pdf"] {
      padding-right: 20px;
      padding-bottom: 5px;
      background-image: url(pdf-icon.png);
      background-repeat: no-repeat;
      background-position: top right;
    }
    a[href^="#"] {
      background-color: gold;
    }
  </style>
 </head>
 <body>
   <a href="report.pdf">Report</a> <a href="#i1">Internal Link</a>
 </body>
</html>
```


Report Internal Link

Pseudo Classes

- A CSS pseudo class is a keyword added to a selector that indicates a particular state of the selected element
- Depending on the type of element, its state could include
  - the navigation history (e.g., whether a link has been visited)
  - the status of its content (e.g., whether a checkbox has been checked)
  - the position of the mouse pointer (e.g., over the element)
- Style directives involving a pseudo class take the form

```
selector:pseudo-class { property: value; ... }
```

Pseudo Classes

- A CSS pseudo class is a keyword added to a selector that indicates a particular state of the selected element
- Pseudo classes include

Pseudo Class / Example	Selects
`:hover` `a:hover`	Links on mouse over
`:visited` `a:visited`	All visited links
`:link` `a:link`	All unvisited links

Pseudo Classes

- A CSS pseudo class is a keyword added to a selector that indicates a particular state of the selected element
- Pseudo classes include

Pseudo Class / Example	Selects

`:nth-child(n)`
`p:nth-child(1)`
	Every p element that is the first child of its parent

`:nth-last-child(n)`
`p:nth-last-child(2)`
	Every p element that is the second last child of its parent

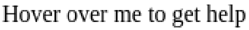
`:nth-of-type(n)`
`p:nth-of-type(1)`
	Every p element that is the first p element of its parent

`:nth-last-of-type(n)`
`p:nth-last-of-type(2)`
	Every p-element that is the second last p element of its parent

- There are a number of additional pseudo classes that relate to the state of form elements

Pseudo Classes: Example

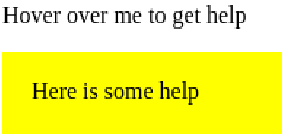
```
<!DOCTYPE html>
<html lang="en-GB">
  <head>
    <title>Pseudo Classes</title>
    <style>
      p.help { display: none;  background-color: yellow;
               padding: 20px;  }
      div:hover p.class {
               display: block; }
    </style>
  </head>
  <body>
    <div>Hover over me to get help
      <p class="help">Here is some help</p>
    </div>
  </body>
</html>
```



Hover over me to get help

turns into

when the mouse



Hover over me to get help

Here is some help

Pseudo Elements

- A CSS pseudo element is a keyword added to a selector that lets you style a specific part of the selected element(s)
- Style directives involving a pseudo element take the form

```
selector::pseudo-element { property:value; ... }
```

- Pseudo elements include

Pseudo Element	Example	Effect
`::first-letter`	`p::first-letter`	Selects the first letter of every p element
`::first-line`	`p::first-line`	Selects the first line of every p element
`::selection`	`::selection`	Selects the portion of an element that is selected by the user

Pseudo Elements

- A CSS pseudo element is a keyword added to a selector that lets you style a specific part of the selected element(s)
- Style directives involving a pseudo element take the form

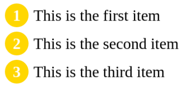
```
selector::pseudo-element { property:value; ... }
```

- Pseudo elements include

Pseudo Element	Example	Effect
`::marker`	`ul::marker`	Select every marker of every `ul` element (experimental)
`::before`	`h1::before`	Creates a pseudo element as first child of every `h1` element
`::after`	`h1::after`	Creates a pseudo element as last child of every `h1` element

Pseudo Elements: Example

```
<!DOCTYPE html>
<html lang="en-GB">
  <head>
    <title>Pseudo Elements</title>
    <style>
      .custom-counter    { list-style-type:   none;
                           padding-right:     0px;
                           margin-left:       0px;   }
      .custom-counter li { counter-increment: step-counter;
                           padding-bottom:    10px;  }
      .custom-counter li::before {
         content:          counter(step-counter);  margin-right: 5px;
         background-color: gold;                   color:        white;
         font-weight:      bold;                   padding:    3px 8px;
         border-radius:    17px;
       }
    </style>
  </head>
  <body>
    <ol class="custom-counter">
      <li>This is the first item</li>
      <li>This is the second item</li>
      <li>This is the third item</li>
    </ol>
  </body>
</html>
```



1 This is the first item

2 This is the second item

3 This is the third item

External Style Sheets

- External style sheets place style directives in a separate file or files
  - Multiple HTML documents can link to the same style sheet
  - Allows for a consistent look across multiple HTML documents on a website
  - Makes it possible change a style in a single place which then propagates automatically to all HTML documents that use that style sheet
- External style sheets represent the best separation of content and presentation

External Style Sheets: Example

```
<html lang="en-GB">
 <head>
  <title>External Style Sheets</title>
  <link rel="stylesheet" type="text/css"
        href="myStyle.css" title="myStyle">
 </head>
 <body>
  <h1>Centred Red Heading</h1>
  <p class="indented">This paragraph will have the first line indented,
    but subsequent lines will be left-aligned.
  </p>
  <p>This paragraph will not be indented, all lines are left-aligned
     as per default.
  </p>
 </body>
</html>
```

```
/* myStyle.css COMP519 2019-10-12 */

h1 { color: Red; text-align: center; font-style: italic; }
p.indented {  text-indent: 2em; }
```

External Style Sheets

- Ideally, the developers of a website will place all style directives in one or more external style sheets
- All web pages of that website will then link to those style sheets in order to maintain a consistent look throughout the website
- Using external style sheets
  - lowers the development cost and
  - simplifies web design,

  as the developers of the web pages can now focus on the content and semantic markup of the pages,
  the presentation is left to the pre-defined styles

Multiple Style Sheets and Adaptive Design

- It is possible to use several external style sheets, and it is quite common to use one that specifies styles for a web browser and another for printing
- The `media` attribute of the `link` element allows to select which external style sheets to use for which medium

Example:

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

- Different style sheets based on media features, e.g. screen resolution, are often used in adaptive design
- Adaptive design uses different web pages or different layouts/styles depending on media features

The Media Attribute

- The value of the media-attribute is a string that specificies a boolean combination of criteria based on media types and media features
- Boolean operators are:
  and (conjunction), not (negation), , (disjunction)
- Media types are:
  print, screen, speech, all
- Media features include:

Feature	Value
orientation	portrait or landscape
resolution | min-resolution | max-resolution	resolution in dpi
width | min-width | max-width	width in px
height | min-height | max-height	height in px

Example: `screen and (min-width:1920px)`

CSS: Resources

1. Refsnes Data: CSS Tutorial. *W3Schools Site.* 12 October 2017.
`https://www.w3schools.com/css/default.asp`
(accessed 12 October 2017).
2. Refsnes Data: CSS Colors. *W3Schools Site.* 12 October 2017.
`https://www.w3schools.com/cssref/css_colors.asp`
(accessed 12 October 2017).
3. Refsnes Data: CSS Units. *W3Schools Site.* 12 October 2017.
`https://www.w3schools.com/cssref/css_units.asp`
(accessed 12 October 2017).
4. Web Education Community Group: Web Standards Curriculum: CSS.
Web Education Community Group Wiki. 28 January 2012.
`http://www.w3.org/community/webed/wiki/Main_Page#CSS`
(accesses 12 October 2017).
5. Bert Bos: Web Style Sheets: CSS tips & tricks. *W3C.* 27 July 2017.
`https://www.w3.org/Style/Examples/007/`
(accessed 12 October 2017).

Revision and Further Reading

Read about Selectors and Inheritance in

- Chapter 11: Introducing Cascading Style Sheets
- Chapter 12: Formatting Text
- Chapter 13: Colors and Backgrounds

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`