



CSS Levels & Selectors

Tessema Mengistu (Ph.D.)

Department of Computer Science

Virginia Tech

mengistu@vt.edu

Outline

- Introduction
- Levels of Style Sheets
- CSS selectors

Introduction

- Controls presentation of HTML documents
- CSS3 is the latest version
 - The CSS1 specification was developed in 1996
- Allows to impose a standard style:
 - Whole document
 - Whole collection of documents

Levels of Style Sheets

- There are three levels :
 - Inline
 - Specified for a specific occurrence of a tag
 - Document-level
 - Apply to the whole document in which they appear
 - External style sheets
 - Can be applied to any number of documents
- Possibility of conflict

Style Specification Formats: Inline

- Inline

- Style sheet appears as the value of the style attribute of a tag
- General form:

```
style = "property_1: value_1;  
        property_2: value_2;  
        ...  
        property_n: value_n"
```

- Example

```
<ul>  
  <li style="font-size: x-large; font-style:  
    italic"> Hello There!!</li>  
</ul>
```

Style Specification Formats : Document-level

- Document-level style sheets appear in the head of the document
- Style sheet is a list of rules that are content of a `<style>` tag
 - The `<style>` tag must have the `type` attribute set to `"text/css"`
- General form:

```
<style type = "text/css">  
  <!--  
    rule list  
  -->  
</style>
```

Style Specification Formats : Document Level

- Example:

```
...  
<style type = "text/css">  
  p {  font-size: 24pt;  
        font-family: Ariel;  
        color: red  
      }  
</style>  
...  
<p>Now is the time to learn CSS </p>  
...
```

Style Specification Formats : External

- External style sheets dwell in separate files, potentially on any server on the Internet
 - File with the extension “.css”
- A list of style rules, just as in the content of a <style> tag for a document-level style sheet
- The CSS file then needs be linked to the web document that uses it
 - Written as text files with the MIME type text/css
 - Must be linked to from an html file using <link> such as

```
<link rel = "stylesheet"  type = "text/css"
  href = "http://www.wherever.org/termpaper.css">
</link>
```


Conflict Resolution

- A conflict occurs when there are two or more values for the same property on the same element
- Sources of conflict:
 1. Conflicting values between levels of style sheets
 2. Within one style sheet
 3. Property values can come from style sheets written by the document author, the browser user, and the browser defaults

Conflict resolution – a multistage process

1. Gather all of the style specs from the different levels of style sheets
2. All available specs, from all sources, are sorted by origin and weight, using the following rules:
 - a. Important declarations with user origin
 - b. Important declarations with author origin
 - c. Normal declarations with author origin
 - d. Normal declarations with user origin
 - e. Any declarations with browser (or other user agent) origin
3. If any conflicts remain, sort them by specificity:
 - a. id selectors
 - b. Class and pseudo-class selectors
 - c. Contextual selectors
 - d. Universal selectors
4. If there are still conflicts, use the “most recently seen” policy

CSS Selectors

- There are 5 selectors:
 - Simple selectors
 - Select elements based on name, id, class
 - Combinator Selectors
 - Select elements based on a specific relationship between them
 - Pseudo-class Selectors
 - Select elements based on a certain state
 - Pseudo-elements selectors
 - Select and style a part of an element
 - Attribute selectors
 - Select elements based on an attribute or attribute value

Simple Selector Forms

- The selector is a tag name or a list of tag names, separated by commas, e.g.,

`p`

`h1, h3`

- Examples:

```
h2, h3 {
```

```
    font-size: 40pt;
```

```
}    /* applies to both */
```

Simple Selector Forms

- Class Selectors

- Allow different occurrences of the same tag to use different style specifications
- A style class has a name, which is attached to a tag name

```
p.narrow {property:value list}
```

```
p.wide {property:value list}
```

- The class you want on a particular occurrence of a tag is specified with the class attribute of the tag
- Use example,

```
<p class = "narrow">
```

```
...
```

```
</p>
```

```
...
```

```
<p class = "wide">
```

```
...
```

```
</p>
```

Simple Selector Forms

- An **id selector** allows the application of a style to one specific element
- General form:

```
#specific-id {property-value list}
```

- Example:

```
#section14 {  
    font-size: 20  
}
```

. . .

```
<h2 id="section14"> my heading </h2>
```

```
/* the id must be defined for the element (via its id  
   attribute) */
```

Simple Selector Forms

- Universal Selector
 - Start with “*” and apply to all elements in a document
 - Example
 - `* {color: red}`

Combinator Selectors

- Descendant selector

```
ol ol li          /* describe a hierarchy, applies only to li */
```

```
body b em {font-size: 40pt;}  
        /* only to em in the hierarchy.*/
```

- Child Selector (>)

```
div > p  
    /* describe a parent-child relationship, applies only to p */
```

```
div >p {font-size: 40pt;}  
    /* only to p that is direct child of div.*/
```

- Adjacent Sibling Selector (+)
- General Sibling Selector (~)

Pseudo-Class Selectors

- Pseudo classes are style classes that apply when something happens, rather than something simply exists
- The class names begin with colons
- Two basic pseudo classes:
 1. `Tag-name:hover` classes apply when the mouse cursor is over the element
 2. `Tag-name:focus` classes apply when an element gets focus
- More pseudo classes:

```
a:link      {color:green;}  
a:visited   {color:green;}  
a:active     {color:yellow;}
```

Pseudo-Element Selectors

- Can be used to style specified parts of an element.
- For example, it can be used to:
 - Style the first letter, or line, of an element
 - Insert content before, or after, the content of an element

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

```
p::first-line {  
    color: red;  
    font-variant: small-caps;  
}
```

Attribute Selectors

- Selector is used to select elements with a specified attribute
- Example:

```
p[font-style] {  
    color: yellow;  
}
```

References

- W3Schools.com
 - <https://www.w3schools.com/css/default.asp>