

CSS

WEB DEVELOPMENT FUNDAMENTALS



Objectives

- Cascading style sheets
 - Embedded and inline style sheets
 - Classes and IDs
 - Linked stylesheets
 - Inheritance
 - CSS3
 - Font Families

Inline Styles

- STYLE attribute can be used on any tag
- Affects that tag only

```
<p style="margin-left: 1in; margin-right: 1in;
line-height:200%">
This text will be shown with one-inch left and
right margins, and double-spaced.
</p>
<p>
This text is formatted as normal for &lt;P>
tags.
</p>
```

3

Embedded Style Sheets

- Use <style> .. </style> before <body> tag
- A style sheet definition contains a list of
 - HTML tags, and
 - Associated format information for that tag

```
.. .. ..
<style type="text/css">
h1 { font-size: 15pt; font-weight:bold}
p { font: bold italic 12pt/20pt times, serif}
</style>
.. .. ..
```

4

Using Classes

- A Class is a set of CSS styling options, which are reusable
- An HTML tag can be given a class
 - The special formatting is then used for that instance
- A class can be defined with no particular element

```
<style type="text/css">
    p.redcaps {color:red; font-style: small-caps}
    .bluearial {font-family: arial; color: blue}
</style>
.....
<h1 class="blueArial">This heading is in blue Arial</h1>
<p class="redcaps">This text is in red and all small caps</p>
<p class="blueArial">This text is in blue Arial, like the
heading.</p>
```

5

ID attribute

- ID attribute is used to name a tag
- Style Sheets use IDs as selectors for style rules
 - A rule is applied to an element

```
<h1 id="FirstHeading">Welcome to Big Company!</h1>
```

```
<head>
<style type="text/css">
    #FirstHeading {background: yellow; font-weight: bold; color: red}
</style>
</head>
<body>
<h1 id="FirstHeading">Welcome to Big Company!</h1>
</body>
```

6

Linking to a Style Sheet

- A style sheet may be on a separate URL
- Allows a global style sheet for all pages in a group
- Good for corporate styling

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

7

Element sizing

- Sizing elements can be achieved in a number of different ways:
 - Pixels (px) - a fixed measurement based on the size of a pixel

```
img { width: 150px; }
```
 - Ems (em) - a relative unit that equates to the font size of the element
 - An em unit is relative to the parent element's font size

```
article{ width: 3em; }
```
 - Points (pts) - Points are an absolute unit equal to 1/72 of an inch
 - Points can be useful when setting type sizes for print

```
body{ font-size: 12pt; }
```

8

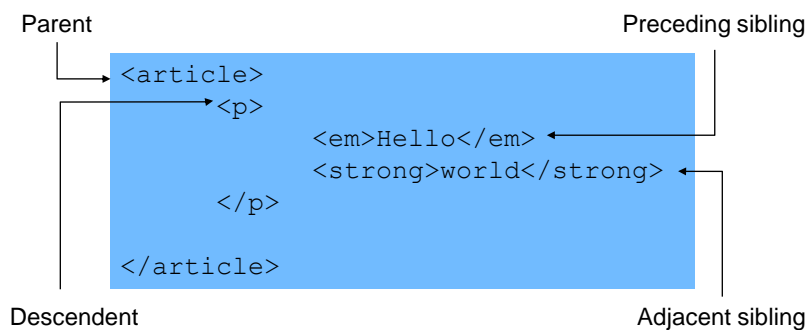
Cascading

- Cascading is the way that styles cascade down
 - Styles from higher levels cascade down to lower levels
 - If the same style appears many times, the latter specified wins
- Cascading implies overriding:
 - Linked stylesheet – overrides the browser default
 - Imported stylesheet – overrides the linked stylesheet
 - Embedded stylesheet – overrides the imported stylesheet
 - Inline style attribute – overrides embedded stylesheet
- The cascade also orders by specificity
 - See later

9

Understanding Inheritance

- HTML tags exist in a hierarchical tree from `<html>` root to text nodes
 - When a tag is surrounded by another tag the tags are nested



10

Hierarchical Inheritance

- Elements inherit from containing parents
 - So we only need to define a style rule at the highest level
 - We can then override rules at descendent levels
- Complex hierarchies can be difficult to manage
 - Chrome's developer tools help greatly
 - Showing which styles are applied
 - Where they come from
 - If rules are being overridden
 - The order in which they applied



11

CSS3 combinators

- Selectors that depend on a specific relationship
- Descendent selector


```
ul li { color: black; }
```
- Child selector


```
ul>li { color: black; }
```
- Adjacent selector


```
h2+p { color: black; }
```
- Sibling selector


```
h2~p { color: black; }
```

12

Attribute Selectors

- CSS3 defines three new attribute selectors
- ^= operator finds attributes starting with a value

```
a[href^="http:"] { color: blue; }
```

- \$= operator finds attributes ending with a value

```
a[src$=".png"] { color: red; }
```

- *= operator finds attributes containing the value

```
div[id*="stuff"] { color: red; }
```

13

CSS3 pseudo-classes

- CSS3 offers a series of new psudeo-element selectors

Selector	Sets the appearance of
:first-child	First instance of selector type if it is the first child
:first-of-type	The first time its selector appears
:nth-child(N)	A specific occurrence of a type
:nth-of-type(N)	The specific occurrence of a selector type
:nth-last-of-type(N)	As above but in reverse
:last-child	Last child of a parent
:last-of-type	Last instance of a specified selector type

14

CSS3 non hierarchical pseudo-classes

- There is a new powerful negation selector

```
:not(S)
p:not(.myClass){
    color: blue;
}
```

- An ability to detect empty nodes

```
:empty
td:empty { background-color: red; }
```

- Style an element that is a target

```
:target
a:target:before {content: url(star.gif); }
```

15

Pseudo classes and elements CSS3

- (Dynamic) pseudo classes, elements...
 - Applying to user actions (pseudo classes)

```
:active { color: blue;}
:hover { color: blue;}
:focus { color: blue;}
:link { color: blue;}
:visited { color: blue;}
```

- Applying to placement (pseudo elements)

```
::after { color: blue;}
::before { color: blue;}
::first-letter { color: blue;}
::first-line { color: blue;}
```

- Selection pseudo element

```
::selection { color: blue;}
```

16

Pseudo classes specificity

- Recall: in the cascade styles are sorted by specificity
 - Latter rules are more specific than earlier rules
- Hence for link pseudo classes to work use this order
 - a
 - :link
 - :visited
 - :hover
 - :active

```
a {color: black;}
a:link {color: blue;}
a:visited {color: red;}
a:hover {color: green;}
a:active {color: orange;}
```

17

Before: and after:

- Used to insert content before or after an element
 - Can be specific content, counters or values of attributes
- Specify style and content of inserted content
 - content: normal | none | <string> | <uri> | <counter> | attr(<identifier>) | open-quote | close-quote | no-open-quote | no-close-quote | inherit

```
p.note:before {font-weight: bold; content: "Note: " }
h1:before {content: "Chapter " counter(chapter) ". ";
           counter-increment: chapter;}
```

18

Working with fonts – setting the character type

- As previously noted, it is important to set the encoding type of a document

- In HTML5

```
<meta charset="utf-8">
```

- In XHTML/HTML4

```
<meta http-equiv="Content- Type"
content="text/html; charset="utf-8" />
```

- A character set is a list of character codes your browser will accept
 - If it does not understand a character a glyph will appear in its place
 - You are also leaving yourself exposed to dangerous JavaScript attacks



19

Font Families

- CSS Defines five font families to which most fonts are categorised

- Serif - Times New Roman
- Sans-serif - Arial
- Monospace - Courier New
- Cursive - Brush Script
- Fantasy - Papyrus

```
font-family: Helvetica, Verdana, Arial, sans-serif
```

- There are also dingbats and other symbol library fonts
- Plus HTML character entities
 - £ for a GBP symbol as an example
- Fonts are set in a comma delimited list
 - Browser checks if font is available, used if present moves on if not

20

Other Font Settings

- Additional typography properties can be set:

Property	Usage
font-size	Font size can use any of the units previously discussed or a value between xx-small and xx-large
font-weight	font-weight controls the normal weight of the font normal bold or a weight scale between 100 and 900
font-style	Normal, italic or oblique – if no oblique is present italic will be used
line-height	The height of each line of text known as leading
vertical-align	Sets the alignment of the text in relation to the line box

21

Setting fonts as a compound rule

- Fonts need to be set in a very specific way using CSS
 - Requiring a minimum set of keywords and a specific order
 - The most basic rule requires:

```
font: <font-size> <font-family>;
```

- When using a complex rule optional values precede the mandatory

```
font: italic small-caps 1.2em Georgia, serif;
```

- With the exception of a sneakily inserted line-height
 - Note the lack of measurement unit
 - You can add them but it can cause issues

```
font: 100%/2.5 Helvetica,
```

22

@font-face

- @font-face is not strictly speaking CSS3
 - It was originally born in CSS 2
 - Although, not appearing in CSS 2.1
 - CSS 3 is attempting to bring it into the standards
 - Their implementation relied on a proprietary Type (.eot)
- No single format is used in every browser
 - .eot, .otf, .ttf, .svg must be provided to support all browsers
 - Useful online resources to assist with this
 - Organise and manage the fonts
- Restrictive policies and older browsers cause issue
 - Test for browser support as usual

23

Adding a drop shadow

- Drop shadow is back as if the 1990's never happened!

```
.shadow {  
    text-shadow: 10px 8px 20px #383499;  
}
```

- text-shadow requires the following properties:
 - X, Y offset
 - Amount of blur
 - Colour (can be a new rgba type)
- Corresponding box shadow rule

```
.shadow {  
    box-shadow: 3px 3px 3px 3px #007;  
}
```

24

Colour - HSL values

- CSS3 supports HSL values as an option
- Used in colour wheels seen in many design IDE
 - Hue - Degree value from 0 to 360 based upon a colour wheel
 - Red is 0°
 - Green is 120°
 - Blue is 240°
 - Saturation - A percentage 0% is no colour 100% full
 - Lightness - A percentage 0% is black and 100% is white

```
ul>li { color: hsl(0, 0, 100%); }
```

25

Colour - rgba and hsla

- Variant on the RGB selector that sets the alpha channel
 - RGB expressed as values between 0 and 255
 - Alpha the last parameter and value between 0 and 1

```
ul>li { color: rgba(255,100,0,0.5); }
```

- There is also an alpha variant of the HSL approach

```
ul>li { color: hsla(0, 0, 100%, 0.5); }
```

26

Rounded borders

- A staple of JavaScript libraries is now achievable via CSS

```
border-radius: 30px
```

- Different radius can be added to different corners

```
border-top-left-radius: 50px;  
border-top-right-radius: 30px;  
border-bottom-right-radius: 50px;  
border-bottom-left-radius: 30px;
```

- Shorthand

```
border-radius: 50px 30px 50px 30px;
```

27

Summary

- Cascading style sheets
 - Embedded and inline style sheets
 - Classes and IDs
 - Linked stylesheets
 - Inheritance
 - CSS3
 - Font Families

28