

# COMP5347: Web Application Development

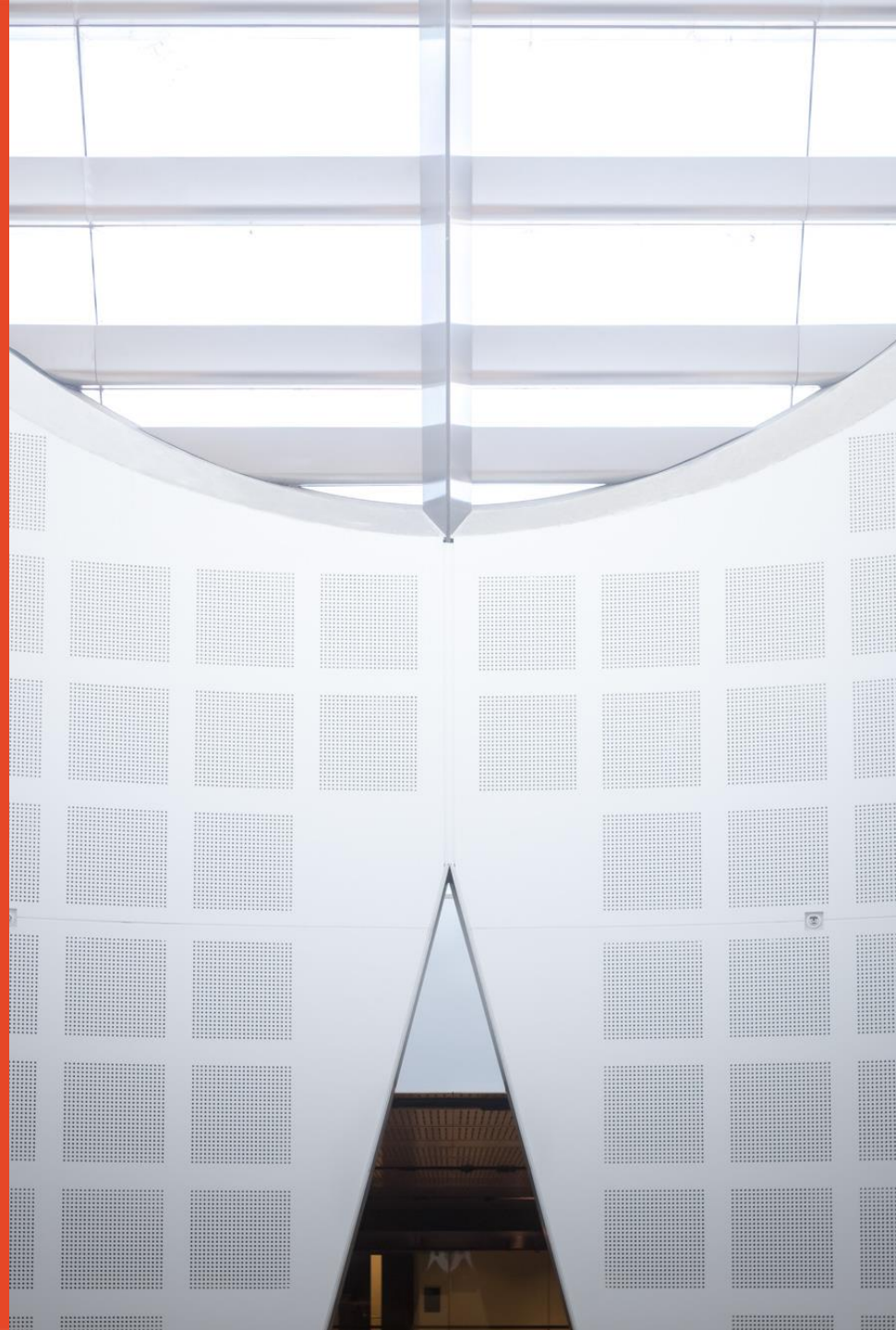
## Introduction to HTML and CSS

Dr. Basem Suleiman

School of Computer Science



THE UNIVERSITY OF  
SYDNEY



**COMMONWEALTH OF  
Copyright Regulations 1969  
WARNING**

This material has been reproduced and communicated to you by or on behalf of the University of Sydney pursuant to Part VB of the Copyright Act 1968 (**the Act**).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

**Do not remove this notice.**

# Outline

- **HTML**
  - **Syntax**
  - **Structure**
  - **Quick Tour**
  - **Semantic Markups**
- **CSS**
  - **Syntax and Location**
  - **Selector**
  - **The box model**
  - **Text Styling**

Content based on Ch2 and Ch3 of Fundamentals of Web Development, Global Edition.

# HTML

*“To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML” –  
(from Hyper Text Markup Language)*

- A markup language is simply a way of annotating a document in such a way to make the annotations distinct from the text being annotated.
  - E.g. Latex
- HTML is used world wide, standardization is essential
  - The W3C is the main standards organization for the World Wide Web.

# Key events in the history of HTML

- First public specification by Tim Berners-Lee (1991)
- W3C standardization (1997)
- Frozen at 4.01 by W3C (1998)
- W3C's XHTML (XML + HTML) (late 1990s)
  - Survived till XHTML 2.0 in mid 2000
- WHATWG(Web Hypertext Application Technology Working Group) under W3C was formed by developers at Opera and Mozilla to challenge the idea of XML+HTML (mid 2000)
  - More practical approach, e.g., specify how should browsers deal with invalid mark-up
- W3C stopped work on XHTML 2.0 and adopted the work by WHATWG and named it **HTML5** (2009)

# HTML5 – Aims

- Specify unambiguously how browsers should deal with invalid markup.
- Provide an open, non-proprietary programming framework (via Javascript) for creating rich web applications.
- Be backwards compatible with the existing web.

# HTML: Elements and Attributes

- **HTML documents** = text content + HTML elements
- An **HTML element**
  - Text, other elements, or be empty
  - Identified in the HTML document by tags
- HTML elements can also contain **HTML attribute**
  - **name=value** pair that provides more information about the HTML element

# What HTML lets you do

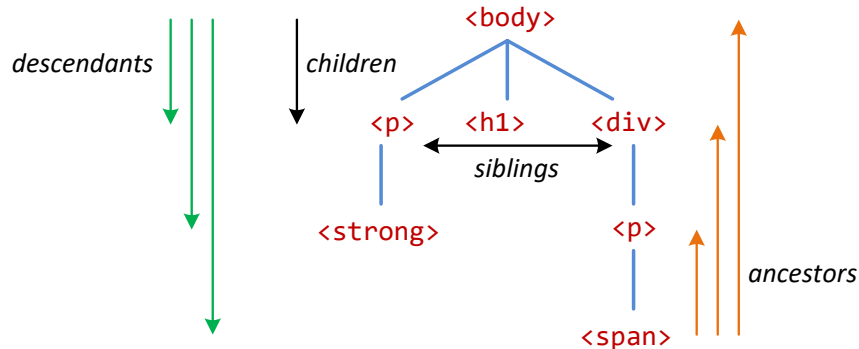
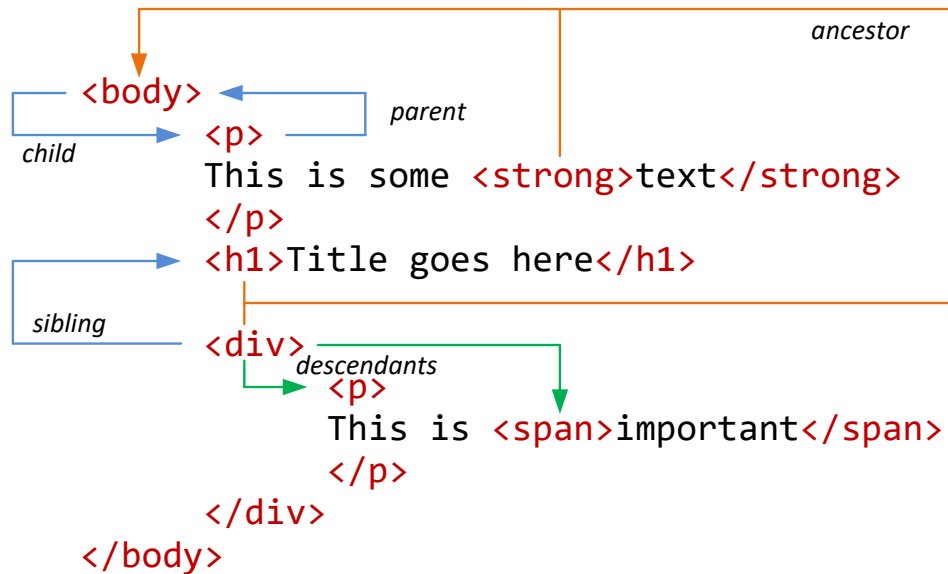
- Insert images using the `<img>` tag
- Create links with the `<a>` tag
- Create lists with the `<ul>`, `<ol>` and `<li>` tags
- Create headings with `<H1>`, `<H2>`, ..., `<H6>`
- Define metadata with `<meta>` tag
- And much more...



# Elements and Attributes



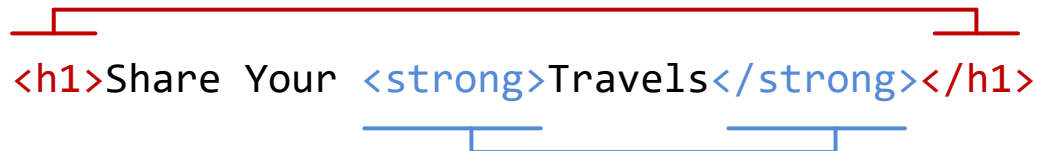
# Nesting HTML elements



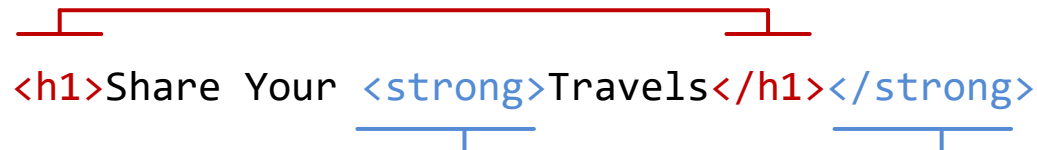
# Nesting HTML elements

- Browsers expect proper nesting of HTML elements
  - Important for proper construction
  - A child's ending tag must occur before its parents ending tag

Correct Nesting



```
<h1>Share Your <strong>Travels</strong></h1>
```



```
<h1>Share Your <strong>Travels</h1></strong>
```

Incorrect Nesting

# Outline

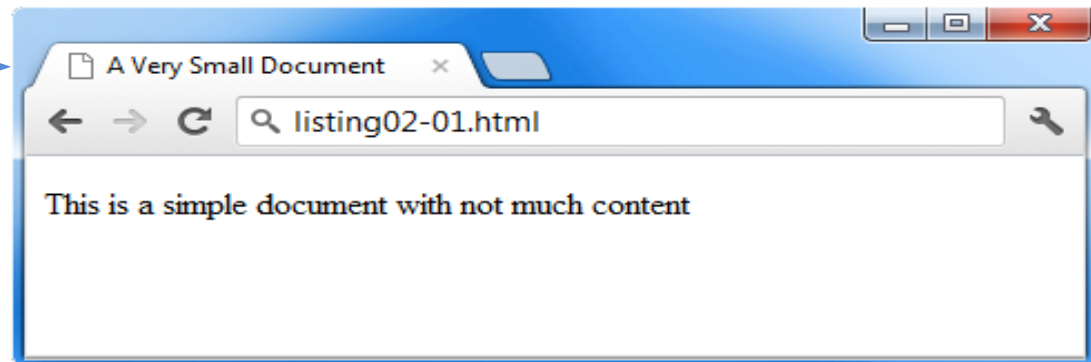
- **HTML**
  - Syntax
  - **Structure**
  - Quick Tour
  - Semantic Markups
- **CSS**
  - Syntax and Location
  - Selector
  - The box model
  - Text Styling

# Simplest HTML document

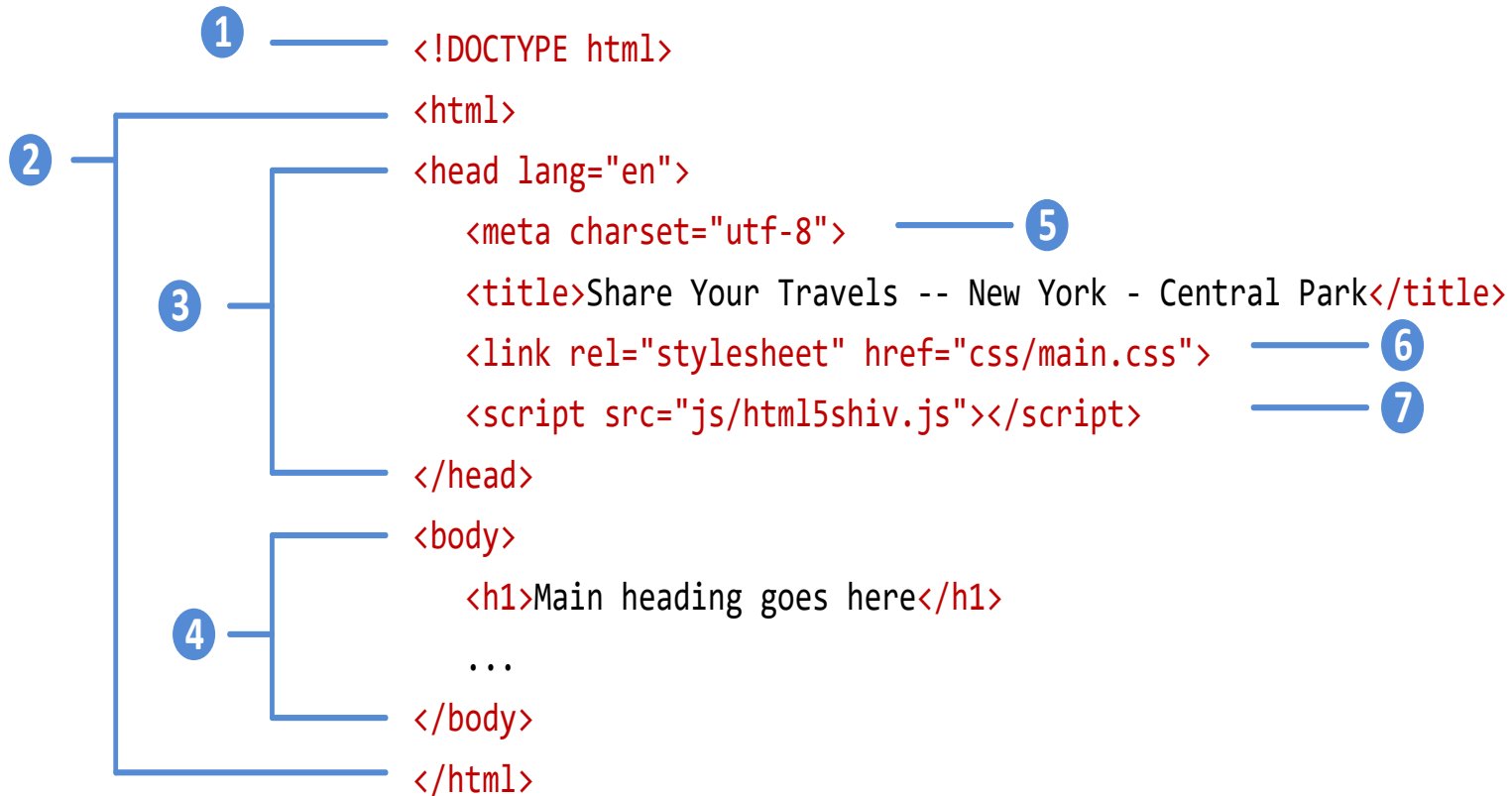
- The <title> element provides a broad description of the content.
- The title is typically displayed by the browser in its window and/or tab

1

```
<!DOCTYPE html>  
<title>A Very Small Document</title>  
<p>This is a simple document with not much content</p>
```

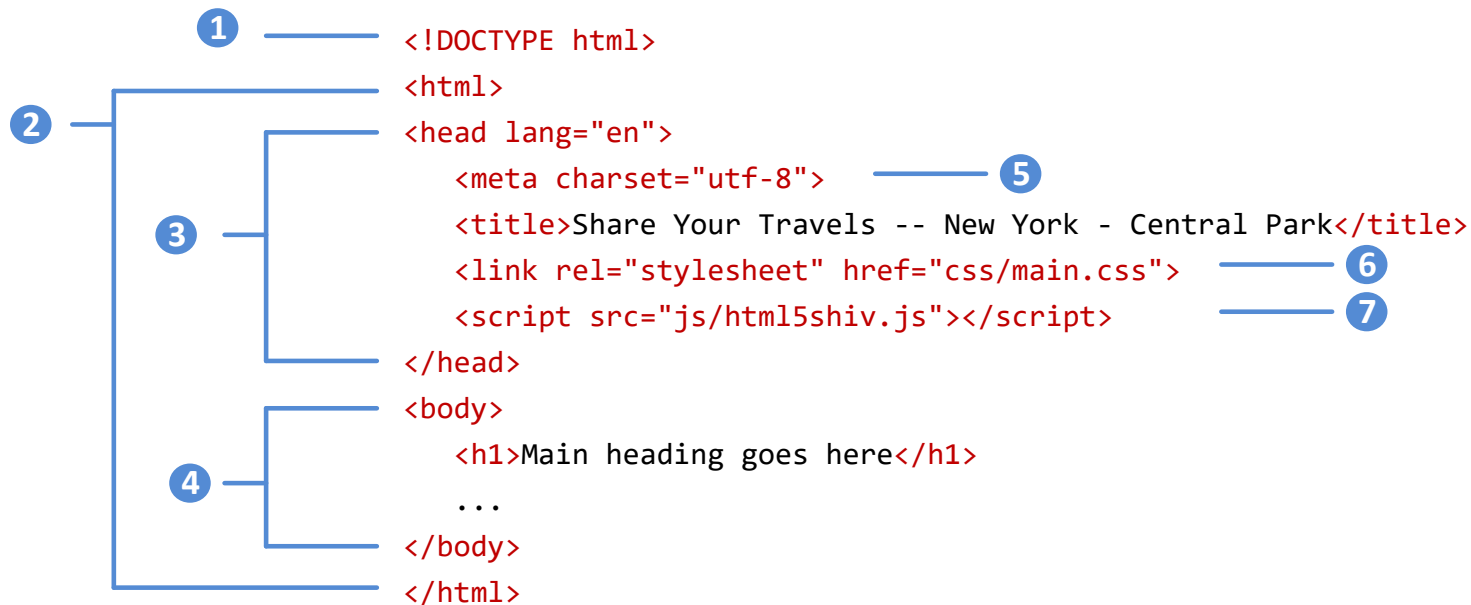


# A more complete document



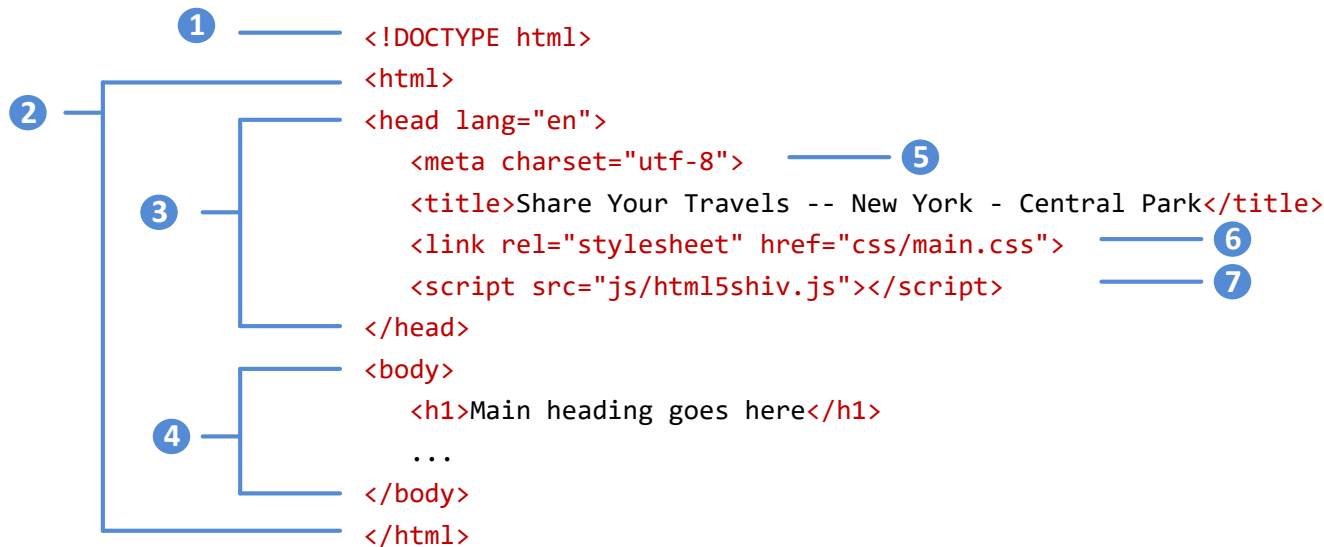
# DOCTYPE

- Tells the browser what type of document it is about to process
- It does not indicate the HTML version



# HTML, Head, and Body

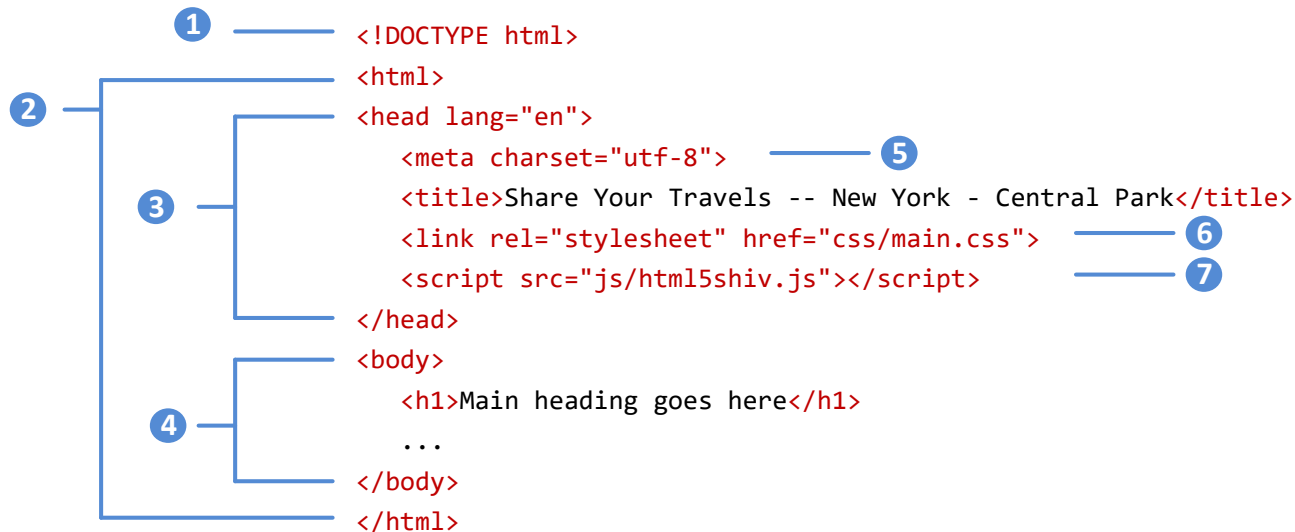
- `<html>`, `<head>`, and `<body>` not required in HTML 5
- Required in XHTML most web authors continue to use them
- The `<html>` contains all other elements (the **root element**)





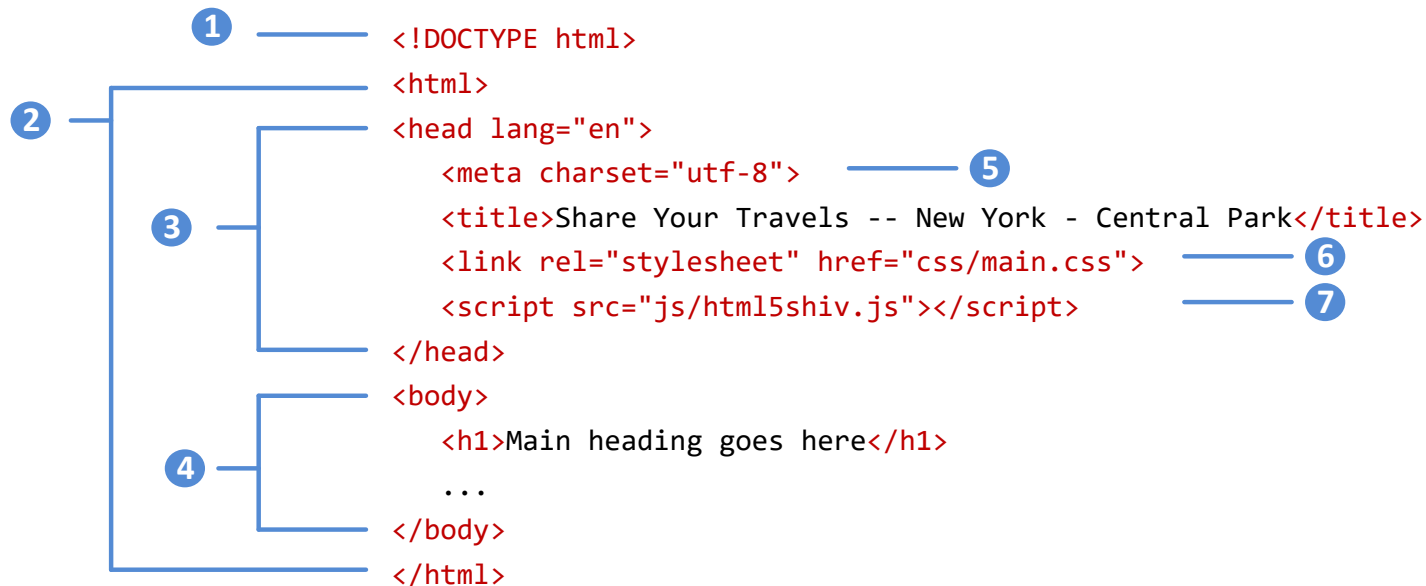
# Head and Body

- HTML pages are divided into: the **head** and the **body**
- The **head** contains descriptive elements *about* the document
- The **body** contains content to displayed by the browser



# Inside the head

- `<meta>` declares the character encoding for the document, e.g., UTF-8
- `<link>` specifies reference to external file (e.g., CSS stylesheet that defines the visual look)



# Outline

- **HTML**
  - Syntax
  - Structure
  - **Quick Tour**
  - Semantic Markups
- **CSS**
  - Syntax and Location
  - Selector
  - The box model
  - Text Styling

# Sample Document

<body>

1 `<h1>Share Your Travels</h1>`

2 `<h2>New York - Central Park</h2>`

`<p>Photo by Randy Connolly</p>`

`<p>This photo of Conservatory Pond in`

`<a href="http://www.centralpark.com/">Central Park</a>` 3

`New York City was taken on October 22, 2011 with a`

`<strong>Canon EOS 30D</strong> camera.`

`</p>`

5 `` 4

`<h3>Reviews</h3>`

6 `<div>`

`<p>By Ricardo on <time>September 15, 2012</time></p>`

`<p>Easy on the HDR buddy.</p>`

`</div>`

`<div>`

`<p>By Susan on <time>October 1, 2012</time></p>`

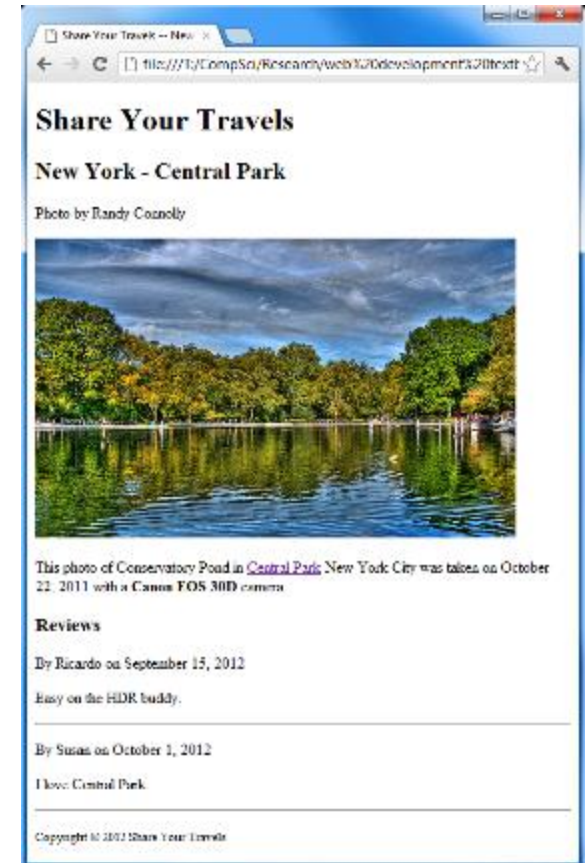
`<p>I love Central Park.</p>`

`</div>`

8 `<p><small>Copyright &copy; 2012 Share Your Travels</small></p>`

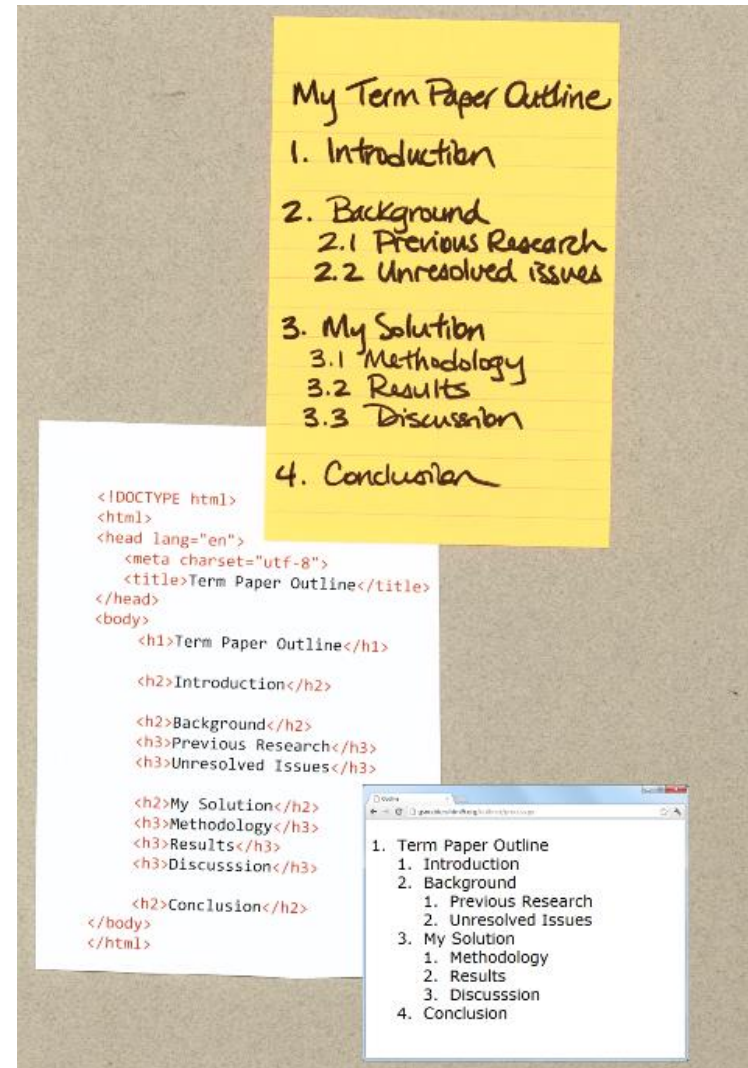
`</body>`

9



# HTML Headings, Paragraphs and Division

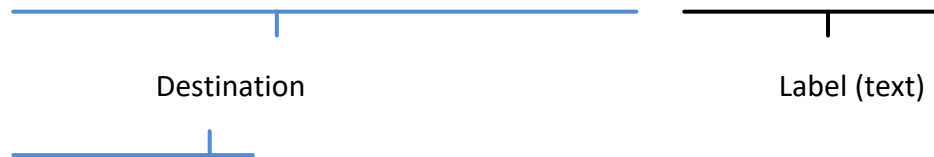
- Six levels of headings (h1, h2, ... h6)
- Used by the browser to create a document outline for the page
  - Tool support: [HTML 5 Outliner](#)
- Browser's default styling (customize by CSS)
- `<p>` (*paragraph*) container for text and HTML elements
- `<div>` (*division*) container element used for logical grouping of content



# Links

- Created using the `<a>` element (*anchor*)
- Consists of: the **destination** and the **label**

```
<a href="http://www.centralpark.com">Central Park</a>
```



```
<a href="index.html"></a>
```



# HTML Types of Links (1)

Link to external site

`<a href="http://www.centralpark.com">Central Park</a>`

Link to resource on external site

`<a href="http://www.centralpark.com/logo.gif">Central Park</a>`

Link to another page on same site as this page

`<a href="index.html">Home</a>`

Link to another place on the same page

`<a href="#top">Go to Top of Document</a>`

...

`<a name="top">`

Defines anchor for a link to another place on same page

# HTML Types of Links (2)

Link to specific place on another page

`<a href="productX.html#reviews">Reviews for product X</a>`

Link to email

`<a href="mailto:person@somewhere.com">Someone</a>`

Link to JavaScript function

`<a href="javascript:OpenAnnoyingPopup();">See This</a>`

Link to telephone (automatically dials the number  
when user clicks on it using a smartphone browser)

`<a href="tel:+18009220579">Call toll free (800) 922-0579</a>`



# Types of Links

- Links to external sites (or to individual resources such as images or movies on an external site)
- Links to other pages or resources within the current site
- Links to other places within the current page
- Links to a particular location on another page
- Links that are instructions to the browser to start the user's email program
- Links that are instructions to the browser to execute a Javascript function

# Images

- Different ways to specify images:
  - Decorative images in CSS (not in the markup)
    - Background gradients and patterns, logos, border art, etc.
  - Content images using `<img>` tag
    - Images in a gallery or the image of a product in a product details page

Specifies the URL of the image to display  
(note: uses standard relative referencing)

Text in title attribute will be displayed in a popup  
tool tip when user moves mouse over image.

```

```

Text in alt attribute provides a brief  
description of image's content for users who  
are unable to see it.

Specifies the width and height of  
image in pixels.

# Lists

- **Unordered lists <ul>**

Collections of items in no certain order; by default rendered by the browser as a bulleted list

- **Ordered lists <ol>**

Collections of items that have a set order; by default rendered by the browser as a numbered list

- **Description lists <dl>**

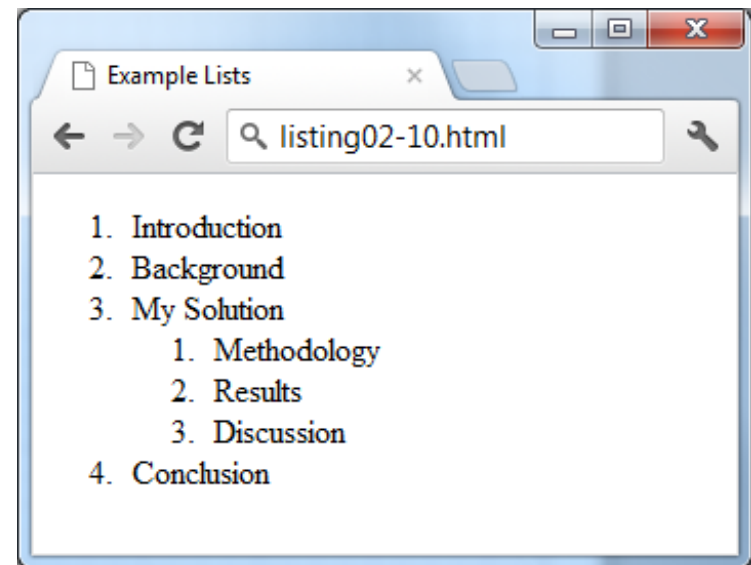
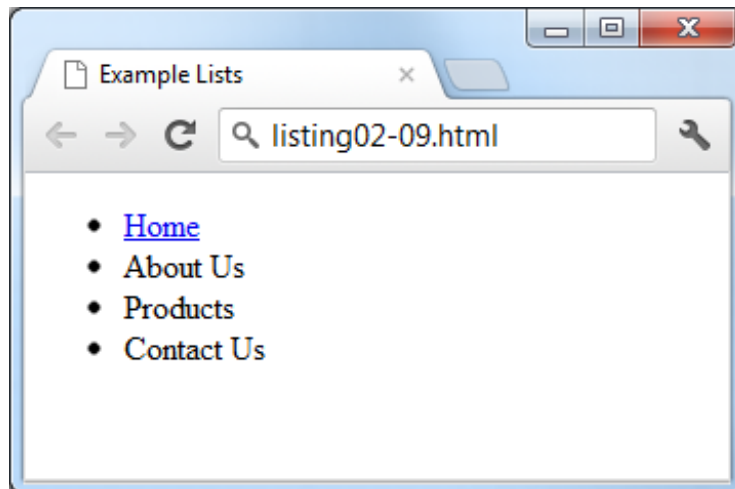
Collection of name and definition pairs

# Lists

Notice that the list item element can contain other HTML elements

```
<ul>
  <li><a href="index.html">Home</a></li>
  <li>About Us</li>
  <li>Products</li>
  <li>Contact Us</li>
</ul>
```

```
<ol>
  <li>Introduction</li>
  <li>Background</li>
  <li>My Solution</li>
  <li>
    <ol>
      <li>Methodology</li>
      <li>Results</li>
      <li>Discussion</li>
    </ol>
  </li>
  <li>Conclusion</li>
</ol>
```



# HTML Elements – Special Characters

- No easy way to type them via a keyboard
- Have a reserved meaning in HTML (like“<“)

Entity	Description
&nbsp;	Nonbreakable space
&lt;	<
&gt;	>
&copy;	©
&trade;	TM

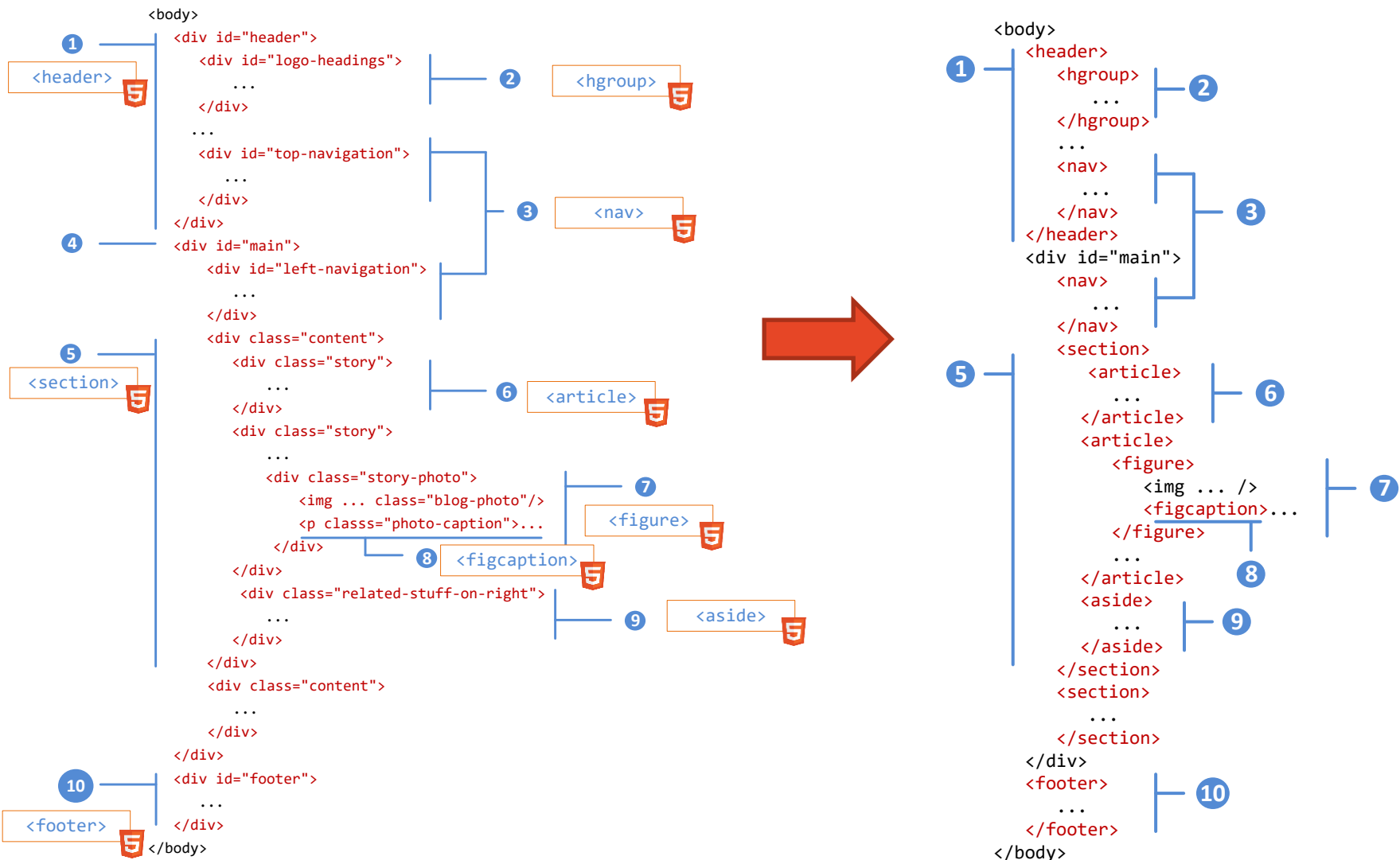
# Outline

- **HTML**
  - Syntax
  - Structure
  - Quick Tour
  - **Semantic Markups**
- **CSS**
  - Syntax and Location
  - Selector
  - The box model
  - Text Styling

# Semantic markups

- The commonly agreed good practice
  - HTML should only focus on the structure of a document
  - CSS should specify how the content should be displayed in a browser window
- HTML5 has introduced many new semantic tags
  - `<article>`, `<section>`, `<header>`, `<footer>`, `<aside>`, etc to replace tags like
    - `<div id="header">`, `<div id = "footer">`, etc

# XHTML versus HTML5





# Header and Footer

– Typically the **header** contains

- the site logo
- title (and perhaps additional subtitles or taglines)
- horizontal navigation links, and
- perhaps one or two horizontal banners.

1

```
<header>
  ...
  <nav>
    ...
  </nav>
</header>
```

– The typical footer contains less important material, e.g.,

- smaller text versions of the navigation,
- copyright notices,
- information about the site's privacy policy, and
- perhaps twitter feeds or links to other social sites.

```
<footer>
  ...
</footer>
```

9

# Navigation

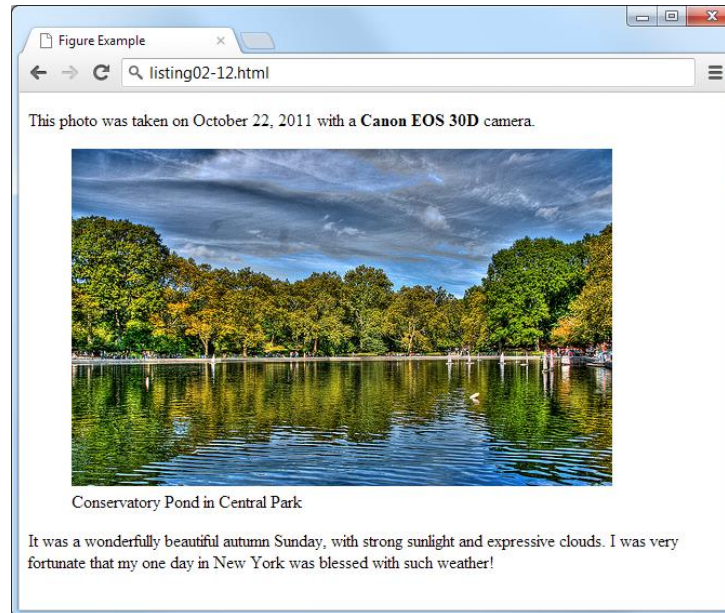
- The `<nav>` represents a section of a page that contains links to other pages or to other parts within the same page
- The browser does not apply any special presentation to the `<nav>`
- The `<nav>` was intended for major navigation blocks, presumably the global and secondary navigation systems

```
<header>
  
  <h1>Fundamentals of Web Development</h1>
  <nav role="navigation">
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="about.html">About Us</a></li>
      <li><a href="browse.html">Browse</a></li>
    </ul>
  </nav>
</header>
```

# Figure and Figure Captions

Figure could be moved to a different location in document  
...  
But it has to exist in the document (i.e., the figure isn't optional)

```
<p>This photo was taken on October 22, 2011 with a Canon EOS 30D camera.</p>  
<figure>  
  <br/>  
  <figcaption>Conservatory Pond in Central Park</figcaption>  
</figure>  
<p>  
  It was a wonderfully beautiful autumn Sunday, with strong sunlight and  
  expressive clouds. I was very fortunate that my one day in New York was  
  blessed with such weather!  
</p>
```



# Semantic HTML

HTML 5 Element	Description / use
<code>&lt;main&gt;</code>	contain the main unique content of the document. It provides a semantic replacement for markup such as <code>&lt;div id="main"&gt;</code> or <code>&lt;div id="main-content"&gt;</code>
<code>&lt;section&gt;</code>	for a much broader element
<code>&lt;article&gt;</code>	for blocks of content that could potentially be read or consumed independently of the other content on the page
<code>&lt;aside&gt;</code>	for sidebars, pull quotes, groups of advertising images, or any other grouping of nonessential elements

# Semantic Markup – Benefits

- Maintainability
  - Easy to update and change (structure only!)
- Performance/speed
  - Faster to download and quicker to author
- Accessibility
  - Sight disabilities experience the web using voice reading software
  - Semantic web makes it easier to read
- Search Engine Optimization
  - Provides semantic elements to tell web crawlers of the important content

# Outline

- **HTML**
  - Syntax
  - Structure
  - Quick Tour
  - Semantic Markups
- **CSS**
  - **Syntax and Location**
  - Selector
  - The box model
  - Text Styling

# Cascading Style Sheets (CSS)

- W3C standard for describing the **presentation (or appearance)** of HTML elements
- With CSS, we can assign
  - font properties, colors, sizes, borders, background images, even the position of elements

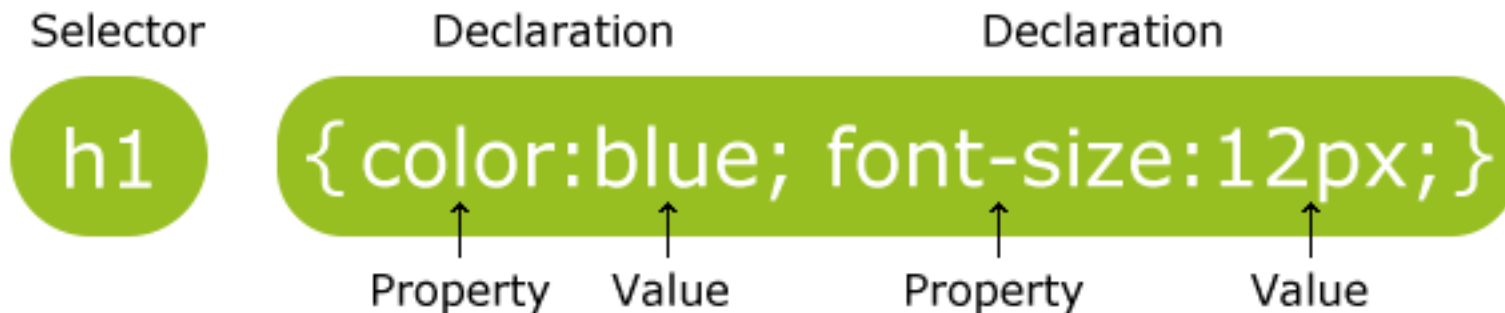
# CSS – brief history

- CSS came not too long after the proposal of HTML
- The first recommendation CSS Level 1 (CSS1) in 1996
- The CSS Level 2 (CSS2) was published in 1998
  - Updated version CSS2.1 became official W3C recommendation in 2011
- The current version is CSS3 with a variety of modules at different stages: recommendation, draft and so on
  - backwards-compatible with earlier versions of CSS.
- Browser support has long been an issue of CSS
  - Not all browsers have all parts of the CSS2 recommendation



# CSS Syntax

- A CSS document consists of one or more style rules
- Each rule consists of a selector followed by list of declarations (property-value pairs)



# CSS Syntax

- Every CSS rule begins with a selector
- The selector identifies which element/elements in the HTML document will be affected by the declarations in the rule
- Each individual CSS declaration must contain a property
- The CSS2.1 recommendation defines over a hundred different property names
- The unit of any given value is dependent upon the property.
- Some property values are from a predefined list of keywords. Others are values such as length measurements, percentages, numbers without units, color values, and URLs.

# CSS Syntax

- **Selector** identifies the element(s) to be affected by the styles
  - Selector can be based on name, class, id, attribute and more
- The **property** name predefined by CSS standard
- The **property value** are either predefined keywords (e.g., color name) or other values (e.g., length measurement, percentage)
  - The unit of any given value is dependent upon the property

# CSS Properties

Property Type	Property
Fonts	Font, font-family, font-size, font-style, font-weight, @font-face
Text	letter-spacing, line-height, text-align, text-decoration, *text-indent
Color and Background	Background, background-color, background-image, background-position, background-repeat, box-shadow, color, Opacity
Borders	border*, border-color, border-width, border-style, border-top, border-left, ...*, border-image*, border-radius

# Units of Measurement

- Relative vs. Absolute measurements
  - Relative units are based on the value of something else, (e.g., the size of a parent element)
  - Absolute units have real world size

Unit	Description	Type
in	Inches	Absolute
cm	Centimeters	Absolute
mm	Millimeters	Absolute
pt	Points (equal to 1/72 of an inch)	Absolute
pc	Pica (equal to 1/6 of an inch)	Absolute

# Relative Units

Unit	Description	Type
px	Pixel. In CSS2 this is a relative measure, while in CSS3 it is absolute (1/96 of an inch).	Relative (CSS2) Absolute (CSS3)
em	Equal to the computed value of the font-size property of the element on which it is used. When used for font sizes, the em unit is in relation to the font size of the parent.	Relative
%	A measure that is always relative to another value. The precise meaning of % varies depending upon which property it is being used.	Relative
ex	A rarely used relative measure that expresses size in relation to the x-height of an element's font.	Relative
ch	Another rarely used relative measure; this one expresses size in relation to the width of the zero ("0") character of an element's font.	Relative (CSS3 only)
rem	Stands for root em, which is the font size of the root element. Unlike <b>em</b> , which may be different for each element, the <b>rem</b> is constant throughout the document.	Relative (CSS3 only)
vw, vh	Stands for viewport width and viewport height. Both are percentage values (between 0 and 100) of the viewport (browser window). This allows an item to change size when the viewport is resized.	Relative (CSS3 only)

# CSS location – Inline

- Inline within an HTML element

```
<h1>Share Your Travels</h1>  
<h2 style="font-size: 24pt">Description</h2>  
...  
<h2 style="font-size: 24pt; font-weight:bold;">  
Reviews</h2>
```

- Not recommended practice
- Good for quick testing

# CSS location – Embedded Stylesheet

- Embedded stylesheet (internal) in a HTML document

```
<head>
  <meta charset="utf-8">
  <title>Share Your Travels -- New York - Central Park</title>
  <style>
    h1 { font-size: 24pt; }
    h2 {
      font-size: 18pt;
      font-weight: bold;
    }
  </style>
</head>
<body>
...
```

- Better than inline but it's not recommended



# CSS location – External

- External style placed in a separate file (.css extension) and are linked to the HTML file
  - `<link href="style.css" rel="stylesheet">`

```
<head>  
    <meta charset="utf-8">  
    <title>Share Your Travels -- New York - Central  
        Park</title>  
    <link rel="stylesheet" href="styles.css" />  
</head>
```

- Highly recommended for maintainability

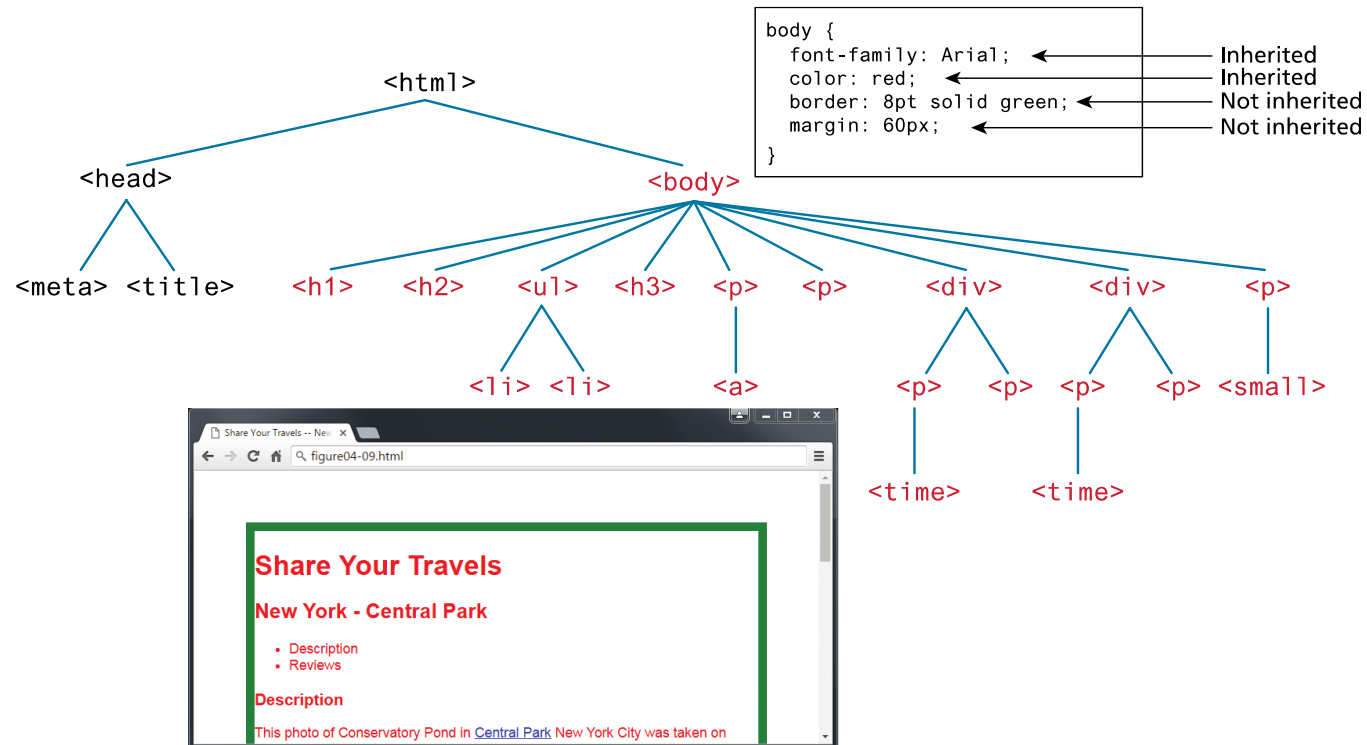
# CSS location – External

- Browser has a default set of rules for pages/elements that do not have a customized style
- Check CSS Default values reference for default CSS values for HTML elements
- In Chrome, check <https://chromium.googlesource.com/chromium/blink/+ /master /Source /core /css /html.css>

# What is “Cascading”?

- Cascading is the principle to resolve conflicting style rules
  - Cascade: style rule will continue to take precedence with child elements
- **Inheritance**
  - Certain properties defined for the <body> element will be inherited by all elements embedded in it
  - Inheritable: font, color, list and text properties
  - Not Inheritable: layout, sizing, border, background and spacing properties
- **Specificity**
  - Style rules with more specific selector will override rules with more general selector
    - E.g. If both <body> and <p> elements define a font rule, the text inside the <p> element will use the font defined in the <p> element
- **Location**
  - The latest (closest) rule wins If inheritance and specificity cannot determine style precedence
    - E.g. inline rule is closer to that element than the rules in an external file

# “Cascading” – Inheritance



# Outline

- **HTML**
  - Syntax
  - Structure
  - Quick Tour
  - Semantic Markups
- **CSS**
  - Syntax and Location
  - **Selector**
  - The box model
  - Text Styling

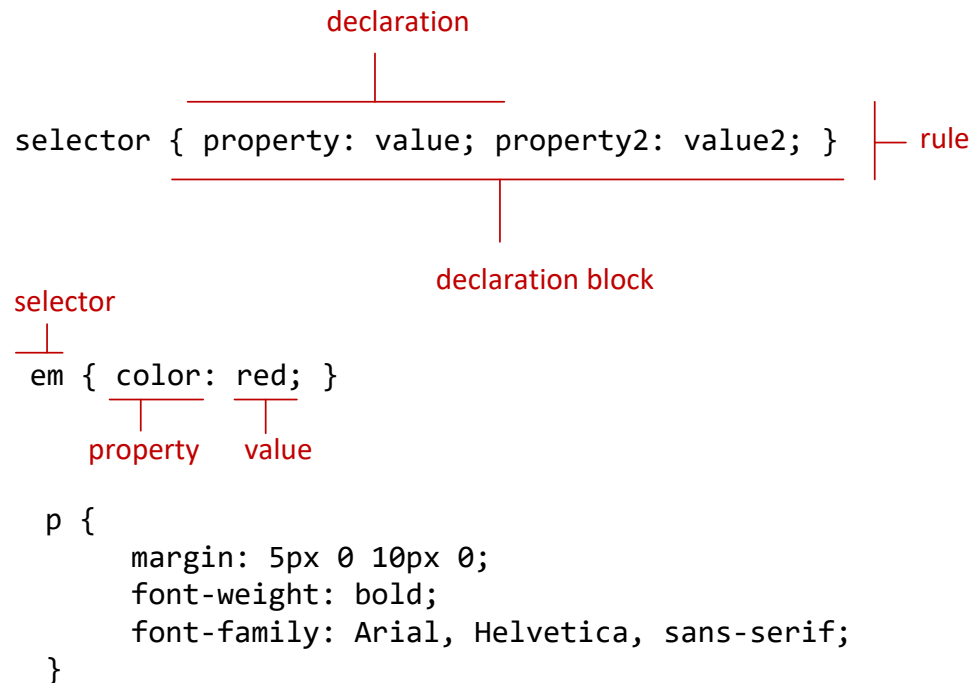
# Selectors

- A **selector** tells the browser which elements will be affected
- CSS selectors allow you to select
  - individual elements
  - multiple HTML elements,
  - elements that belong together in some way, or
  - elements that are positioned in specific ways in the document hierarchy.

# Element Selectors

Uses the HTML element name

You can select all elements by using the **universal element selector**, the **\*** (asterisk) character



# Grouped Selectors

```
/* commas allow you to group selectors */  
p, div, aside {  
    margin: 0;  
    padding: 0;  
}  
/* the above single grouped selector is equivalent to the  
   following: */  
p {  
    margin: 0;  
    padding: 0;  
}  
div {  
    margin: 0;  
    padding: 0;  
}  
aside {  
    margin: 0;  
    padding: 0;  
}
```

**LISTING 3.4** Sample grouped selector



# Class Selectors

- Allows you to simultaneously target different HTML elements regardless of their position in the document tree
- A series of HTML elements labeled with ***the same class attribute value*** can be targeted for styling by using a class selector
- How: period (.) followed by the **class name**

# Class Selectors

```
<head>
  <title>Share Your Travels </title>
  <style>
    .first {
      font-style: italic;
      color: brown;
    }
  </style>
</head>
<body>
  <h1 class="first">Reviews</h1>
  <div>
    <p class="first">By Ricardo on <time>September 15,
2012</time></p>
    <p>Easy on the HDR buddy.</p>
  </div>
  <hr/>

  <div>
    <p class="first">By Susan on <time>October 1, 2012</time></p>
    <p>I love Central Park.</p>
  </div>
  <hr/>
</body>
```



```
.first {
  font-style: italic;
  color: brown;
}
```

# Id Selectors

Allows you to target a specific element by its id attribute regardless of its type or position

If an HTML element has been labeled with an id attribute, then you can target it for styling by using an id selector

How: pound/hash (#) followed by the id name.

# Id vs. Class Selectors

- Referencing a single HTML element → id selector
  - id attribute can only be assigned to a single HTML element
- (Potentially) referencing several related elements → class selector

# Pseudo Selectors

- **Pseudo-element selector** a way to select something that does not exist explicitly as an element in the HTML document tree, but is still a recognizable selectable object
- **pseudo-class selector** does apply to an HTML element, but targets either a particular state or, in CSS3, a variety of family relationships
- The most common use of this type of selectors is for targeting link states

# Pseudo Selectors

```
<head>
  <title>Share Your Travels</title>
  <style>
    a:link {
      text-decoration: underline;
      color: blue;
    }
    a:visited {
      text-decoration: underline;
      color: purple;
    }
    a:hover {
      text-decoration: none;
      font-weight: bold;
    }
    a:active {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <p>Links are an important part of any web page. To learn more about
    links visit the <a href="#">W3C</a> website.</p>
  <nav>
    <ul>
      <li><a href="#">Canada</a></li>
      <li><a href="#">Germany</a></li>
      <li><a href="#">United States</a></li>
    </ul>
  </nav>
</body>
```

**LISTING 3.8** Styling a link using pseudo-class selectors

# Contextual Selectors

- It selects elements based on their context or their relation to other elements in the document tree
- Ancestors, descendants, or siblings
- Called combinators in CSS3

Selector	Matches	Example
<b>Descendant</b>	A specified element that is contained somewhere within another specified element	<code>div p</code>  Selects a <code>&lt;p&gt;</code> element that is contained somewhere within a <code>&lt;div&gt;</code> element. That is, the <code>&lt;p&gt;</code> can be any descendant, not just a child.
<b>Child</b>	A specified element that is a direct child of the specified element	<code>div&gt;h2</code>  Selects an <code>&lt;h2&gt;</code> element that is a child of a <code>&lt;div&gt;</code> element.
<b>Adjacent Sibling</b>	A specified element that is the next sibling (i.e., comes directly after) of the specified element.	<code>h3+p</code>  Selects the first <code>&lt;p&gt;</code> after any <code>&lt;h3&gt;</code> .
<b>General Sibling</b>	A specified element that shares the same parent as the specified element.	<code>h3~p</code>  Selects all the <code>&lt;p&gt;</code> elements that share the same parent as the <code>&lt;h3&gt;</code> .

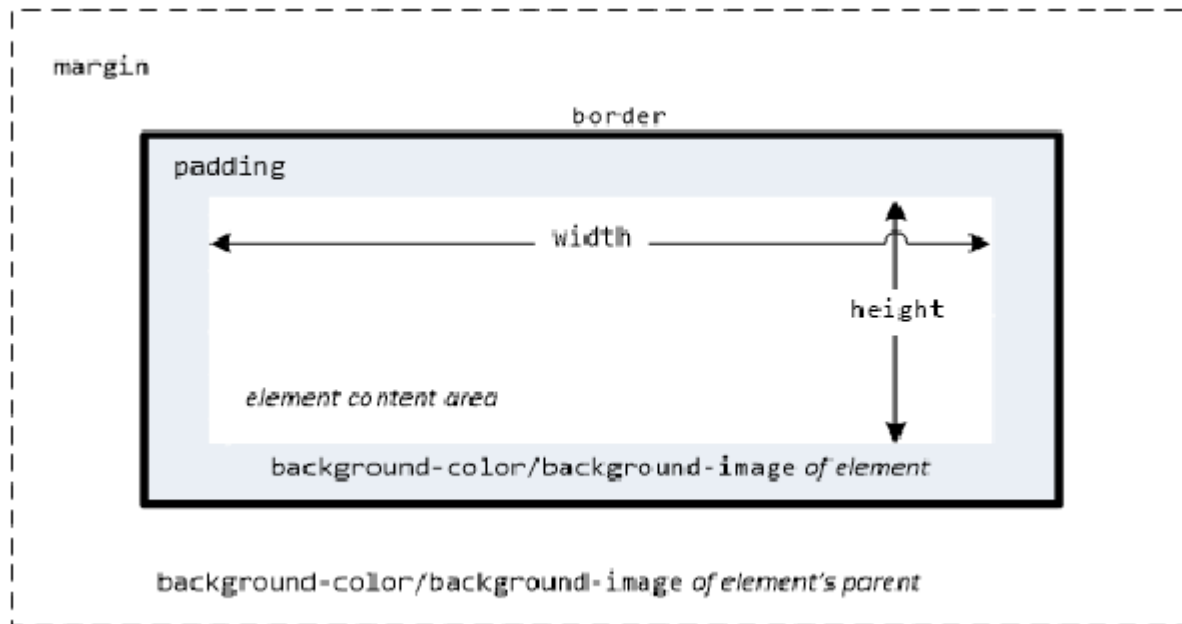
# Outline

- **HTML**
  - Syntax
  - Structure
  - Quick Tour
  - Semantic Markups
- **CSS**
  - Syntax and Location
  - Selector
  - **The box model**
  - Text Styling

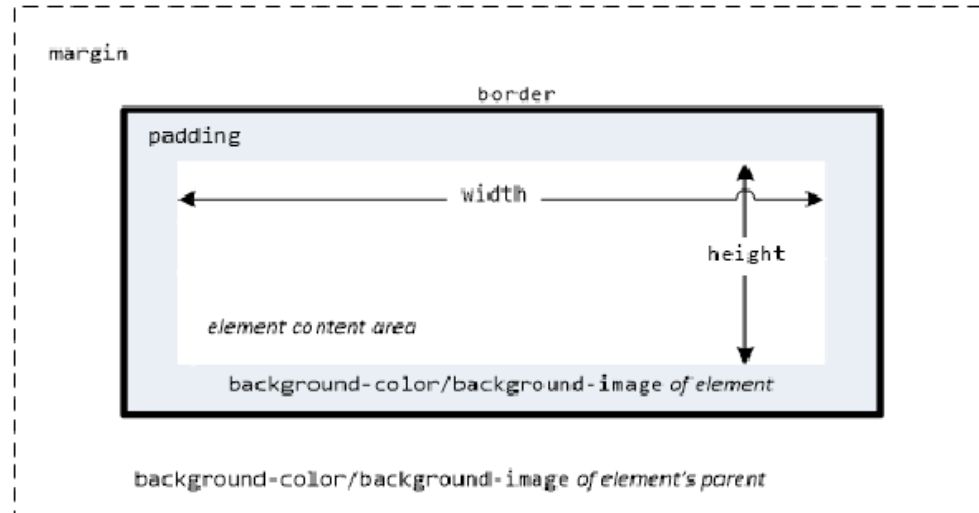


# The Box Model

In CSS, all HTML elements exist within an **element box**.



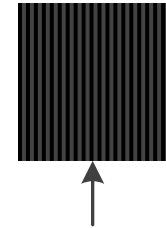
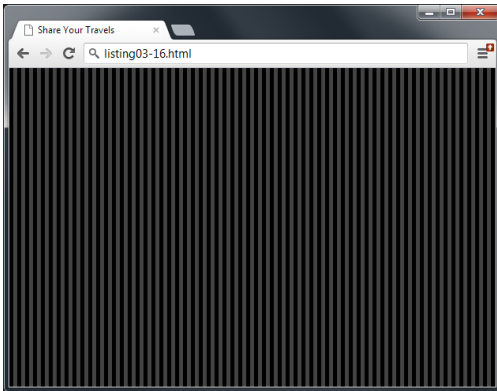
# The Box Model – Example



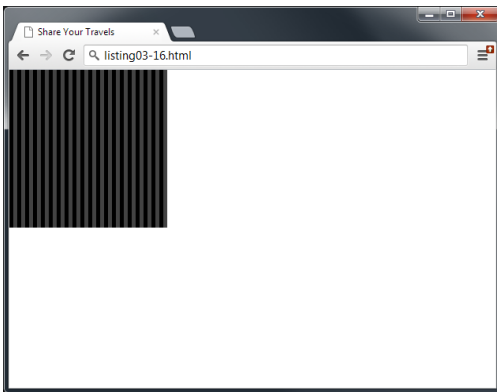
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML elements that will receive

# Background properties

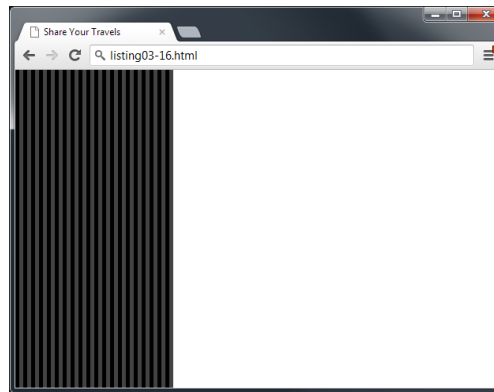
The background color or image fills an element out to its border (if it has one that is).



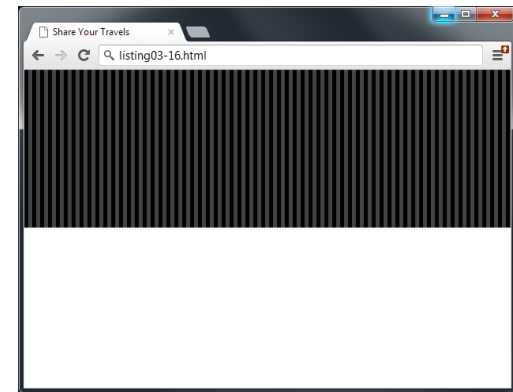
```
background-image: url(../images/backgrounds/body-background-tile.gif);  
background-repeat: repeat;
```



background-repeat: no-repeat;

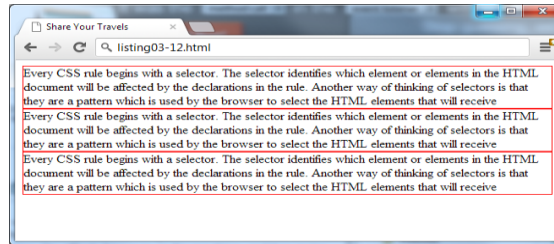


background-repeat: repeat-y;

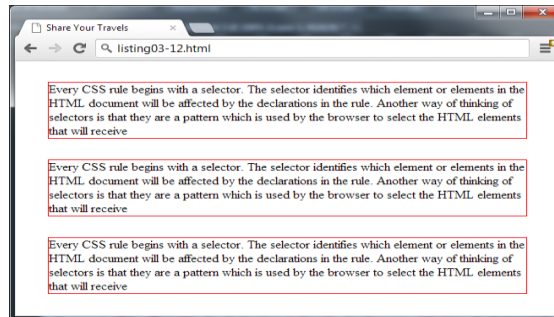


background-repeat: repeat-x;

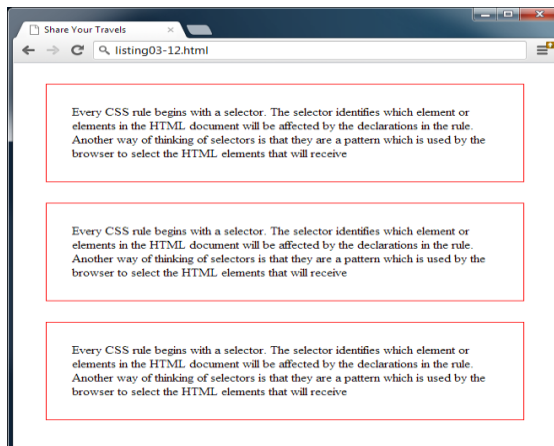
# Margins and Padding



```
p {  
  border: solid 1pt red;  
  margin: 0;  
  padding: 0;  
}
```



```
p {  
  border: solid 1pt red;  
  margin: 30px;  
  padding: 0;  
}
```



```
p {  
  border: solid 1pt red;  
  margin: 30px;  
  padding: 30px;  
}
```

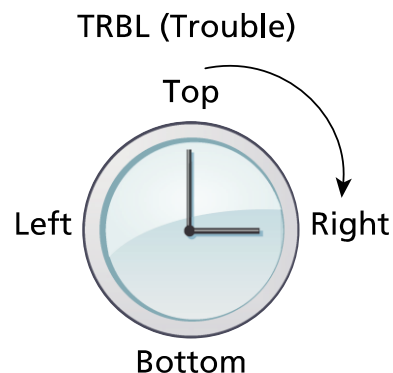
# Margins and padding properties

```
border-top-color: red; /* sets just the top side */  
border-right-color: green; /* sets just the right side */  
border-bottom-color: yellow; /* sets just the bottom side */  
border-left-color: blue; /* sets just the left side */
```

Alternately, we can set all four sides at once:

```
border-color: red; /* sets all four sides to red */
```

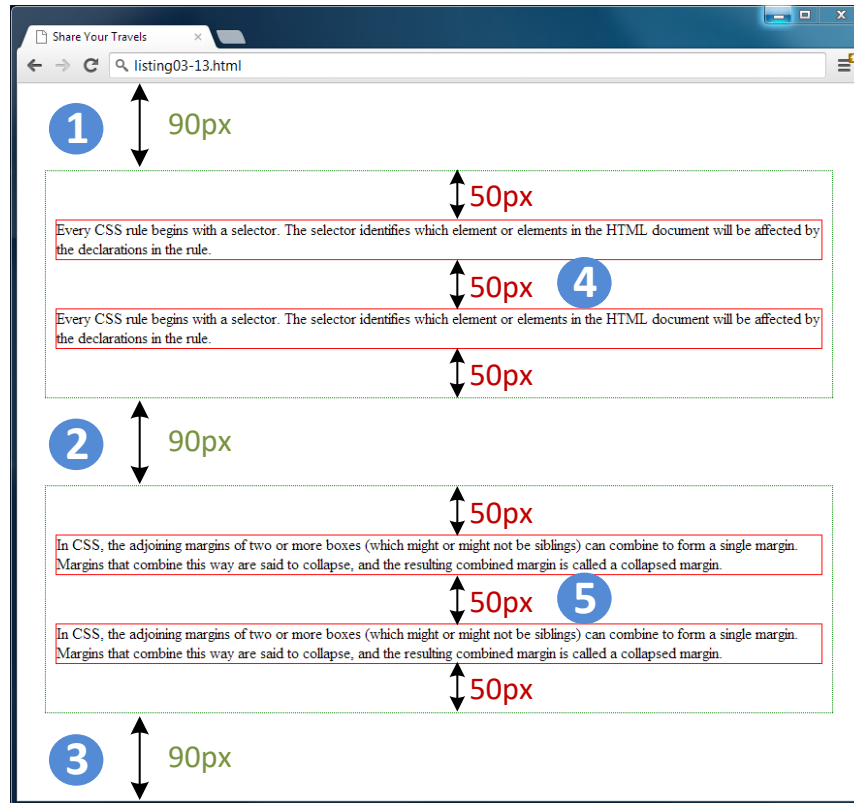
```
border-color: red green orange blue; /* sets 4 colors */
```



```
border-color: top right bottom left;
```

```
border-color: red green orange blue;
```

# Collapsing Margins



```
<div>
  <p>Every CSS rule ...</p>
  <p>Every CSS rule ...</p>
</div>
<div>
  <p>In CSS, the adjoining ... </p>
  <p>In CSS, the adjoining ... </p>
</div>
```

```
div {
  border: dotted 1pt green;
  padding: 0;
  margin: 90px 20px;
}
```

```
p {
  border: solid 1pt red;
  padding: 0;
  margin: 50px 20px;
}
```

If overlapping margins did not collapse, then margin space for ② would be 180p (90pixels for the bottom margin of the first <div> + 90 pixels for the top margin of the second <div>), while the margins ④ and ⑤ for would be 100px.

# Collapsing Margins

When the **vertical** margins of two elements touch,

- the largest margin value of the elements will be displayed
- the smaller margin value will be collapsed to zero

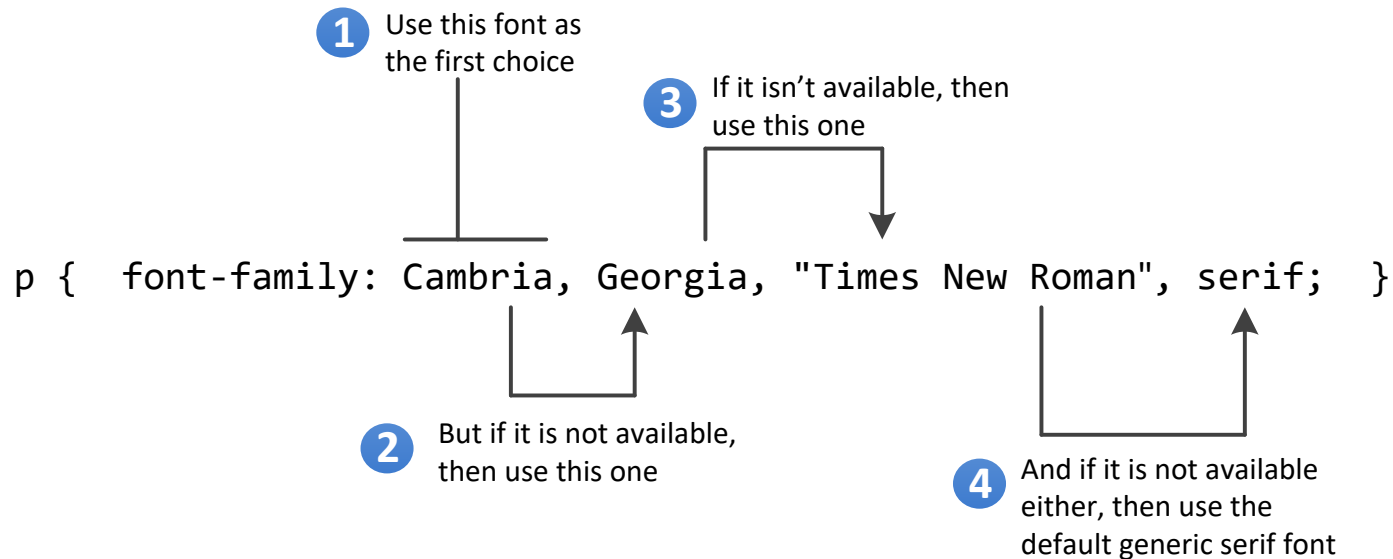
Horizontal margins, on the other hand, **never** collapse

# Outline

- **HTML**
  - Syntax
  - Structure
  - Quick Tour
  - Semantic Markups
- **CSS**
  - Syntax and Location
  - Selector
  - The box model
  - Text Styling



# Specifying the Font-Family



# Font sizes

- How do we specify font sizes?
  - Typically in word document, we specify 10 points, 12 points
  - Printer based, the size of point varied through history
- On screen-based media
  - Pixel has been a relatively stable measure
  - Device Pixel vs. CSS pixel
- Using relative measure gives better results on various devices

# How to use ems and percents

<code>&lt;body&gt;</code>	Browser's default text size is usually 16 pixels
<code>&lt;p&gt;</code>	100% or 1em is 16 pixels
<code>&lt;h3&gt;</code>	125% or 1.125em is 18 pixels
<code>&lt;h2&gt;</code>	150% or 1.5em is 24 pixels
<code>&lt;h1&gt;</code>	200% or 2em is 32 pixels

*/\* using 16px scale \*/*

```
body { font-size: 100%; }  
h3 { font-size: 1.125em; } /* 1.25 x 16 = 18 */  
h2 { font-size: 1.5em; }   /* 1.5 x 16 = 24 */  
h1 { font-size: 2em; }     /* 2 x 16 = 32 */
```

`<body>`

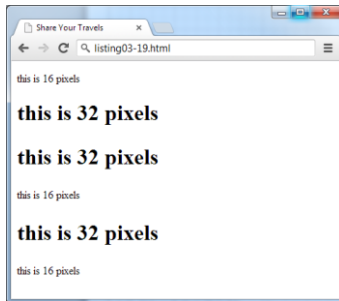
```
<p>this will be about 16 pixels</p>  
<h1>this will be about 32 pixels</h1>  
<h2>this will be about 24 pixels</h2>  
<h3>this will be about 18 pixels</h3>  
<p>this will be about 16 pixels</p>  
</body>
```

- Useful Tool: [PX to EM Converter](#)

# ems and percent's complications – Question

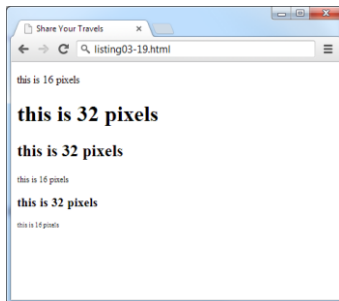
```
<body>
  <p>this is 16 pixels</p>
  <h1>this is 32 pixels</h1>
  <article>
    <h1>this is 32 pixels</h1>
    <p>this is 16 pixels</p>
    <div>
      <h1>this is 32 pixels</h1>
      <p>this is 16 pixels</p>
    </div>
  </article>
</body>
```

*Percent and em units are relative to their parents, so if the parent font size changes, this affects all of its contents.*



*/\* using 16px scale \*/*

```
body { font-size: 100%; }
p    { font-size: 1em;  }
h1   { font-size: 2em;  }
```



*/\* using 16px scale \*/*

```
body { font-size: 100%; }
p    { font-size: 1em;  }
h1   { font-size: 2em;  }

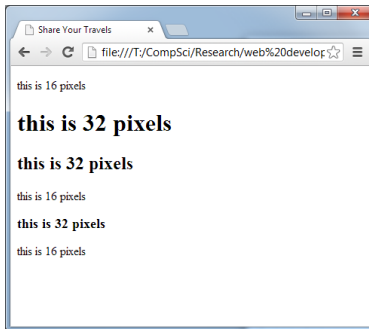
article { font-size: 75% }

div    { font-size: 75% }
```

# The rem unit

CSS3 supports a new relative measure, the **rem** (for root em unit).

This unit is always relative to the size of the root element (i.e., the `<html>` element).



```
/* using 16px scale */
```

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

```
p {
```

```
    font-size: 16px; /* for older browsers: won't scale properly though */
```

```
    font-size: 1rem; /* for new browsers: scales and simple too */
```

```
}
```

```
h1 { font-size: 2em; }
```

```
article { font-size: 75% } /* h1 = 2 * 16 * 0.75 = 24px
```

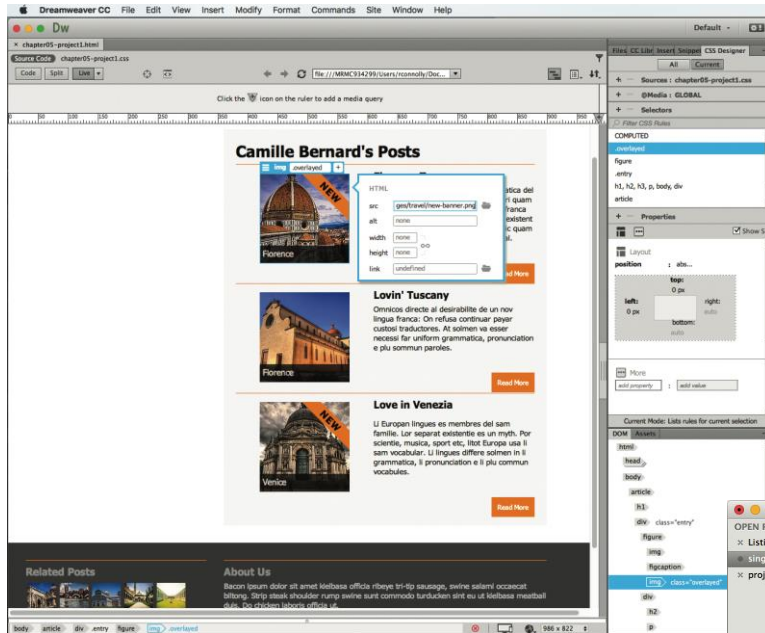
```
p = 1 * 16 = 16px */
```

```
div { font-size: 75% }
```

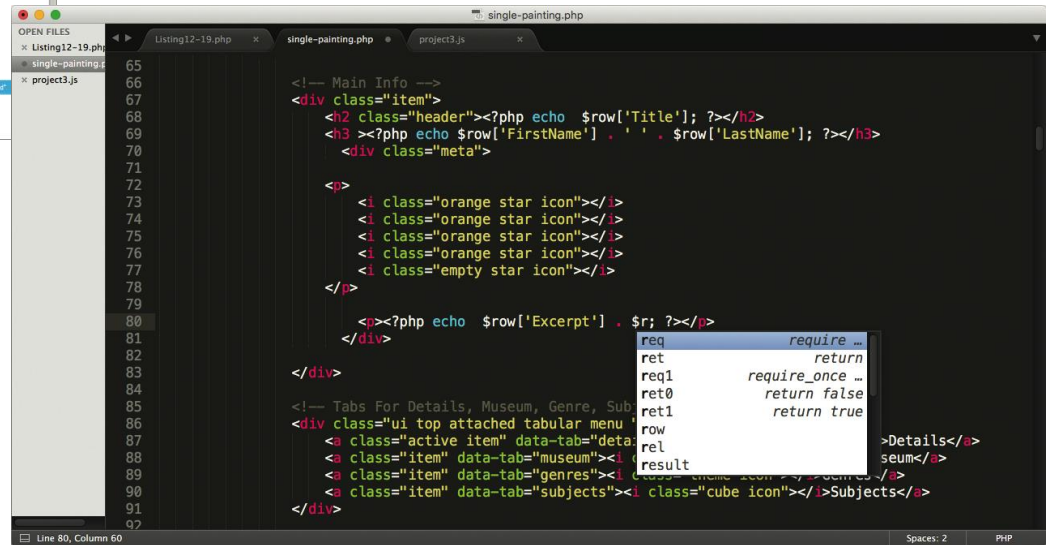
```
/* h1 = 2 * 16 * 0.75 * 0.75 = 18px
```

```
p = 1 * 16 = 16px */
```

# Web Development Tools



## WYSIWYG Editors



## Code Editors

# References

- Randy Connolly, Ricardo Hoar, Fundamentals of Web Development, Global Edition, Pearson
- W3Schools, HTML Tutorial  
[<https://www.w3schools.com/html/default.asp>]
- W3Schools, CSS tutorial  
[<https://www.w3schools.com/css/default.asp>]

# W2 Tutorial: HTML and CSS

## Week 3: More HTML and JavaScript Client-side Scripting



THE UNIVERSITY OF  
SYDNEY

