

CSS INTRO

- **Cascading Style Sheets (CSS)**
- **Made up of selectors and rules that define the visual style of elements**
- **CSS provides the rules to aesthetically style your webpage**
 - Change fonts and the way text looks, colors
 - Width, height, background colors and images
 - Positioning
 - Lines and space between elements
- **Rules cascade**
 - An algorithm defining how to combine properties
 - Ones defined later override or build upon earlier ones
 - More specific ones override earlier ones
 - <https://developer.mozilla.org/en-US/docs/Web/CSS/Cascade>



ADDING CSS TO YOUR PAGE

Three main methods (only teaching 1)

- **External Style Sheet**
 - Text document with a .css extension
 - CSS File is linked to the HTML document in the head section
 - `<link rel="stylesheet" href="style.css" />`
 - Preferred way, separates presentation in another file

ANATOMY OF A CSS RULE

declaration

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

declaration block

```
selector {  
  property1: value1;  
  property2: value2;  
  property3: value3;  
}
```

CSS CONCEPTS

- **Inheritance**
 - Some properties inherit their settings from their parent element
 - Mostly styles that effect text
- **Parents & Children**
 - Elements nested inside other elements are said to be children of that element.
 - The element that a given element is nested inside is its parent.
- **It is very important that you understand the way your HTML is structured and how each of your elements are nested within each other to be successful writing complex CSS rules.**

CSS INHERITANCE

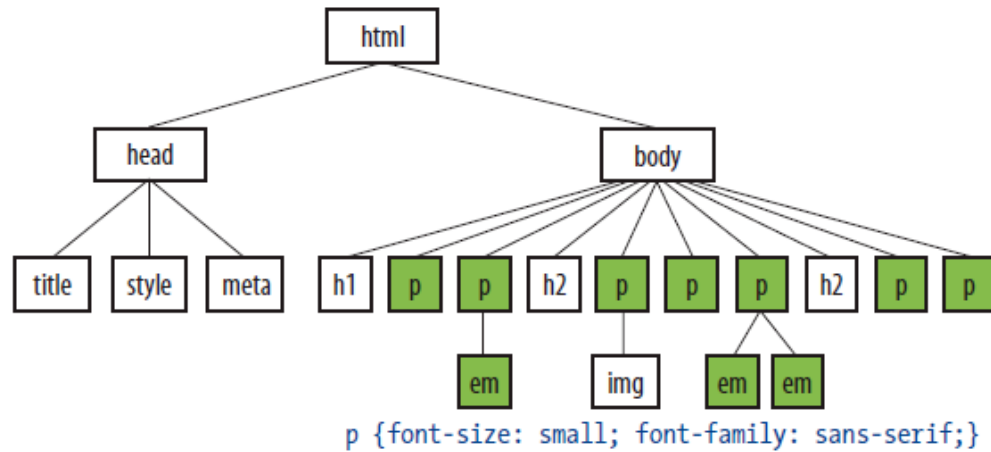


Figure 11-7. Certain properties applied to the `p` element are inherited by their children.

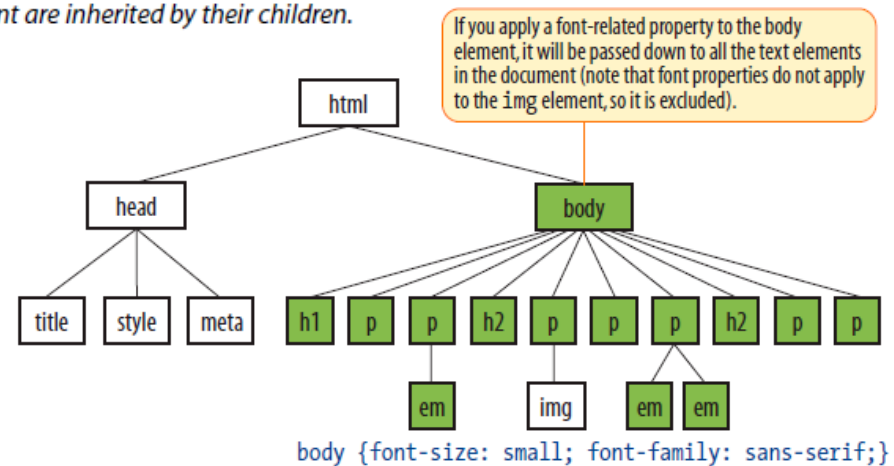


Figure 11-8. All the elements in the document inherit certain properties applied to the body element.

font-family

Values: one or more font or generic font family names, separated by commas | inherit

Default: depends on the browser

Applies to: all elements

Inherits: yes

Use the **font-family** property to specify a font or list of fonts (known as a **font stack**) by name, as shown in these examples.

```
body { font-family: Arial; }  
var { font-family: Courier, monospace; }  
p { font-family: "Duru Sans", Verdana, sans-serif; }
```

font-size

Values: length unit | percentage | xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | inherit

Default: medium

Applies to: all elements

Inherits: yes

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

font-style

Values: normal | italic | oblique | inherit

Default: normal

Applies to: all elements

Inherits: yes

sample of oblique Times

sample of true italic Times

This is an example of oblique Times as rendered in a browser.

This is an example of italic Times as rendered in a browser.

color

Values: *color value (name or numeric) | inherit*
Default: *depends on the browser and user's preferences*
Applies to: *all elements*
Inherits: *yes*

```
h1 { color: gray; }  
h1 { color: #666666; }  
h1 { color: #666; }  
h1 { color: rgb(102,102,102); }
```

			
Black #000000	Gray #808080	Silver #C0C0C0	White #FFFFFF
			
Maroon #800000	Red #FF0000	Purple #800080	Fuchsia #FF00FF
			
Green #008000	Lime #00FF00	Olive #808000	Yellow #FFFF00
			
Navy #000080	Blue #0000FF	Teal #008080	Aqua #00FFFF
			
Orange (CSS 2.1) #FFA500			

Figure 13-1. The 17 standard color names in CSS2.1.

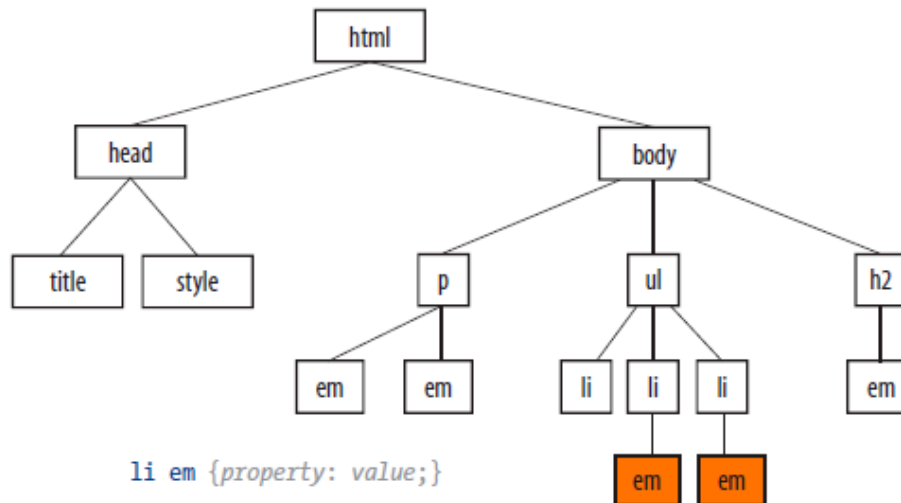
Figure 13-2. The 140 extended color names in CSS3. Bear in mind that these will look quite different on a screen.

Element selector

```
p { color: navy; }
```

Grouped selectors

```
p, ul, td, th { color: navy; }
```



It is also possible to nest descendant selectors several layers deep. This example targets `em` elements that appear in anchors (`a`) in ordered lists (`ol`).

```
ol a em { font-variant: small-caps; }
```

Child selector

A [child selector](#) is similar to a descendant selector, but it targets only the direct children of a given element. There may be no other hierarchical levels in between. They are indicated with the greater-than symbol (>). The following rule affects emphasized text, but only when it is directly contained in a **p** element. An **em** element inside a link (**a**) within the paragraph would not be affected.

```
p > em {font-weight: bold;}
```

Adjacent sibling selector

An [adjacent sibling selector](#) targets an element that comes directly after another element with the same parent. It is indicated with a plus (+) sign. This rule gives special treatment to paragraphs that follow an **h1**. Other paragraphs are unaffected.

```
h1 + p {font-style: italic;}
```

General sibling selectors

NEW IN CSS3

A [general sibling selector](#) selects an element that shares a parent with the specified element and occurs after it in the source order. They do not need to follow one another directly. This type of selector is new in CSS3 and is not supported by Internet Explorer 8 and earlier. The following rule selects any **h2** that both shares a parent element (such as a **section** or **article**) with an **h1** and appears after it in the document.

```
h1 ~ h2 {font-weight: normal;}
```

ITMD-361

CSS SPECIFICITY & BOX MODEL

CSS CASCADE & SPECIFICITY

- **Style passes down (cascades) until a rule with more weight overrides a previous style**
- **First goes by style sheet hierarchy**
- **Then goes by order defined in the style sheets**
- **If there is a conflict it resolves with a point system**
 - Creators developed a point system
 - inline is 1000, id is 100, class is 10, element is 1
- **See Charts on next two slides**
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Cascade>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>
- <http://css-tricks.com/specifcs-on-css-specificity/>
- <http://www.smashingmagazine.com/2007/07/27/css-specificity-things-you-should-know/>
- <http://code.tutsplus.com/tutorials/quick-tip-understanding-css-specificity--net-10963>

CSS SPECIFICITY

CSS Selector Specificity - *Cheat Sheet*

	*	id	classes	elements	result [*]
	a	b	c	d	
li	0	0	0	1	1
.lorem	0	0	10	0	10
#ipsum	0	100	0	0	100
style = " "	1000	0	0	0	1000
ul li	0	0	0	2	2
ul ol + li	0	0	0	3	3
ul ol li.red	0	0	10	3	13
li:first-line	0	0	0	2	2
#ipsum li	0	100	0	1	101
a:link	0	0	10	1	11

Legend:

a=1000

b=100

c=10

d=1

style = " "

ID attributes

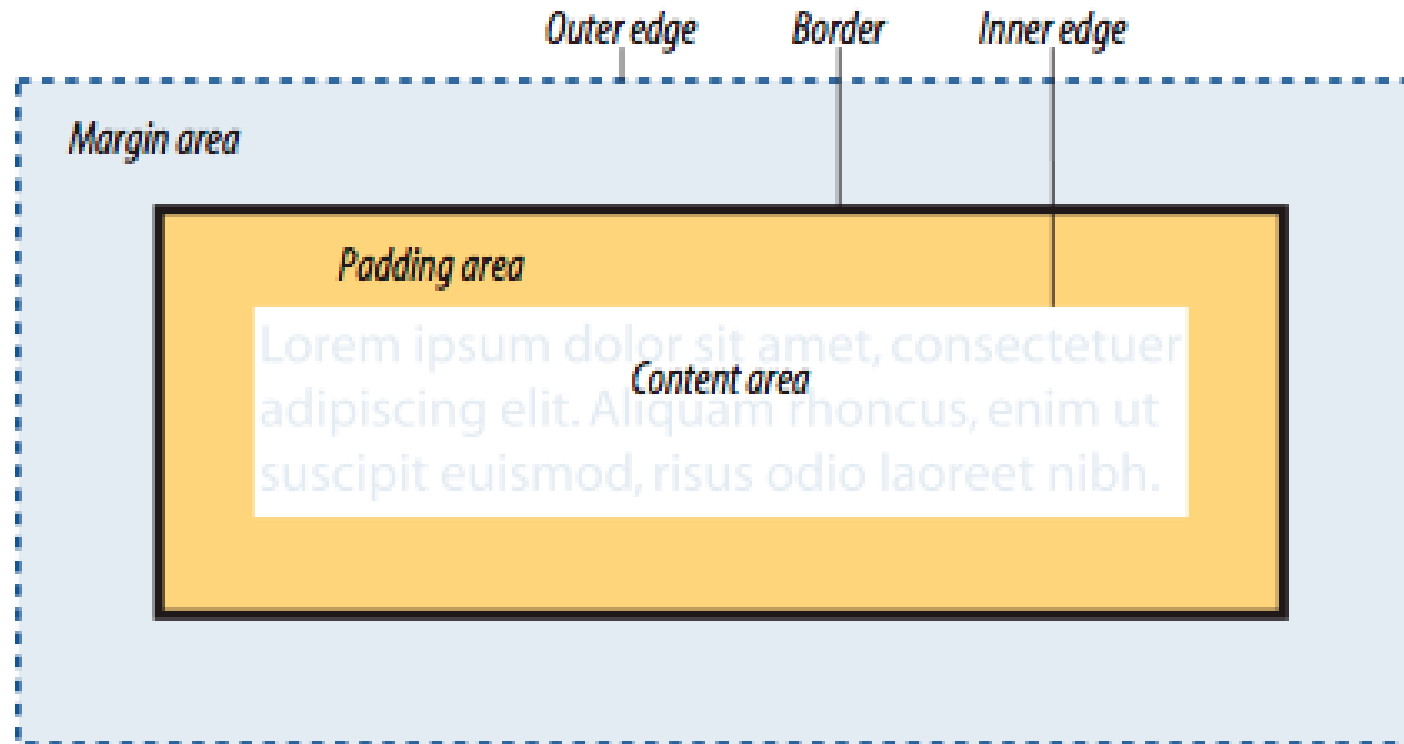
Classes and pseudo-classes

Elements and pseudo-elements

^{*} Bigger
is
more
specific

CSS BOX MODEL

The Browser sees every element, block or inline, as a little rectangular box



CSS BOX MODEL

Standard CSS Box Model

- This is the standard way the box model is calculated.
- **CSS *box-sizing* property is:**
 - Not set
 - Set to content-box
- **Width of an element on screen is:**
 - *width + padding + border*
 - *width* property sets content area width
 - You must account for padding and borders when setting the width to get the exact size you want.
- **Supported in CSS 1**
- https://developer.mozilla.org/en-US/docs/Web/CSS/box_model
- <http://css-tricks.com/the-css-box-model/>

CSS BOX MODEL

New Border Box CSS Box Model

- This is the new way the box model can be calculated.
- **CSS *box-sizing* property is:**
 - Set to border-box
- **Width of an element on screen is:**
 - Whatever you set the width property to.
 - The browser will take the width of the padding and border out of the content area so your final box is the size you set.
- **CSS 3 Property that has fairly good browser support**
 - <http://caniuse.com/#feat=css3-boxsizing>
 - Even back to IE8
- <http://www.paulirish.com/2012/box-sizing-border-box-ftw/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing>
- <http://css-tricks.com/box-sizing/>

ITMD-361

NAVIGATION

CSS HORIZONTAL NAVIGATION

- **Navigation is a list of links so we should use a list as the html element**
- **ul**
 - Remove the bullets – list-style-type: none
 - Remove the margins and padding
- **ul li**
 - Float list items left - float: left;
- **ul li a**
 - Make anchor block and apply styles – display:block;
- **Clear the element that follows – clear:left;**

ITMD-361

CSS RESETS

CSS RESETS

- Used to reset all css properties so they will be consistent in all browsers.
- Eric Meyer's css reset
 - Complete reset to no styles
 - <http://meyerweb.com/eric/tools/css/reset/>
- Normalize
 - Targets only styles that need to be reset
 - <http://necolas.github.com/normalize.css/>
- Yahoo YUI css reset
 - Mostly complete reset
 - <http://yuilibrary.com/yui/docs/cssreset/>
- <http://www.cssreset.com/>

ITMD-361

CSS FLOATS



CSS FLOATS

- **“The float CSS property specifies that an element should be taken from the normal flow and placed along the left or right side of its container, where text and inline elements will wrap around it. A floating element is one where the computed value of float is not none.” – Mozilla MDN**
- **<https://developer.mozilla.org/en-US/docs/Web/CSS/float>**
- **Float value can be left, right, or none. Value of none turns off float**
- **Float property causes element to move as far as possible to the left or right and allow other elements to wrap around it.**
- **Always provide a width to floated element**
- **Floated elements behave like block elements**
- **Margins do not collapse like they do on a standard block element (top and bottom margins)**

CSS FLOATS

- **Clearing the float.**
 - Apply the clear property to the element following the float that you don't want to wrap around the content.
 - It tells the element it is on that it can not be positioned next to a floated element.
- **If a containing element only contains floated elements it will have no height unless you set some other properties.**
 - Search clearfix
 - Or use overflow property set to hidden.
- **You can use floats to create column layouts**
- **Floated elements need to appear in html before content that wraps around them**

ITMD-361

CSS POSITION

CSS POSITION

4 values for CSS positioning (position property)

- **Static**
 - Normal default position in the document flow
- **Relative**
 - element is moved relative to position in document flow
 - Space in document flow **is reserved**
- **Absolute**
 - Removed from the document flow
 - Positioned with respect to **containing element with a position**
- **Fixed**
 - Removed from flow and positioned fixed to the window
- **Then use top, bottom, left, or right properties to set position**
- **z-index property sets render layer. Only works when element has position set**