

Cascading Style Sheets

Mary ET Boyle Ph.D

Department of Cognitive Science

CSS History

- ❑ W3C recommended the adoption of a standard set of style sheets.
 - ❑ 1996 Cascading Style Sheets Level 1 (CSS1) was introduced.
 - ❑ 1998 CSS2 was introduced.
More features added.
 - ❑ The purpose was to give HTML authors more formatting support
 - ❑ Apply uniform styles to multiple documents
-

Why use style sheets?

- ❑ It is all about the separation of contents from presentation.
 - ❑ Gives you more control over document format.
 - ❑ Allows you to change!
You can change the presentation format without changing the content!
 - ❑ Apply one style sheet to multiple documents. It's efficient!!
-

CSS Syntax

- ❑ CSS contain rules and declarations that instruct the browser how to display certain elements.
- ❑ Many styles can be applied to XHTML documents.
- ❑ CSS is extremely large.
I will cover a small subset in class.
- ❑ Go here for more information:
<http://www.w3.org/Style/CSS>

Defining the style sheets...

- In order to use a style sheet you need to tell your XHTML document where to locate the style definitions.
 - There are three ways:
 1. Linked Style Sheets
 2. Global Style Sheets
 3. Inline Styles
-

Linked Style Sheets

- ❑ Linked style sheets are stored in a separate HTML file.
 - ❑ Linked style sheets provide style definitions to many documents.
 - ❑ Each document has to simply reference a single style sheet.
-

Linking to a separate style sheet

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

rel attribute must be set to the value "stylesheet".

href attribute gives the URL for your style sheet.

type attribute declares the file to be plain text containing CSS styles.

link tag should be placed in the document's head.

Global Style Sheets

- ❑ Global Style Sheets are defined in the XHTML document.
 - ❑ It is located within the `<head>` section of the document.
 - ❑ It is within the `<style> </style>` element
-

Example of global style sheet:

```
<head>
```

```
<title> This is an example </title>
```

```
<style type="text/css">
```

```
<!--
```

```
  p {
```

```
    color: purple;
```

```
    font-family: arial
```

```
  }
```

```
-->
```

```
</style>
```

```
</head>
```

This defines the type of style sheet to be plain text CSS

The comments are to shield your code from older browsers that may not support style sheets.

Inline Styles

- Inline styles are applied to a single element within the start tag of the element.
 - If one wants to assign certain properties to the text within a paragraph, we would include style definitions inside that particular tag.
-

Inline style example:

```
<p style="color:purple;  
           font-family:arial" >
```

Paragraph text goes in here..

```
</p>
```

When Style sheets Fight It Out

- ❑ Three different style sheet rules at work and all of them use **p** as the selector.
- ❑ A **linked style sheet** tells the browser to display <p> text in **red**.
- ❑ A **global style sheet** tells the browser to display <p> text in **green**.
- ❑ An **inline style** tells the browser to display <p> text in **brown**.

What is a poor web browser to do???

Example Style sheet

A simple style sheet file might look like the following:

```
/* style.css - a simple style sheet */  
body  
{  
    margin-left: 10%;  
    margin-right: 10%;  
    color: black;  
    background: white;  
}
```

CSS Properties

□ Font

- Font styles: font family
- Type: size, weight, or variant

□ Text

- Define the layout of blocks of text
 - Words and characters
 - Spacing
 - Alignment
 - Transformation (forced uppercase or lowercase)
 - Decoration (underline, overline, and blinking)
-

CSS Properties

□ Color and image

- Color and background formatting of text and images
- Define position of background image

□ Border

- Define the style of borders for elements like tables, images, documents
- Border properties include width, height, style, color, margins, and padding.

□ Display

- Define styles for the structure of the doc.
 - Placement of elements (block or inline)
-

CSS Rules: 2 Parts

Selector

Property Declarations

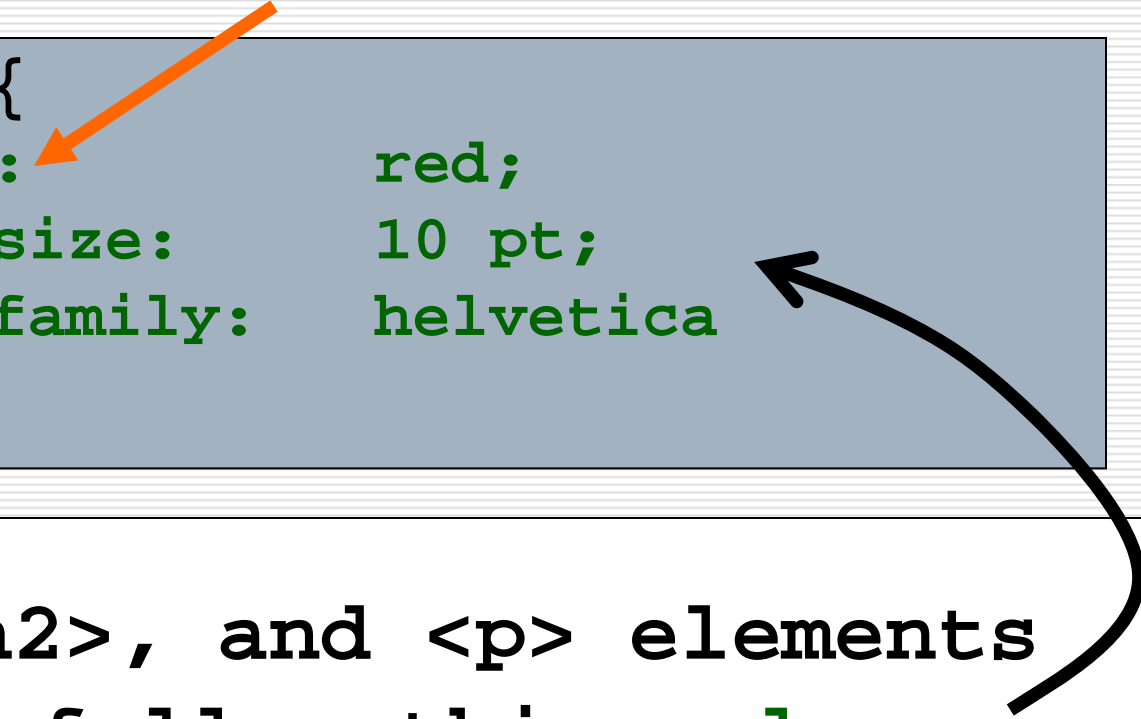
```
h1 {  
  color:          black;  
  font-size:     12 pt;  
  font-family:   arial  
}
```

Each property takes a value.

```
selector {  
  property:value  
}  
selector {  
  property1:value1;  property2:value2  
}
```

Multiple selectors

```
h1, h2, p {  
  color: red;  
  font-size: 10 pt;  
  font-family: helvetica  
}
```



`<h1>`, `<h2>`, and `<p>` elements
will follow this **rule**.

Inheritance

Stylesheets rules are inherited from “parent” to “child.”

```
b { color:purple }
```

```
<b> All of your web pages will use cascading  
stylesheets for the <i> next </i> assignment.  
</b>
```

All of your web pages will use cascading stylesheets for the *next* assignment.

Do you really want all paragraphs to be in red?

```
h1, h2, p {  
  color:      red;  
  font-size:  10 pt;  
  font-family: helvetica  
}
```

- ❑ How does one assign more than one style to the same element?
- ❑ What if you need some paragraphs to be formatted in red and others in black?



class

Classes

```
p.first { color: green }  
p.second { color: purple }  
p.third { color: gray }
```

```
<p class="first"> The first paragraph, with a class name  
of "first." </p>  
<p class="second">The second paragraph, with a class name  
of "second." </p>  
<p class="third">The third paragraph, with a class name  
of "third." </p>
```

```
The first paragraph, with a class name of "first."  
The second paragraph, with a class name of "second."  
The third paragraph, with a class name of "third."
```

Comment your code...

XHTML Comments

```
<!-- This is a XHTML comment -->
```

```
<!-- Comments can also  
span multiple lines -->
```

CSS Comments

```
/* This is a CSS comment */
```

```
/* Comments can also  
span multiple lines */
```

Built-in cascading order of rules

Order of importance:

1. Inline styles
 2. Embedded styles
 3. Linked styles
 4. Imported styles
 5. Default browser styles
-

The bad news about browsers....

- ❑ Cascading stylesheets are great.
- ❑ Web browsers are not so great.

Internet Explorer 3.0 was the first browser to try to support stylesheets. As a result, IE 3 supports most of the CSS properties

IE-4 and Netscape Communicator 4:
Each product had their own interpretations of some of the CSS properties, and other properties were not supported at all.

It is getting better: IE 5 and up will support most of CSS.