

# CSS 1: Introduction

## Chapter 3

# What is CSS?

You be styling soon

CSS is a 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.

# What is CSS?

You be styling soon

CSS is a language in that it has its own syntax rules.

CSS can be added directly to any HTML element (via the style attribute), within the **<head>** element, or, most commonly, in a separate text file that contains only CSS.

# Benefits of CSS

Why using CSS is a better way of describing presentation than HTML

- The degree of formatting control in CSS is significantly better than that provided in HTML.
- Web sites become significantly more maintainable because all formatting can be centralized into one, or a small handful, of CSS files.
- A site built using a centralized set of CSS files for all presentation will also be quicker to download because each individual HTML file will contain less markup.
- CSS can be used to adapt a page for different output mediums.

# CSS Versions

Let's just say there's more than 1

- W3C published the CSS Level 1 Recommendation in 1996.
- A year later, the CSS Level 2 Recommendation (also more succinctly labeled simply as CSS2) was published.
- Even though work began over a decade ago, an updated version of the Level 2 Recommendation, [CSS2.1](#), did not become an official W3C Recommendation until June 2011.
- And to complicate matters even more, all through the last decade (and to the present day as well), during the same time the CSS2.1 standard was being worked on, a different group at the W3C was working on a [CSS3](#) draft.

# Browser Adoption

Insert obligatory snide comment about Internet Explorer 6 here

While Microsoft's Internet Explorer was an early champion of CSS, its later versions (especially IE5, IE6, and IE7) for Windows had uneven support for certain parts of CSS2.

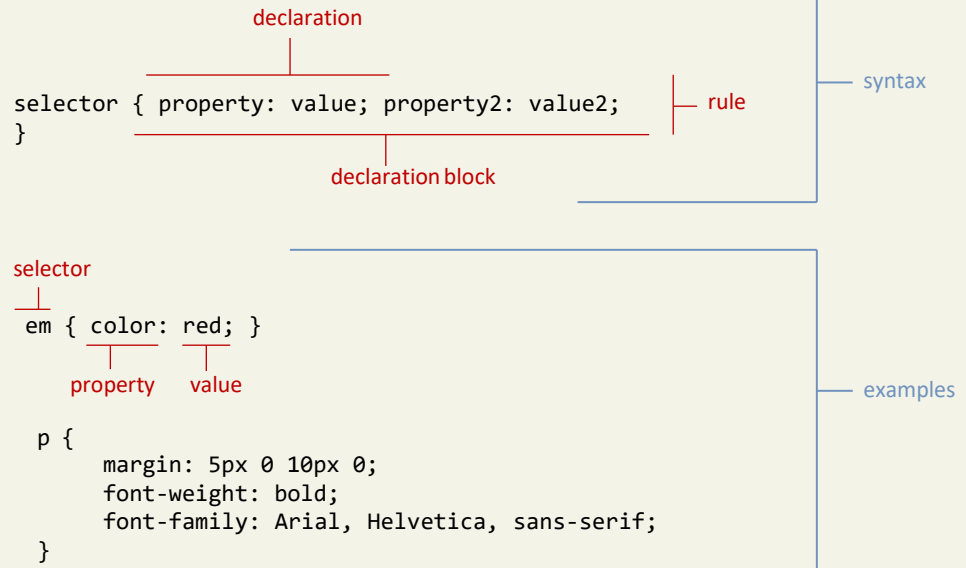
In fact, all browsers have left certain parts of the CSS2 Recommendation unimplemented.

# CSS Syntax

Rules, properties, values, declarations

A CSS document consists of one or more **style rules**.

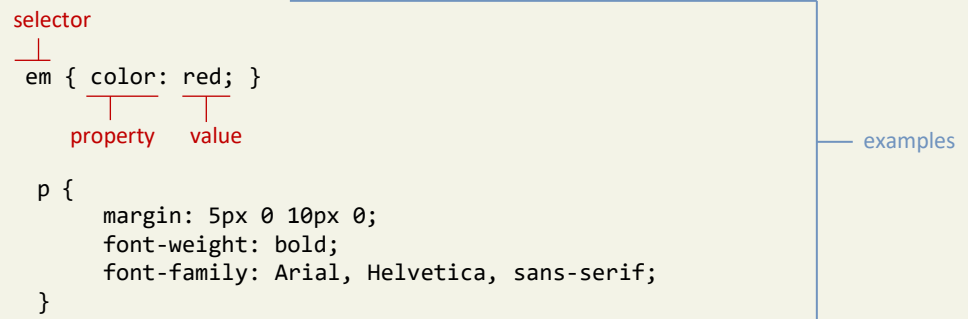
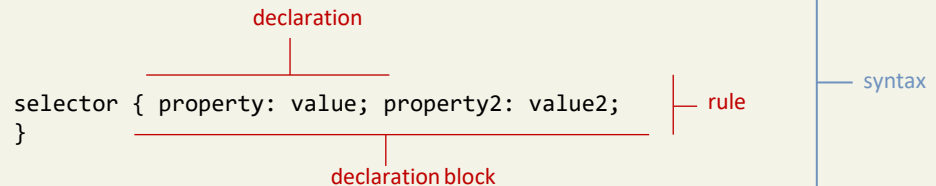
A rule consists of a selector that identifies the HTML element or elements that will be affected, followed by a series of **property** and **value** pairs (each pair is also called a **declaration**).



# Declaration Blocks

The series of declarations is also called the **declaration block**.

- A declaration block can be together on a single line, or spread across multiple lines.
- The browser ignores white space
- Each declaration is terminated with a semicolon.



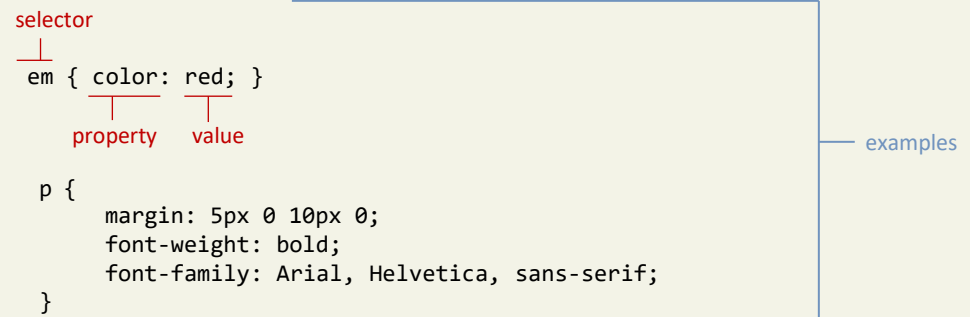
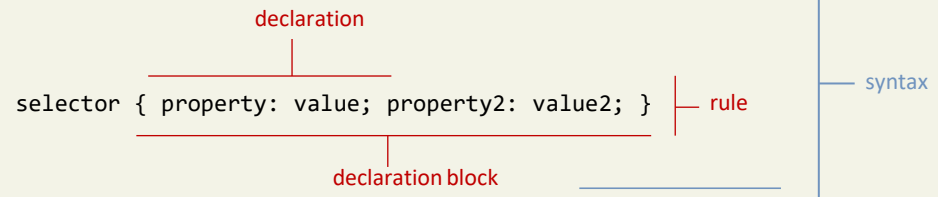


# Selectors

Which elements

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.



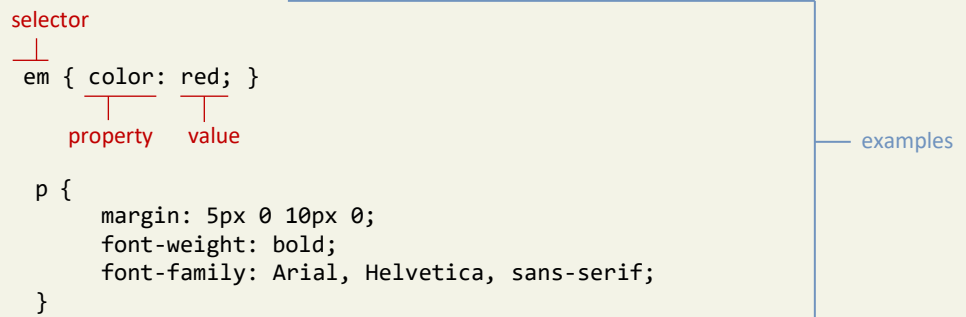
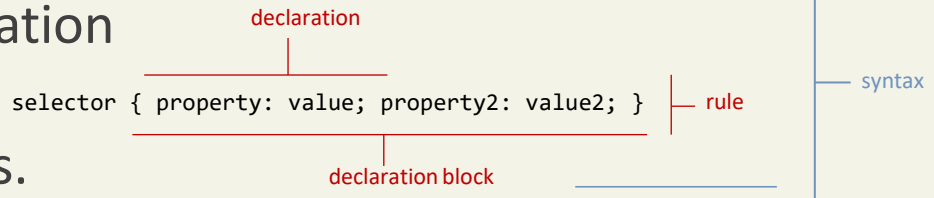
# Properties

Which style properties of the selected elements

Each individual CSS declaration must contain a **property**.

These property names are predefined by the CSS standard.

The CSS2.1 Recommendation defines over a hundred different property names.



# Values

What style value for the properties

Each CSS declaration also contains a **value** for a property.

- The unit of any given value is dependent upon the property.
- Some property values are from a predefined list of keywords, font family.
- Others are values such as length measurements, percentages, color values, and URLs.

Section 3 of 7

# LOCATION OF STYLES

# Actually there are three ...

Different types of style sheet

**Author-created style sheets** (what we are learning in this presentation).

**User style sheets** allow the individual user to tell the browser to display pages using that individual's own custom style sheet.

The **browser style sheet** defines the default styles the browser uses for each HTML element.

# Style Locations

Author Created CSS style rules can be located in three different locations

CSS style rules can be located in three different locations.

- Inline
- Embedded
- External

You can combine all 3!

# Inline Styles

Style rules placed within an HTML element via the style attribute

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

**LISTING 3.1** Internal styles example

An inline style only affects the element it is defined within and will override any other style definitions for the properties used in the inline style.

Using inline styles is generally discouraged since they increase bandwidth and decrease maintainability.

# Embedded Style Sheet

Style rules placed within the `<style>` element inside the `<head>` element

```
<head lang="en">
  <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>
  <h1>Share Your Travels</h1>
  <h2>New York - Central Park</h2>
  ...

```

**LISTING 3.2** Embedded styles example

While better than inline styles, using embedded styles is also by and large discouraged.

Since each HTML document has its own `<style>` element, it is more difficult to consistently style multiple documents when using embedded styles.



# External Style Sheet

Style rules placed within a external text file with the .css extension

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

**LISTING 3.3** Referencing an external style sheet

This is by far the most common place to locate style rules because it provides the best maintainability.

- When you make a change to an external style sheet, all HTML documents that reference that style sheet will automatically use the updated version.
- The browser is able to cache the external style sheet which can improve the performance of the site

# Cascading

```
<head>
<title>Cascade Order Example</title>
  <style>
    p {
      color: blue;
    }
    p {
      color: red;
    }
  </style>
</head>
<body>
  <p>This text will be red due to the cascade order.</p>
</body>
```

Section 4 of 7

# SELECTORS

# Selectors

Things that make your life easier

When defining CSS rules, you will need to first need to use a **selector** to tell the browser which elements will be affected.

CSS selectors allow you to select

- individual elements
- multiple HTML elements
- There are a number of different selector types.

# Element Selectors

Selects all instances of a given HTML element

Uses the HTML element name.

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

The diagram illustrates the components of CSS rules and declarations using red lines and labels:

- rule**: A bracket on the right side of a CSS rule, spanning from the selector to the closing curly brace.
- selector**: A bracket on the left side of a CSS rule, identifying the element being targeted.
- declaration block**: A bracket below the curly braces of a rule, encompassing all the declarations within.
- declaration**: A bracket above a single line of code, identifying the property-value pair.
- property**: A bracket below the text part of a declaration, identifying the CSS property.
- value**: A bracket below the text part of a declaration, identifying the value assigned to the property.

```
selector { property: value; property2: value2; }  
  
selector  
em { color: red; }  
      property  value  
  
p {  
  margin: 5px 0 10px 0;  
  font-weight: bold;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

# Grouped Selectors

Selecting multiple things

```
/* 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

You can select a group of elements by separating the different element names with commas.

This is a sensible way to reduce the size and complexity of your CSS files, by combining multiple identical rules into a single rule.

# Reset

```
html, body, div, span, h1, h2, h3, h4, h5, h6, p {  
  margin: 0;  
  padding: 0;  
  border: 0;  
  font-size: 100%;  
  vertical-align: baseline;  
}
```

Grouped selectors are often used as a way to quickly **reset** or remove browser defaults.

The goal of doing so is to reduce browser inconsistencies with things such as margins, line heights, and font sizes.

These reset styles can be placed in their own css file (perhaps called `reset.css`) and linked to the page **before** any other external styles sheets.

# Class Selectors

Simultaneously target different HTML elements

A **class selector** allows you to simultaneously target different HTML elements in the document tree.

If a series of HTML elements have been labeled with ***the same class attribute value***, then you can target them for styling by using a class selector, which takes the form: 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

Target a specific element by its id attribute

An **id selector** allows you to target a specific element by its id attribute regardless of its type.

If an HTML element has been labeled with an id attribute, then you can target it for styling by using an id selector, which takes the form: pound/hash (#) followed by the id name.

Note: You should only be using an **id** once per page

# Id Selectors

```
<head lang="en">
  <meta charset="utf-8">
  <title>Share Your Travels -- New York - Central Park</title>
  <style>
    #latestComment {
      font-style: italic;
      color: brown;
    }
  </style>
</head>
<body>
  <h1>Reviews</h1>
  <div id="latestComment">
    <p>By Ricardo on <time>September 15, 2012</time></p>
    <p>Easy on the HDR buddy.</p>
  </div>
  <hr/>

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



```
#latestComment {
  font-style: italic;
  color: brown;
}
```

# Id versus Class Selectors

How to decide

Id selectors should only be used when referencing a single HTML element since an id attribute can only be assigned to a single HTML element.

Class selectors should be used when (potentially) referencing several related elements.

Section 5 of 7

# THE CASCADE: HOW STYLES INTERACT

# Why Conflict Happens

In CSS that is

Because

- there are three different types of style sheets (author-created, user-defined, and the default browser style sheet),
- author-created stylesheets can define multiple rules for the same HTML element,

CSS has a system to help the browser determine how to display elements when different style rules conflict.

# Cascade

How conflicting rules are handled in CSS

The “Cascade” in CSS refers to how conflicting rules are handled.

The visual metaphor behind the term **cascade** is that of a mountain stream progressing downstream over rocks. Waterfall.

The downward movement of water down a cascade is meant to be analogous to how a given style rule will continue to take precedence with child elements.

# Cascade Principles

CSS uses the following cascade principles to help it deal with conflicts:

- inheritance,
- specificity,
- location



# Inheritance

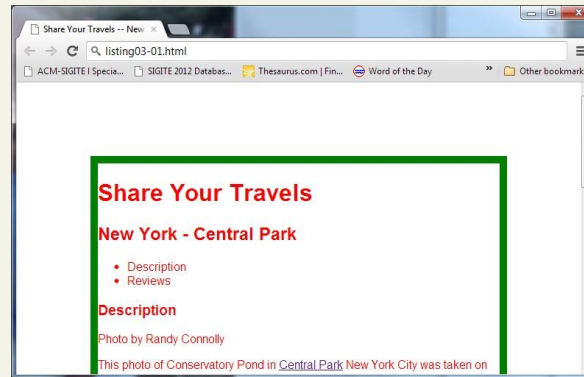
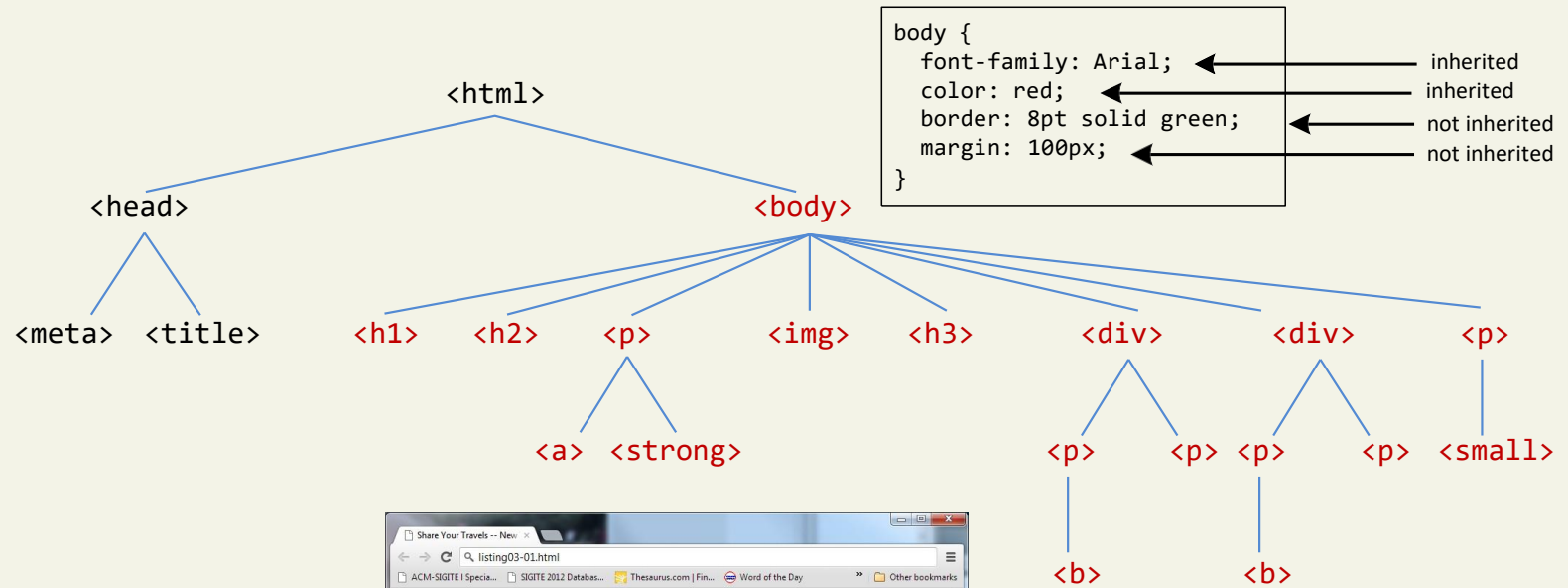
## Cascade Principle #1

Many (but not all) CSS properties affect not only themselves but their descendants as well.

Font, color, list, and text properties are inheritable.

Layout, sizing, border, background and spacing properties are not.

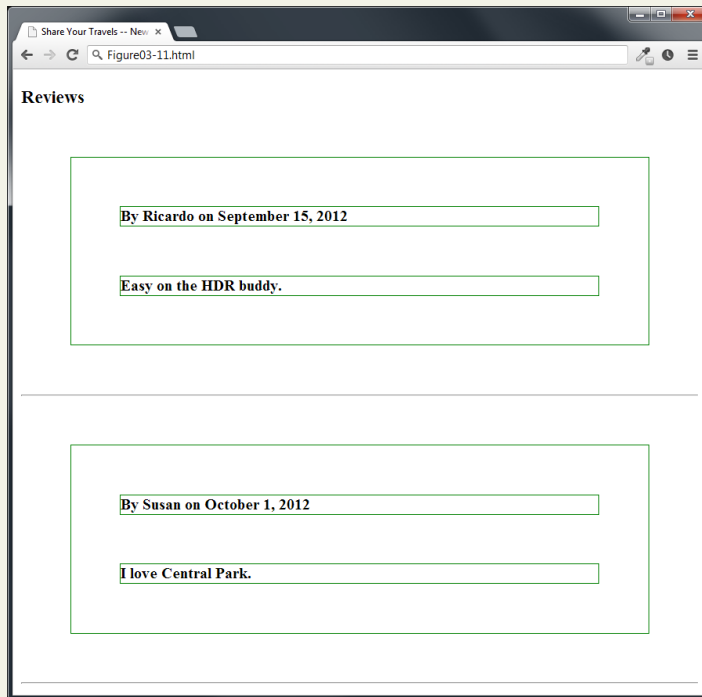
# Inheritance



# Inheritance

How to force inheritance

It is possible to tell elements to inherit properties that are normally not inheritable.



```
div {  
  font-weight: bold;  
  margin: 50px;  
  border: 1pt solid green;  
}  
p {  
  border: inherit;  
  margin: inherit;  
}  
  
<h3>Reviews</h3>  
<div>  
  <p>By Ricardo on <time>September 15, 2012</time></p>  
  <p>Easy on the HDR buddy.</p>  
</div>  
<hr/>  
  
<div>  
  <p>By Susan on <time>October 1, 2012</time></p>  
  <p>I love Central Park.</p>  
</div>  
<hr/>
```

Section 6 of 7

# THE BOX MODEL

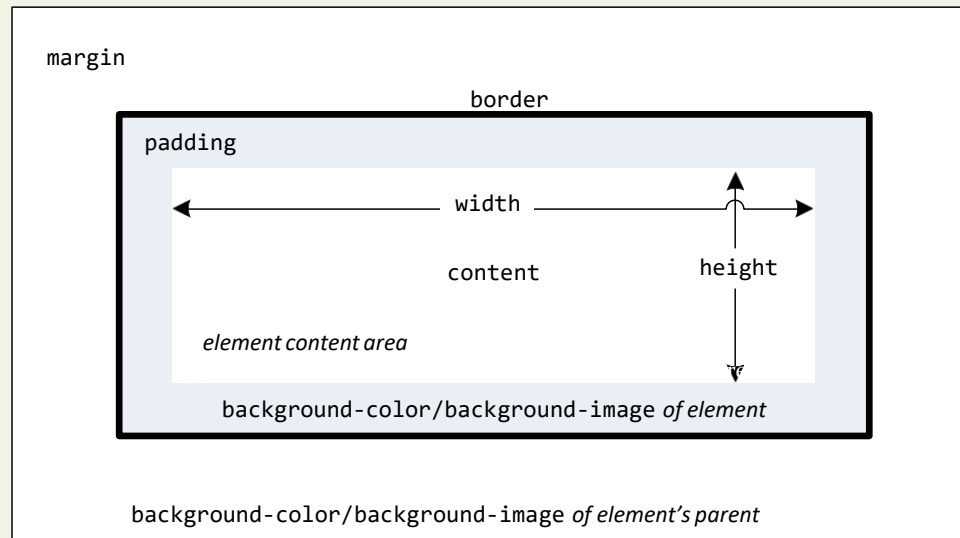
# The Box Model

Time to think inside the box

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins.

# The Box Model



# Developer Tools

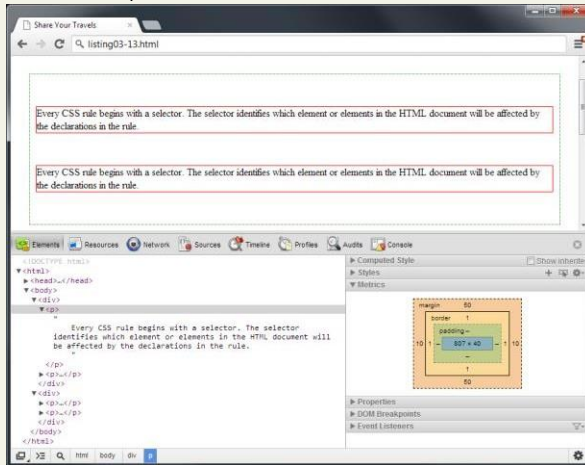
Help is on the way

Developer tools in current browsers make it significantly easier to examine and troubleshoot CSS.

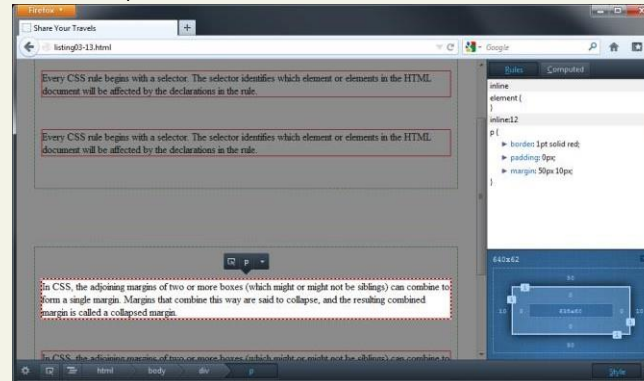
You can use the various browsers' CSS inspection tools to examine, for instance, the box values for a selected element.

# Developer Tools

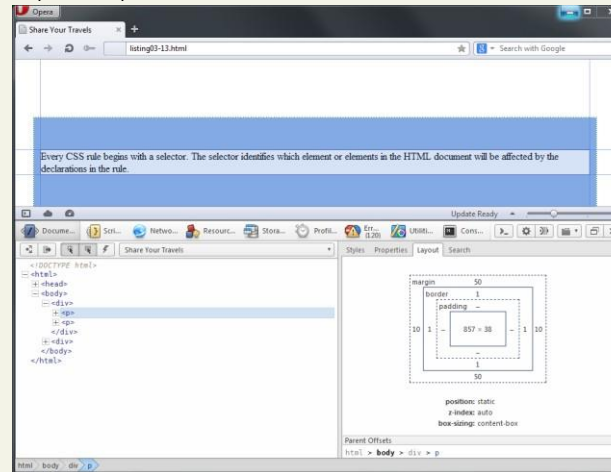
Chrome – Inspect Element



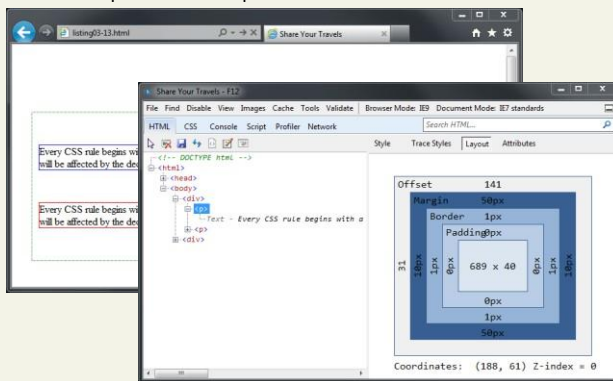
Firefox – Inspect



Opera – Inspect Element



Internet Explorer – Developer Tools





Section 7 of 7

# TEXT STYLING

# Text Properties

Two basic types

CSS provides two types of properties that affect text.

- **font properties** that affect the font and its appearance.
- **paragraph properties** that affect the layout and spacing of text, like alignment, line spacing, and indentation, regardless of which font is used.

# Font-Family

A few issues here

A word processor on a desktop machine can make use of any font that is installed on the computer; browsers are no different.

However, just because a given font is available on the web developer's computer, it does not mean that that same font will be available for all users who view the site.

For this reason, it is conventional to supply a so-called **web font stack**, that is, a series of alternate fonts to use in case the original font choice is not on the user's computer.

# Specifying the Font-Family

