

EC201 – WEB TECHNOLOGIES

Cascading Style Sheets (CSS)

What is CSS?

- A set of style rules that tell the web browser how to present a web page or document.
- In earlier versions of HTML, style characteristics, such as fonts, backgrounds, and other aspects of the look of the webpage were controlled from within the page itself.
- The evolution of cascading style sheets allowed for the rules of formatting and presentation to be done in a different manner than before.

Top 10 Reasons for Using CSS

1. Build from the ground up to replace traditional Web design methods
 - CSS to replace HTML tables, font tags, frames, and other presentational hacks of HTML elements
2. Faster download times
 - Reduction of file size is about 50% less than a Web page built with traditional Web design methods.
3. Shorter development time
 - Easily tweak the design of a thousand-page site with just a few edits of one CSS file
4. It's easy to write
 - You can create and edit CSS as easily as you can hand code HTML

Top 10 Reasons for Using CSS

5. Greater control over the typography in a Web page
 - CSS's ability to control typography better
6. Improvements in accessibility
 - Using CSS effectively means structuring content with HTML elements e.g. <P>, <Tables>, <H1> etc.
7. Print designs as well as Web page designs
 - Create presentations not only for Web browsers, but also for other media like print or PowerPoint-like presentations.
8. Better control over the placement of elements in Web page

Top 10 Reasons for Using CSS

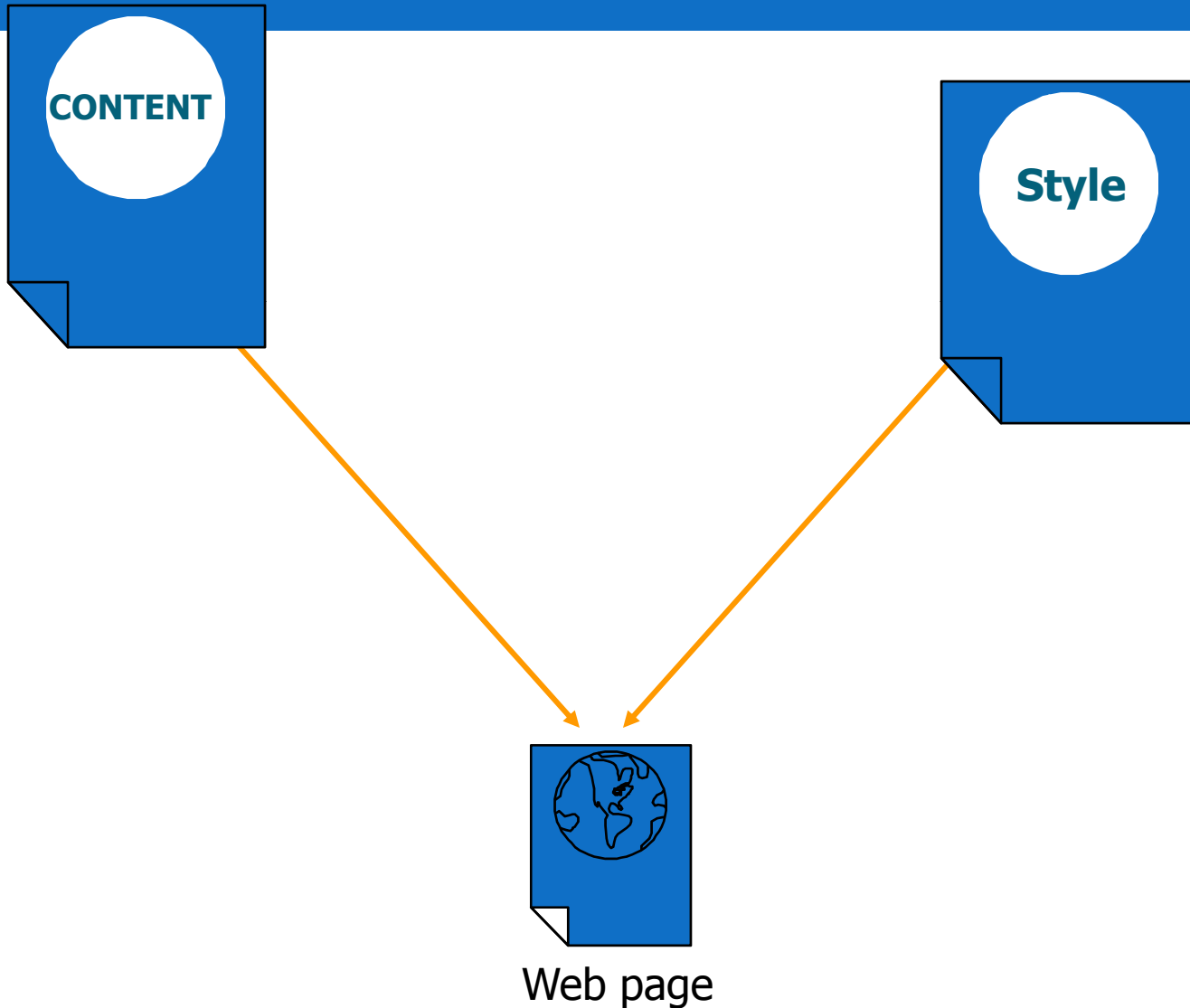
9. The design of Web pages is separated from the content

- By keeping the design in separate files linked to the HTML pages, you reduce the likelihood of your page designs falling apart over time as different contributors add to the Web page

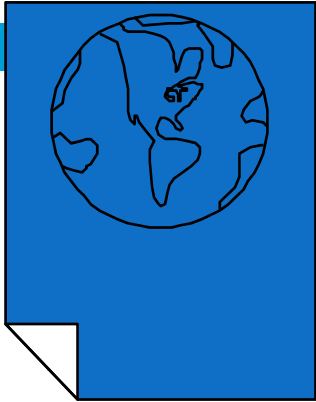
10. Better search engine rankings

- With only HTML used for structuring content in a Web document instead of rigging tags for design, search engines will spider your Web page effectively, and more likely give you a higher page ranking.

What is CSS?



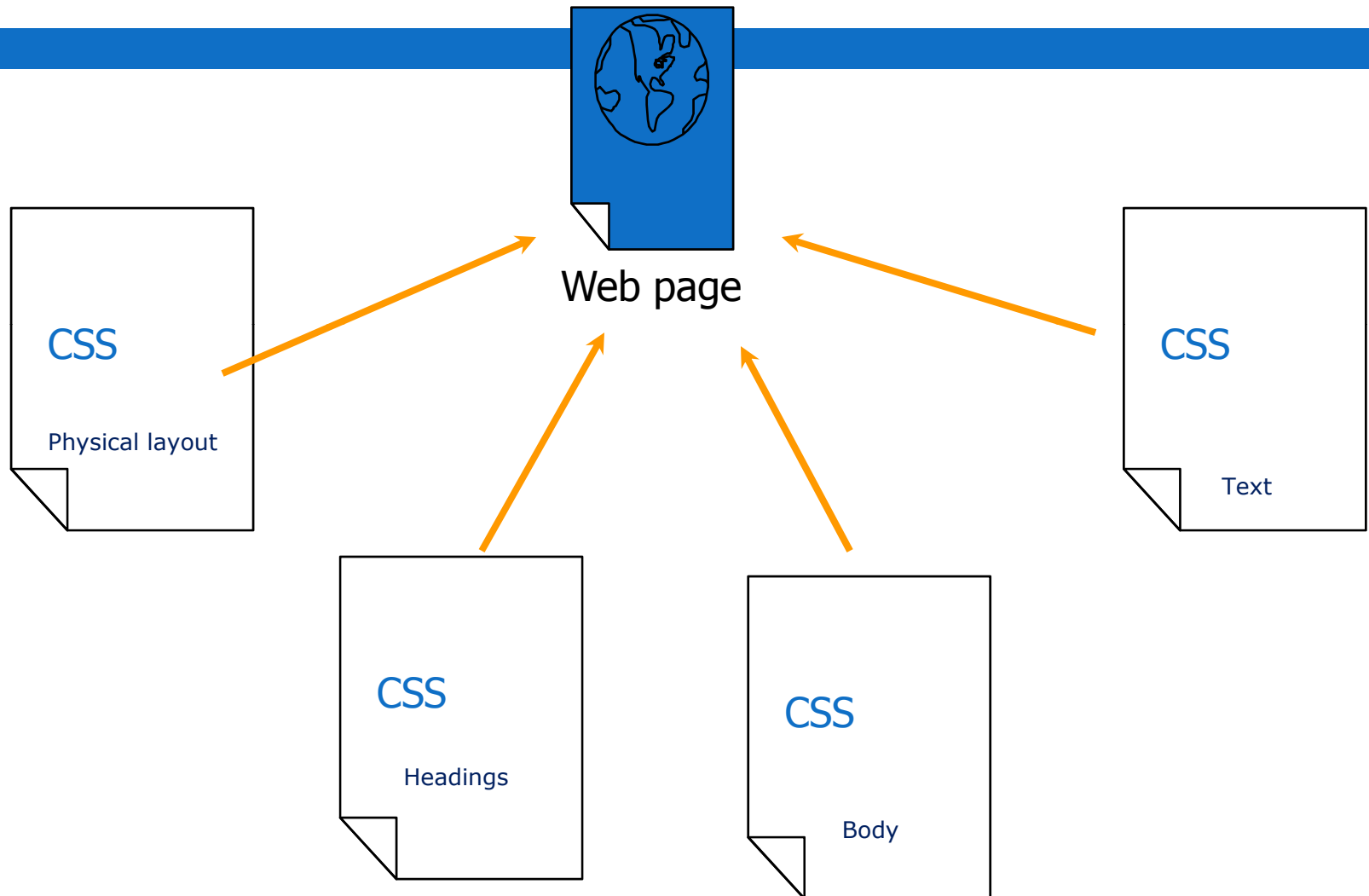
What is CSS?



Web page

Essentially, what we are doing when we use cascading style sheets, it is separating the content of the page from the style or formatting of the page. This has advantages, in that once the style sheet is loaded locally, all other pages that are based on that style sheet will load quicker. The reason being that the stylistic markup for defining the appearance of the page is loaded separately from the content of the page.

What is CSS?



What is CSS?

- ❑ One cascading style sheet can be used to set the style and format for many different web pages.
- ❑ The biggest advantage is that if the web author wants to make whole changes to a website, the style editing takes place in one location (the CSS), yet is applied to all related locations (the pages).
- ❑ Imagine a large site with hundreds of pages. To change even one item on each page, without a CSS, would require editing of each of the hundreds of pages.
- ❑ In contrast, a CSS based website would require the editing of only one file – the CSS.

Types of CSS

- ❑ External
- ❑ Internal/Embedded
- ❑ Inline

External

- Allows for using style sheets from other sources
- Connection made via the LINK tag
- Use the optional TYPE attribute to specify a media type
- Two Methods

Linking a CSS file: (Recommended)

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

Importing a CSS file: (Not supported in older browsers)

```
<head>  
  <style type="text/css">@import url(path/to/cssdoc.css);</style>  
</head>
```

Internal/Embedded

- Style characteristics are embedded in the HEAD section of the webpage
- Perhaps best used when a single page requires a unique style sheet

```
<head>  
<style type="text/css">  
hr { color: navy}  
body {margin-left: 20px}  
</style>  
</head>
```

Inline

- ❑ Least flexible
- ❑ Requires each element to be tagged if you want them to appear differently
- ❑ Loses the advantage of using CSS

```
<p style="color: yellow; font-family: verdana">
```

This is a paragraph

```
</p>
```

Using multiple sheets

- You can use multiple sheets to define the style of your document
- Internal styles will override external styles, if they are duplicated

Using multiple sheets

h3 {color: red; text-align: right; font-size: 8pt} *(external CSS)*

h3 {text-align: center; font-size: 20pt} *(internal CSS)*

will yield

Using multiple sheets

h3 {color: red; text-align: right; font-size: 8pt} *(external CSS)*

h3 {text-align: center; font-size: 20pt} *(internal CSS)*

will yield

h3 {color: red; text-align: center; font-size: 20pt }

Understanding the Cascade

- Cascading
 - ▣ Determining rule weight by order
 - Based on order of rule within style sheet
 - Those listed later take precedence over those listed earlier in the style sheet

Understanding the Cascade

□ Inheritance

- Based on hierarchical structure of documents
- CSS rules inherit from parent elements to child elements:
 - thus elements will inherit style rules from elements unless a style rule is specifically set for the element

Basic CSS Syntax

- Three parts:
 - ▣ selector
 - ▣ property
 - ▣ value

selector {property: value} } *declaration*

Basic CSS Syntax

selector {property: value}

selector: the basic HTML element tag you wish to define

property: the attribute of the selector that you wish to change

value: the particular markup value for that attribute

body {color : black}

Basic CSS Syntax

If the **value** has multiple words, put the value in quotes

```
p {font-family: "sans serif"}
```



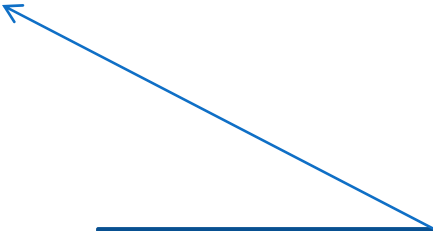
Paragraph element of HTML

Apply this style to all p element
used in this document

Basic CSS Syntax

You can specify multiple **properties** to a single selector.
Properties must be separated by a **semicolon**.

```
p { text-align: left; color: red }
```



Use semicolon for multiple
properties

Basic CSS Syntax

To make properties more readable, put each on a separate line.

```
p { text-align: center;  
    color: navy;  
    font-family: arial  
}
```

Basic CSS Syntax – Grouping Selectors

Selectors can be grouped so that a common property can be specified

p, h1, h2, h3, h4

```
{  
    font-family: Arial;  
    color: yellow  
}
```

`<h1> This is a level 1 heading </h1>`

`<h2> This is a level 2 heading </h2>`

Basic CSS Syntax – Descendants

Selectors can be descendants

p b { color: yellow }



Only those **** element would be yellow
that are within a **<p>** element

<p> This would be yellow </p>

This would not be yellow


<p>Not this either</p>

CSS Syntax – class selector

The **class** selector allows you to create different styles for the same HTML element.

```
p.center { text-align: center; color: green }
```

```
p.right { text-align: right; color: red }
```



This style will be applied to only those **<p>** element that has a value of class attribute equal to **“right”**

CSS Syntax - class

```
p.right { text-align: right }
```

```
<p class="right">
```

This paragraph will be right aligned.

```
</p>
```

***Note:** the class name must be in quotes inside the opening tag*

CSS Syntax - class

This is **improper use** of the class **selector**:

```
<p class="right" class="center">
```

This paragraph will be right aligned.

```
</p>
```

Only one class selector can be included inside the tag

CSS Syntax – class free of tag name

You can also create a **class selector free of a tag name** if you want all tags that have a particular class to be formatted the same. E.g.

```
.center { text-align: center; font-style: italic }
```

Any tag with a “center” class will be aligned center and italic

CSS Syntax – class free of tag name example

```
.center { text-align: center }
```

```
<h1 class="center">
```

This heading will be centered

```
</h1>
```

```
<p class="center">
```

So will this text

```
</p>
```

CSS Syntax – id Selector

While the class selector can apply to several different elements, the **id** selector can only apply to one, unique element with in a document.

```
p#para1 { text-align: center; color: green }
```



This style will apply to only that **<p>** element that has id value equal to **para1**

CSS Syntax – id Selector

```
p#para1 { text-align: center; color: green }
```

```
<p id="para1">
```

This text would be centered and green

```
</p>
```


CSS Syntax - comment

You can insert **comments** to help you describe the particular style

Comments open with **/*** and are closed with ***/**

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

```
P { color: red;
```

```
/* This is another comment */
```

```
Font-family: verdana }
```

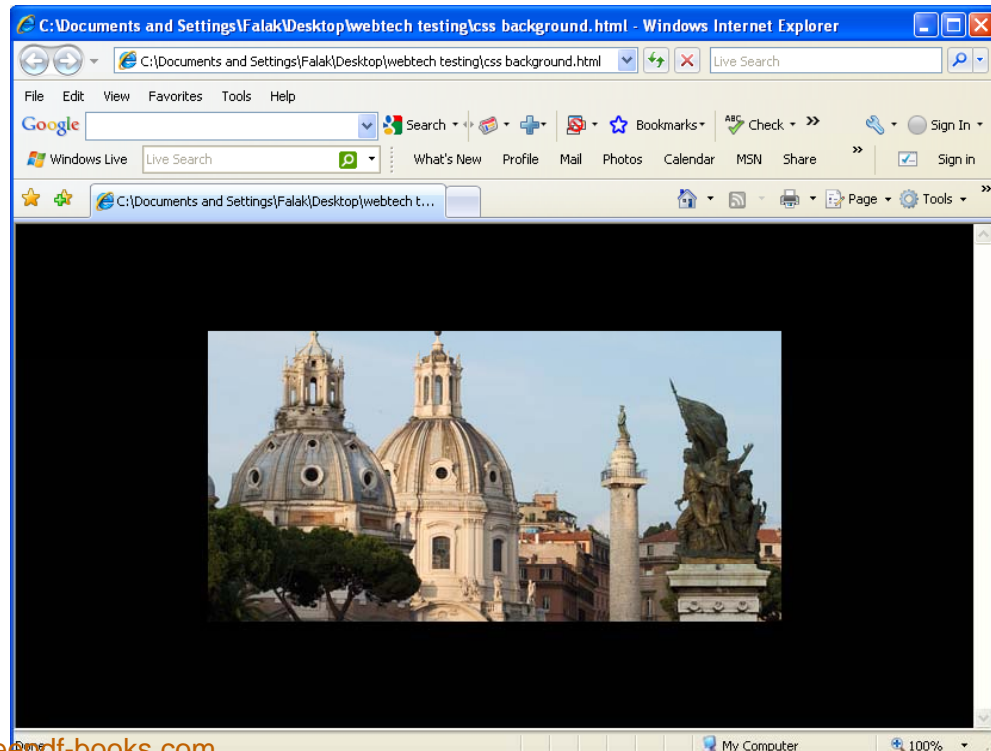
CSS Properties

34

- Basic CSS Properties
 - ▣ Background Properties
 - ▣ Text Properties
 - ▣ Font Properties
 - ▣ Border Properties
 - ▣ Margin Properties

Background properties

- Define the background effects of an element
- Effects include color, using an image for a background, repeating an image and positioning an image



Background properties

- Basic syntax
 - **background**
 - background-color
 - background-image
 - background-repeat
 - background-attachment
 - background-position

Background properties

- All attributes can be set in a single declaration:

Shorthand Property:

`background: #000000 url('psumark.gif') no-repeat
fixed center`

Equivalent to:

`background-color: #000000; background-image:
url('psumark.gif'); background-repeat: no-repeat;
background-attachment: fixed; background-
position: center`

Background properties

- Setting the body background (internal CSS)

<style>

```
body { background: #000000 url('psumark.gif') no-  
      repeat fixed center }
```

</style>

Background properties

- Elements can also be set separately (internal CSS)

<style>

```
body { background-color: #000000; background-  
image: url('psumark.gif'); background-repeat: no-  
repeat; background-attachment: fixed;  
background-position: center }
```

</style>

CSS Properties

40

- Basic CSS Properties
 - ▣ Background Properties
 - ▣ **Text Properties**
 - ▣ Font Properties
 - ▣ Border Properties
 - ▣ Margin Properties

Here we can see how the background for the body element would be specified in an internal style sheet. Here we can see how the background for the body element would be specified in an internal style sheet. Here we can see how the background for the body element would be specified in an internal style sheet. Here we can see how the background for the body element would be specified in an internal style sheet. Here we can see how the background for the body element would be specified in an internal style sheet. Here we can see how the background for the body element would be specified in an internal style sheet. Here we can see how the background for the body element would be specified in an internal style sheet.



Text properties

- Controls the appearance of text in the web page
- Commonly used attributes
 - ▣ **color**
 - ▣ **direction**
 - ▣ **text-align**
 - ▣ **text-decoration**
 - ▣ **text-indent**

Text properties

- **color**
 - ▣ sets the color of the text
 - ▣ color can be represented by the color **name** (red), an **rgb** value (rgb(255,255,255)), or by a **hexadecimal** number (#ffffff)
- Syntax
 - ▣ body {**color**: #ffffff}

Text properties

- **direction**

- sets the direction of the text
- can be set as left to right (**ltr**) or right to left (**rtl**)

- Syntax

- body {**direction**: ltr}

Text properties

- **text-align**

- aligns the text in an element
- possible values are **left**, **right**, **center** and **justify**

- Syntax

- p {**text-align**: justify}

Text properties

□ **text-decoration**

- adds certain decoration elements to the text
- possible values are **none**, **underline**, **overline**, **line-through** and **blink**

□ Syntax

- p {**text-decoration**: underline}

Text properties

- **text-indent**

- indents the first line of text inside an element
- possible values are **length** (defines a fixed value) and %

- Syntax

- p {**text-indent**: 20px} or
- p {**text-indent**: 5%}

CSS Properties

47

- Basic CSS Properties
 - ▣ Background Properties
 - ▣ Text Properties
 - ▣ **Font Properties**
 - ▣ Border Properties
 - ▣ Margin Properties

HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET.



Font properties

- Define the look of the font in text areas
- One of the broader sets of properties in CSS
 - **font**
 - font-style
 - font-variant
 - font-weight
 - font-size/line-height
 - font-family

Font properties

49

- All attributes can be set in a single declaration:

Shorthand Property:

font: italic small-caps 900 12px arial

Equivalent to:

font-style: italic; **font-variant:** small-caps; **font-weight:** 900; **font-size:** fixed; **font-family:** arial

Font properties

- **font-style**

- ▣ Normal

- ▣ Italic

- ▣ oblique

- **Syntax**

- ▣ body {**font-style**: italic}

Font properties

□ **font-variant**

▣ Normal

- font displays as is

▣ small-caps

- font displays in all capitals, with lower case letters in smaller size

Syntax:

body {**font-variant**: small-caps}

Font properties

□ font-weight

- ▣ normal
- ▣ bold
- ▣ bolder
- ▣ lighter
- ▣ weighted values

range from 100 – 900

400 is the same as normal weight

700 is the same as bold weight

Syntax:

body {**font-weight**: bold}

Font properties

□ font-size

- xx-small
- x-small
- Small
- Medium
- Large
- x-large
- xx-large
- %

Syntax:

```
body {font-size: 20px}  
      {font-size: x-large}  
      {font-size: 125%}
```

Font properties

- **font-family**
 - ▣ family-name
 - “times”, “arial”, “courier”, “verdana”
 - ▣ generic-family
 - “serif”, “sans-serif”, “monospace”

Syntax:

body {font-family: verdana, sans-serif}

CSS Properties

55

- Basic CSS Properties
 - ▣ Background Properties
 - ▣ Text Properties
 - ▣ Font Properties
 - ▣ **Border Properties**
 - ▣ Margin Properties



Border properties

- Allows you to specify the width, style, and color of an element's border
- Many different properties can be applied

- **Border**

border-color

border-style

border-width

border-bottom

border-bottom-color

border-bottom-style

border-bottom-width

border-left

border-left-color

border-left-style

border-left-width

border-right

border-right-color

border-right-style

border-right-width

border-top

border-top-color

border-top-style

border-top-width

Border properties

- You can specify the **width**, **style**, **color** and on which sides the border appears

Shorthand Property:

border: medium double rgb(250,0,255)

Equivalent to:

border-width: medium; **border-style:** double; **border-color:** rgb(255,0,255)

CSS Properties

58

- Basic CSS Properties
 - ▣ Background Properties
 - ▣ Text Properties
 - ▣ Font Properties
 - ▣ Border Properties
 - ▣ **Margin Properties**

HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET. HERE WE CAN SEE HOW THE BACKGROUND FOR THE BODY ELEMENT WOULD BE SPECIFIED IN AN INTERNAL STYLE SHEET.



Margin properties

- ❑ Define the space around elements
- ❑ You can use negative values to overlap content
- ❑ Margins can be set independently or collectively
- ❑ Can be set to auto, a fixed length or a % of the total height of the document

Margin properties

- Properties
 - ▣ margin
 - ▣ margin-top
 - ▣ margin-right
 - ▣ margin-bottom
 - ▣ margin-left

Margin properties

- Can be set in **one declaration**
- Think clock face (clockwise)
 - ▣ **top, right, bottom, left**

```
h1 {margin: 10px 20px 30px 40px}
```

Margin properties

- All margins can be set the same

```
h1 {margin: 40px}
```

Margin properties

- Margin settings can be paired (left and right, top and bottom)

```
h1 {margin: 40px 5%}
```

In this example, the top and bottom margins would be 40 pixels, While the left and right margins would be 5% of the total height of the document.

Margin properties

- 0 size margins do not need to be specified. 0px, 0pt and 0 are all equivalent.

```
h1 {margin: 40px 0 5% 0}
```

In this example, the top margin would be 40 pixels, the left and right margins would be 0, and the bottom margin would be 5% of the total height of the document.

Summary

- ❑ CSS allows much richer control of document appearance (than HTML)
- ❑ CSS simplifies HTML
- ❑ CSS makes maintenance easier
- ❑ CSS allows for “cascading” rules
- ❑ CSS can help reduce download time
- ❑ CSS is the future of presentation on the web
- ❑ Browser support is spotty, but getting better

Thank You

- Thank you