



WEBD 162

Class 06

Chapter 11 / 12 Introduction to Cascading Style Sheets (CSS)

1

Cascading Style Sheets (CSS)

- ▶ W3C standard
- ▶ Define the presentation of HTML document
 - ▶ Presentation – how it is displayed
- ▶ **Separate language with its own syntax**



▶ 2

Cascading Style Sheets (CSS)

- ▶ CSS level 1 - 1996
- ▶ CSS level 2 – 1998
- ▶ CSS level 2.1 – 2011
- ▶ **CSS level 3 – implemented / not officially standard**
- ▶ CSS level 4 – on drawing board

▶ 3

Cascading Style Sheets (CSS)

- ▶ Layout control
- ▶ Text formatting
- ▶ **Accessible sites**
 - ▶ **Presentation is separate from content**
- ▶ Tailored for other devices



▶ 4

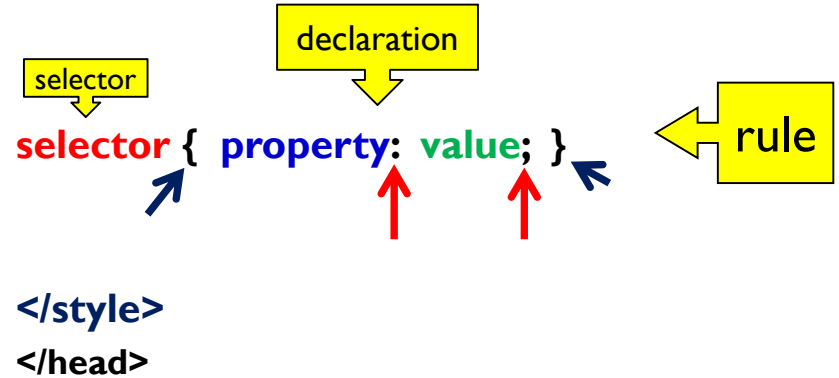
HTML Document Layers

- ▶ HTML – defines structure and markup
- ▶ **CSS – defines presentation**
- ▶ JavaScript – defines behavior

▶ 5

CSS Rules – pg 211

```
<head>  
<style>
```



The diagram illustrates the syntax of a CSS rule. It shows the following code snippet: `selector { property: value; }`. Annotations include: a yellow box labeled 'selector' with an arrow pointing to the 'selector' text; a yellow box labeled 'declaration' with an arrow pointing to the 'property: value;' part; a yellow box labeled 'rule' with an arrow pointing to the entire '{ ... }' block; and a speech bubble labeled 'syntax' in the top right. Red arrows point to the colon and semicolon, and blue arrows point to the opening and closing curly braces.

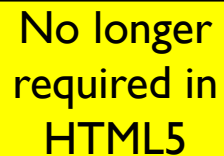
```
selector { property: value; }
```

```
</style>  
</head>
```

▶ 6

CSS Rules – pg 211

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



No longer required in HTML5


```
selector { property: value; }
```

```
</style>  
</head>
```

▶ 7

CSS Rules – pg 211

```
<head>  
<style>
```



Use whitespace for readability

```
selector {  
    property : value;  
    property:value; property:value;  
    property:value;  
}  
  
</style>  
</head>
```

▶ 8

selector { **property**: **value**; }

Mastering selectors

- that is choosing the best type of selector and using it strategically
- **is an important step in becoming a CSS Jedi Master**

selector { **property**: **value**; }

- ▶ **Selector** – any HTML element (for now)
- ▶ **Property** – a defined CSS property
- ▶ **Value** – a defined choice of CSS value

CSS Example Using Element Selector

<head>

<style>

Element selector

h1 { **color**: **green**; }

p { **font-size**: **small**;
 font-family: **sans-serif**;
 }

</style>

</head>

Where can the CSS rules be placed?

- ▶ **Inline**
 - ▶ <p style="color: red">content here</p>
- ▶ **Embedded**
 - ▶ <style> </style> - head section
- ▶ **External file**
 - ▶ <link rel="stylesheet" href="styles.css"> - **ch 13**

CSS Comments – pg 213

HTML comments

`<!-- <p>content here</p> -->`

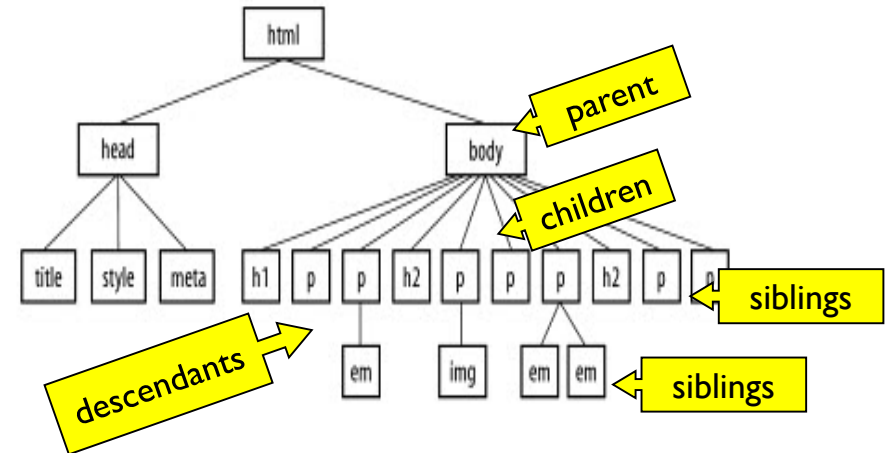
CSS comments

`/* h2 { color: red; } */`

`/*
h2 { color: red; }
*/`

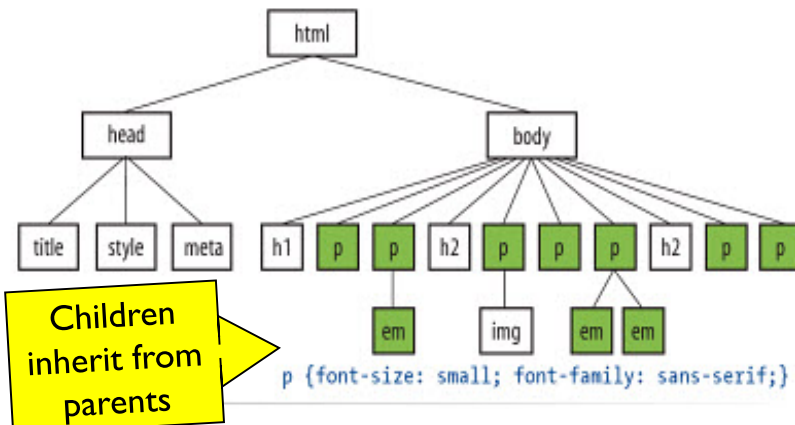
▶ 13

Document Structure– pg 215



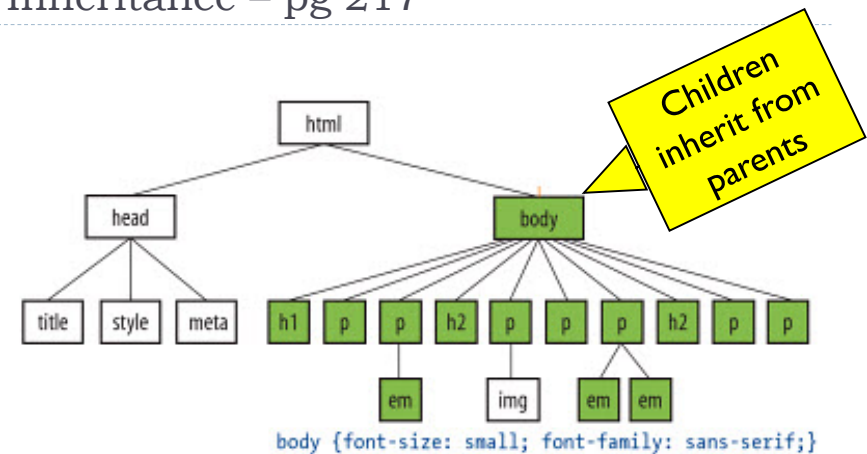
▶ 14

Inheritance – pg 216



▶ 15

Inheritance – pg 217



Not all styles are inherited (margin, padding, border)

▶ 16

The Cascade – pg 219

Order in which styles are applied
in the event of a **CONFLICT** 

Style Sheet Hierarchy

Style information can come from various sources, listed here from general to specific. Items lower in the list will override items above them:

- Browser default settings
- User style settings (set in a browser as a "reader style sheet")
- Linked external style sheet (added with the `link` element)
- Imported style sheets (added with the `@import` function)
- Embedded style sheets (added with the `style` element)
- Inline style information (added with the `style` attribute in an opening tag)
- Any style rule marked **!important** by the author
- Any style rule marked **!important** by the reader (user)

Closest rule
applies

▶ 17

The Cascade – pg 219

What color is the text in the `<p>`?

```
<style>
  p { color: red; }
  p { color: green; }
  p { color: blue; }
</style>
```



```
<body>
<p> some content here </p>
```

▶ 18

The Cascade – pg 219

What color is the text in the `<p>`?

```
<style>
  p { color: red; }
  p { color: green; }
  p { color: blue; }
</style>
```



```
<body>
<p style="color: red"> some content here </p>
```

▶ 19

The Cascade – pg 219

```
<style>
  p { color: green !important; }
  p { font-family: serif; }
</style>
```

```
<body>
<p style="color: red">
  some content here
</p>
```

!important takes
precedence over all

What color is the text in the `<p>`?



▶ 20

The Cascade – pg 219

```
<style>
  p { color: red; }
  p { color: green; }
  p { font-family: serif; }
</style>
```

Closest rule applies in conflict, but other style rules still apply

```
<body>
<p> some content here </p>
```

▶ 21

The Box Model – pg 220 – will cover in future



Figure 11-9. Rules around all the elements reveal their element boxes.

▶ 22

Font Family – Chapter 12 – pg 226

- ▶ Specifies the type face
- ▶ **font-family**

```
<style>
  body { font-family: Arial; }
  body { font-family: Arial,
  ↓
  "Duru Sans",
  ↓
  sans-serif; }
  Generic font family
  ↓
  sans-serif; }
</style>

<body>
<p> some content here </p>
```

▶ 23

Generic font families– pg 229

serif

Examples: Times, Times New Roman, Georgia

Serif typefaces have decorative slab-like appendages (serifs) on the ends of certain letter strokes.

sans-serif

Examples: Arial, Arial Black, Verdana, Trebuchet MS, Helvetica, Geneva

Sans-serif typefaces have straight letter strokes that do not end in serifs.

monospace

Examples: Courier, Courier New, and Andale Mono

In monospace (also called constant width) typefaces, all characters take up the same amount of space on a line.

cursive

Examples: Apple Chancery, Zapf-Chancery, and Comic Sans

Cursive fonts emulate a script or handwritten appearance.

fantasy

Examples: Impact, Western, or other decorative font

Fantasy fonts are purely decorative and would be appropriate for headlines and other display type.

▶ 24

@font-face rule – pg 228

- ▶ To import your own fonts – **download time for user**



```
@font-face {
  font-family: 'Font_name';
  src: url('myfont-webfont.eot?#iefix')
    format('embedded-opentype'),
    url('myfont-webfont.woff') format('woff'),
    url('myfont-webfont.ttf') format('truetype'),
    url('myfont-webfont.svg#svgFontName')
    format('svg');
}
```

Then you just refer to **the** established font name in your font rules, like so:

```
p {font-family: Font_name; }
```

▶ 25

Google Hosted Web fonts – pg 229

- ▶ Alternative to hosted fonts – **little download time**
- ▶ <http://www.google.com/fonts/>



```
<head>
<title>Black Goose Bistro</title>
<link href="http://fonts.googleapis.com/
css?family=Marko+One" rel="stylesheet">
</head>
```

← **<link> element**

```
<style>
  body {font-family: Verdana, sans-serif;}
  h1 {font-family: 'Marko One', Georgia,
  serif;}
</style>
```

▶ 26

Font Size – pg 233

- ▶ Specifies the size of the font in exact or relative measurements
- ▶ **font-size**

```
<style>
```

```
  body { font-size: 100%; }
```

```
  h1 { font-size: 150%; }
```

```
  h2 { font-size: 1.25em; } /* not 1.25 em */
```

Body set to 100%
for consistency – pg 235



```
</style>
```

```
<body>
```

```
<h1> some content here </h1>
```

```
<h2> some content here </h2>
```

▶ 27

CSS Units of Measure– pg 234

Relative units

Relative units are based on **the** size of something else, such as **the** default text size or **the** size of the parent element.

px pixel, considered relative in CSS2.1 because it varies with display resolution.

em a unit of measurement equal to **the** current font size

ex x-height, approximately **the** height of a lowercase "x" in **the** font

The following units are new in CSS3. Browser support may take a while to ramp up.

rem root em, equal to **the** em size of **the** root element (**html**)

ch zero width, equal to **the** width of a zero (0) in **the** current font and size

vw viewport width unit, equal to 1/100 of **the** current viewport (browser window) width

vh viewport height unit, equal to 1/100 of **the** current viewport height

vm viewport minimum unit, equal to **the** value of **vw** or **vh**, whichever is smaller

← **em = standard**

Absolute units

Absolute units have predefined meanings or real-world equivalents.

px pixel, defined as an absolute measurement equal to 1/96 of an inch in CSS3.

pt points (1/72 inch in CSS2.1)

pc picas (1 pica = 12 points)

mm millimeters

cm centimeters

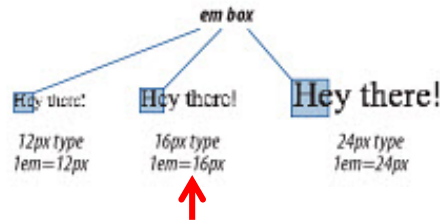
in inches

← **pt, px - used**

▶ 28

em Measurement– pg 235

- ▶ Relative unit based on width of letter M for that font family
- ▶ Default size is **16px (100%)**



```
<style>
  body { font-size: 100%; }
  h1 { font-size: 1.5em; } /* 1.5 x 16 = 24 */
</style>
<body>
<h1> some content here </h1>
```

▶ 29

em and percent calculations– pg 236

- ▶ **Formula:** target_size / size_of_content = result

```
<style>
  body { font-size: 100%; }
  article { font-size: 0.875em; } /* 14 / 16 = .875 */
</style>
```

```
<body>
<article> some content here </article>
```

▶ 30

Font Weight – pg 238

- ▶ Specifies the boldness of the font
- ▶ **font-weight**

| | |
|-------------|--|
| Values: | normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit |
| Default: | normal |
| Applies to: | all elements |
| Inherits: | yes |

```
<style>
  p { font-weight: bold; }
</style>
```

```
<body>
<p> some content here </p>
```

▶ 31

Font Style – pg 240

- ▶ Specifies if a font is italic
- ▶ **font-style**

| | |
|-------------|-------------------------------------|
| Values: | normal italic oblique inherit |
| Default: | normal |
| Applies to: | all elements |
| Inherits: | yes |

```
<style>
  p { font-style: italic; }
</style>
```

```
<body>
<p> some content here </p>
```

▶ 32

Font Variant – pg 241

- Specifies if a font is small caps

- **font-variant**

| | |
|-------------|-------------------------------|
| Values: | normal small-caps inherit |
| Default: | normal |
| Applies to: | all elements |
| Inherits: | yes |

```
<style>
  h2 { font-variant: small-caps; }
</style>
```

```
<body>
<h2> some content here </h2>
```

Font Shorthand / Shortcut property– pg 242

- A shortcut of values that must go in a specific order

- **font**

| | |
|-------------|--|
| Values: | font-style font-weight font-variant font-size/line-height font-family inherit |
| Default: | depends on default value for each property listed |
| Applies to: | all elements |
| Inherits: | yes |

```
<style>
  h2 { bold 1.5em sans-serif; }
</style>
```

Color property– pg 242 – will cover in ch 13

- Sets the color of the text

- **color**

```
<style>
  h2 { color: red; }
  h2 { color: #ff0000; }
  h2 { color: rgb(255,0,0); }
</style>
```

```
<body>
<h2> some content here </h2>
```