CSS : Introduction

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development

Objectives

1. What is CSS?
2. CSS Syntax
3. Location of Styles
4. Selectors
5. The Cascade: How Styles Interact
6. The Box Model
7. CSS Text Styling

Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.

This separation can **improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting** by specifying the relevant CSS in a separate .css file which **reduces complexity and repetition** in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

What is CSS?

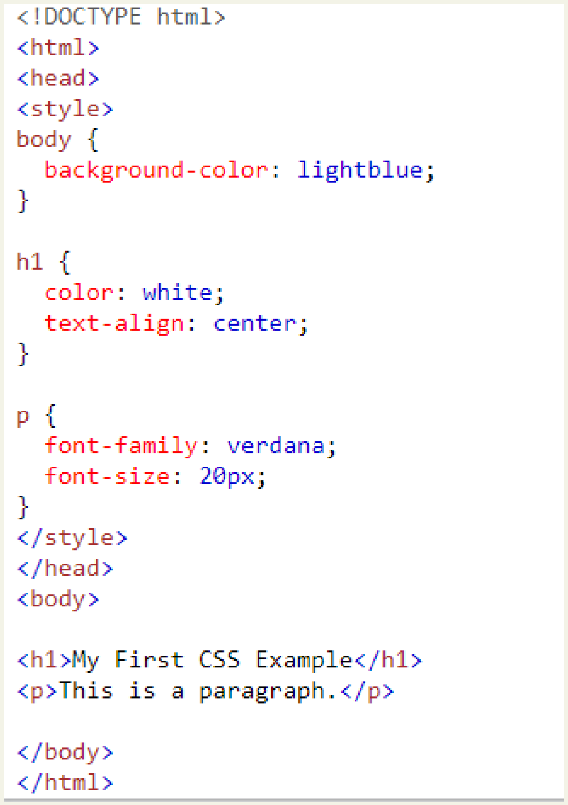
You be styling soon

CSS is a W3C standard for describing the **presentation (or appearance)** of HTML elements. With CSS, we can assign

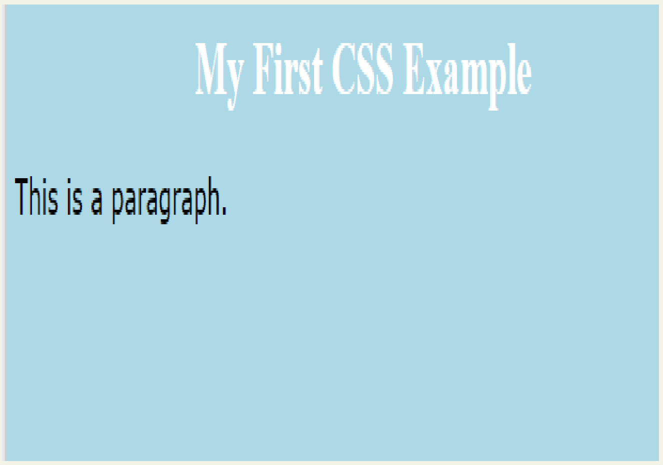
- font properties,
- colors,
- sizes,
- borders,
- background images,
- even the position of elements.

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.



```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



My First CSS Example

This is a paragraph.

Benefits of CSS

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

- **Improved control over formatting:**
- **Improved site maintainability:**
- **Improved accessibility:**
- **Improved page download speed:**
- **Improved output flexibility:**

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.
- CSS3 is the latest standard of CSS
- CSS3 became a W3C recommendation in June 1999 and builds on older versions CSS. It has divided into documentation is called as Modules and here each module having new extension features defined in CSS2.

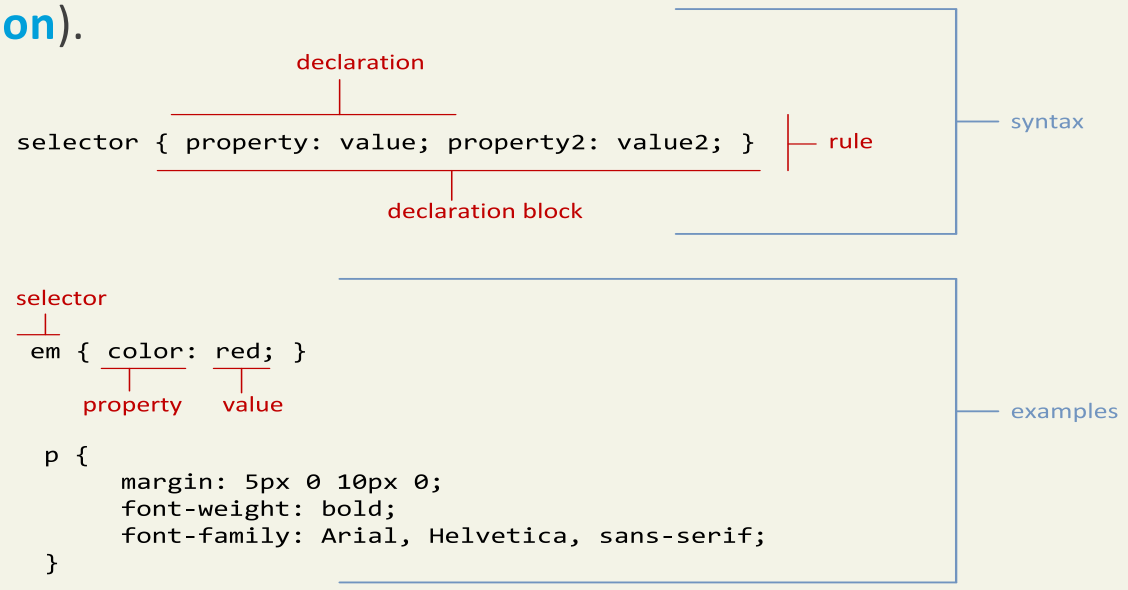
CSS SYNTAX

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**).



```
selector { property: value; property2: value2; }
```

```
em { color: red; }
```

```
p {
    margin: 5px 0 10px 0;
    font-weight: bold;
    font-family: Arial, Helvetica, sans-serif;
}
```

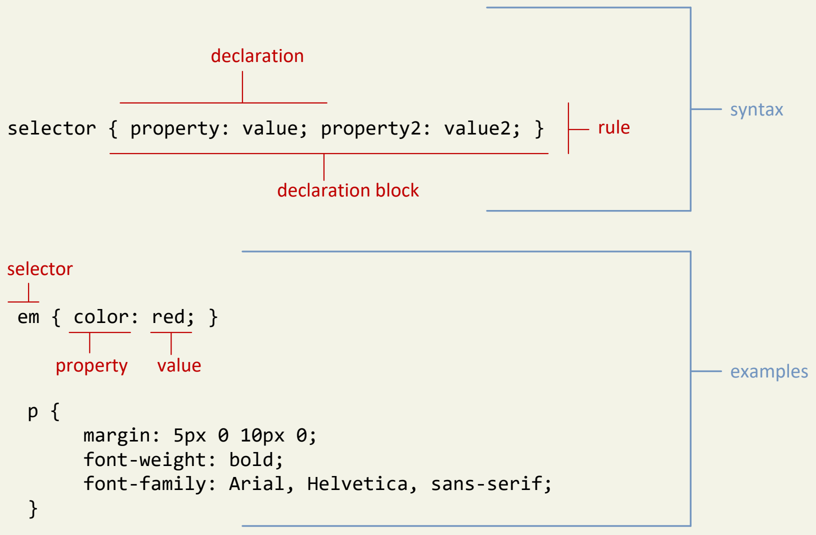
Selectors

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.

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 the style.



```
selector { property: value; property2: value2; }
```

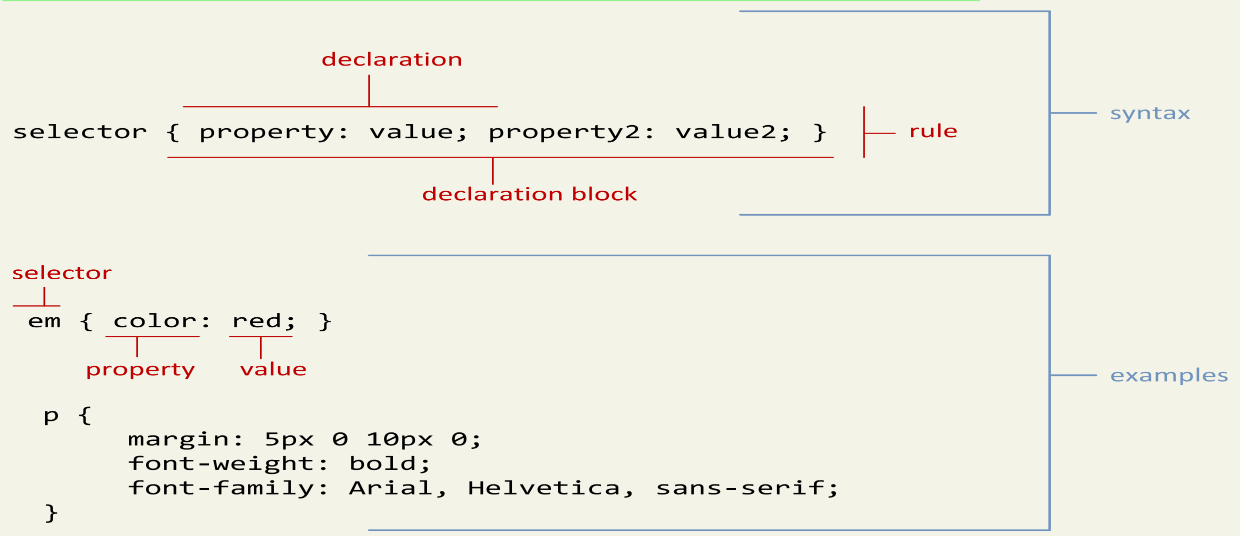
```
em { color: red; }
```

```
p {
     margin: 5px 0 10px 0;
     font-weight: bold;
     font-family: Arial, Helvetica, sans-serif;
}
```

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.



```
selector { property: value; property2: value2; }

em { color: red; }

p {
      margin: 5px 0 10px 0;
      font-weight: bold;
      font-family: Arial, Helvetica, sans-serif;
}
```

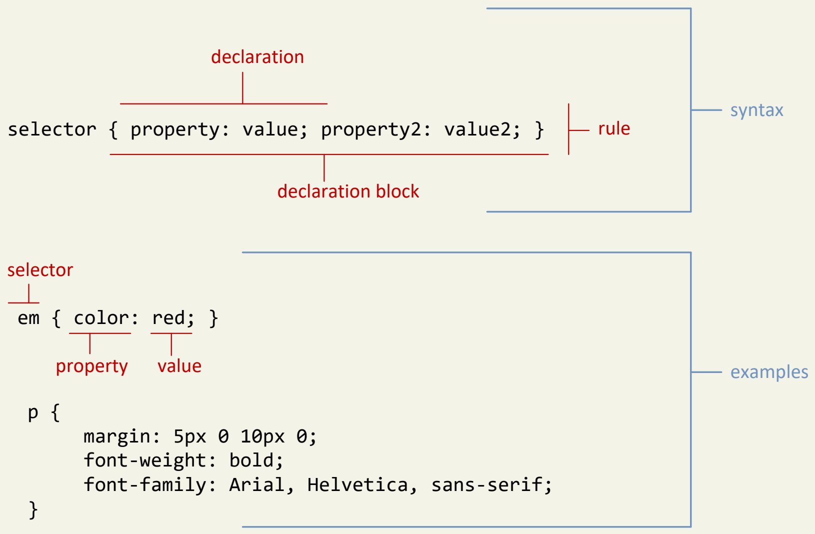
Properties

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.



```
selector { property: value; property2: value2; }
```

```
em { color: red; }
```

```
p {
     margin: 5px 0 10px 0;
     font-weight: bold;
     font-family: Arial, Helvetica, sans-serif;
}
```

Properties

Common 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 color
Borders	border border-color border-width border-style border-top border-top-color border-top-width etc

Properties

Common CSS properties continued.

Property Type	Property
Spacing	padding padding-bottom, padding-left, padding-right, padding-top margin margin-bottom, margin-left, margin-right, margin-top
Sizing	height max-height max-width min-height min-width width
Layout	bottom, left, right, top clear display float overflow position visibility z-index
Lists	list-style list-style-image list-style-type

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.
- Others are values such as length measurements, percentages, numbers without units, color values, and URLs.

Color Values

CSS supports a variety of different ways of describing color

Method	Description	Example
Name	Use one of 17 standard color names. CSS3 has 140 standard names.	color: red; color: hotpink; /* CSS3 only */

RGB	Uses three different numbers between 0 and 255 to describe the Red, Green, and Blue values for the color.	color: rgb(255,0,0);
		color: rgb(255,105,180);

Hexadecimal	Uses a six-digit hexadecimal number to describe the red, green, and blue value of the color; each of the three RGB values is between 0 and FF (which is 255 in decimal). Notice that the hexadecimal number is preceded by a hash or pound symbol (#).	color: #FF0000;
		color: #FF69B4;

RGBa	Allows you to add an alpha, or transparency, value. This allows a background color or image to "show through" the color. Transparency is a value between 0.0 (fully transparent) and 1.0 (fully opaque).	color: rgb(255,0,0, 0.5);

HSL	Allows you to specify a color using Hue Saturation and Light values. This is available only in CSS3. HSLA is also available as well.	color: hsl(0,100%,100%);
		color: hsl(330,59%,100%);

Units of Measurement

There are multiple ways of specifying a unit of measurement in CSS

Some of these are **relative units**, in that they are based on the value of something else, such as the size of a parent element.

Others are **absolute units**, in that they have a real-world size.

- Unless you are defining a style sheet for printing, it is recommended to avoid using absolute units.
- Pixels are perhaps the one popular exception (though as we shall see later there are also good reasons for avoiding the pixel unit).

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 em, which may be different for each element, the rem 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)

Absolute Units

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

Comments in CSS

It is often helpful to add comments to your style sheets. Comments take the form:

/* comment goes here */

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. This option is available in a browser usually in its accessibility options area.

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

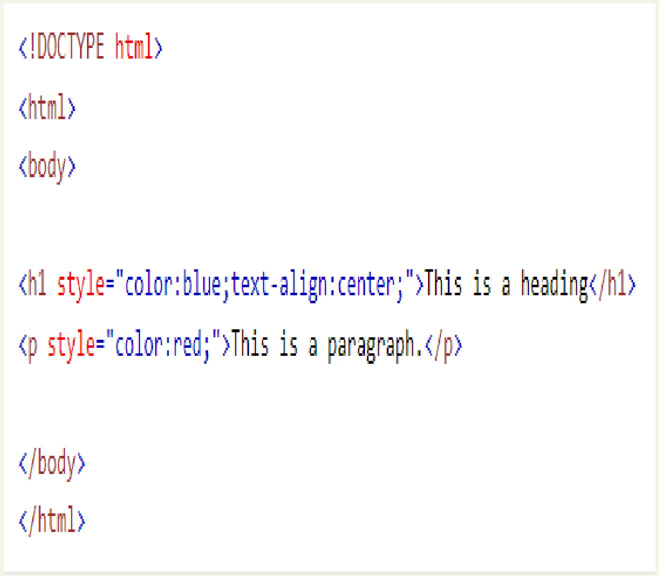
You can combine all 3!

Inline Styles

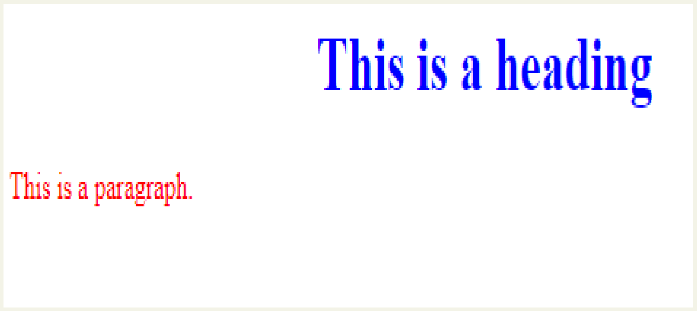
Style rules placed within an HTML element via the style attribute

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.



```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```



This is a heading

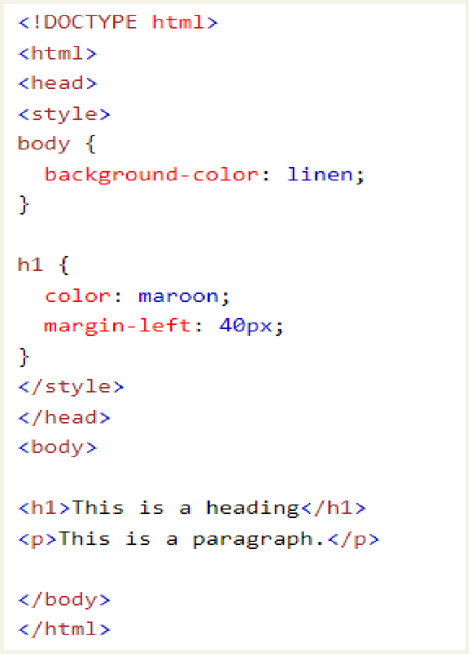
This is a paragraph.

Embedded Style Sheet

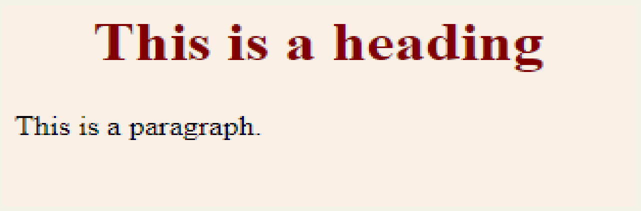
Style rules placed within the <style> element inside the <head> element

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.



```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

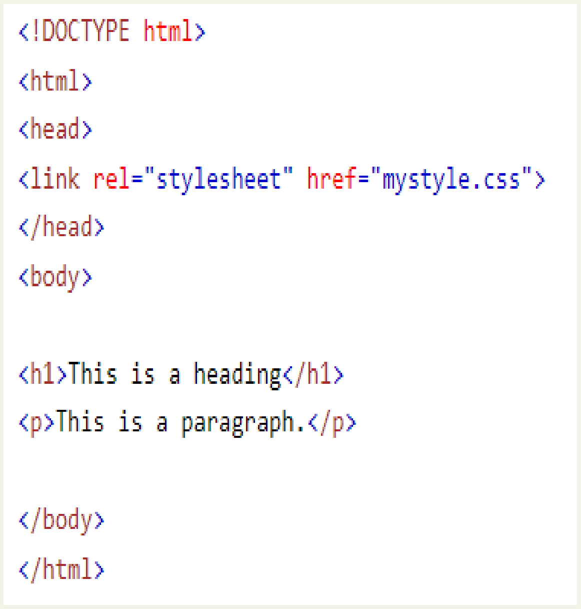
This is a paragraph.

External Style Sheet

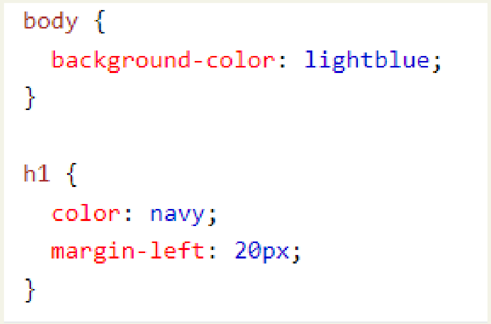
Style rules placed within a external text file with the .css extension

With an external style sheet, you can change the look of an entire website by changing just one file!

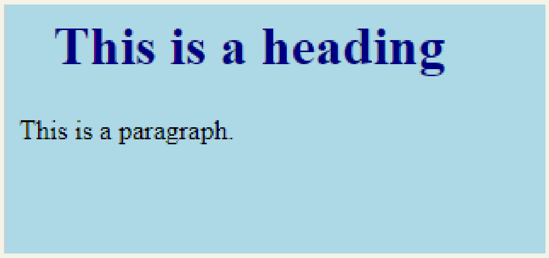
Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.



```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```



This is a heading

This is a paragraph.

SELECTORS

Selectors

Things that make your life easier

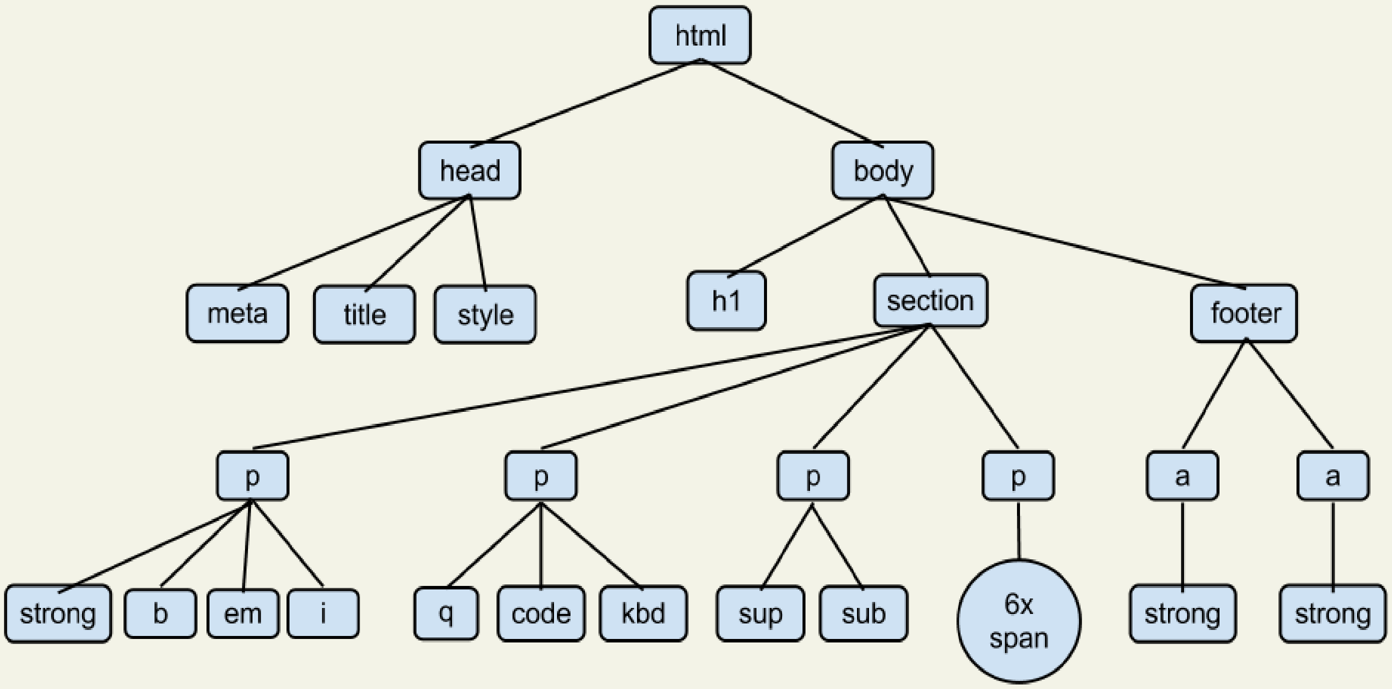
When defining CSS rules, you will need to first need to use a **selector** to tell the browser which elements will be affected.

CSS selectors allow you to select

- individual elements
- multiple HTML elements,
- elements that belong together in some way, or
- elements that are positioned in specific ways in the document hierarchy.

There are a number of different selector types.

HTML document tree

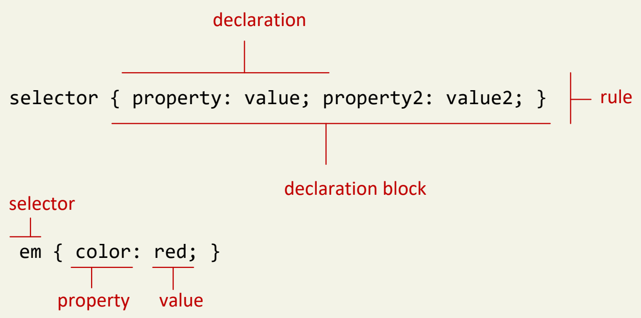


Element Selectors

Selects all instances of a given HTML element

The element selector selects HTML elements based on the element name.

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

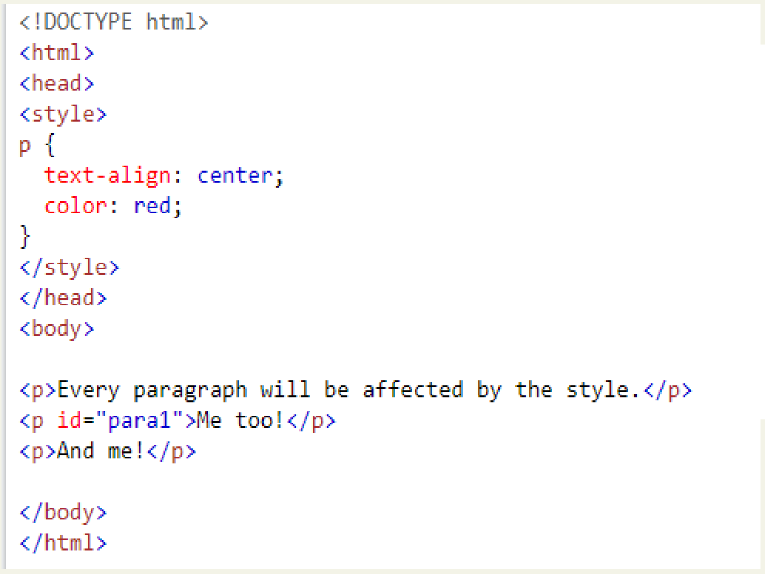


```
selector { property: value; property2: value2; }
```

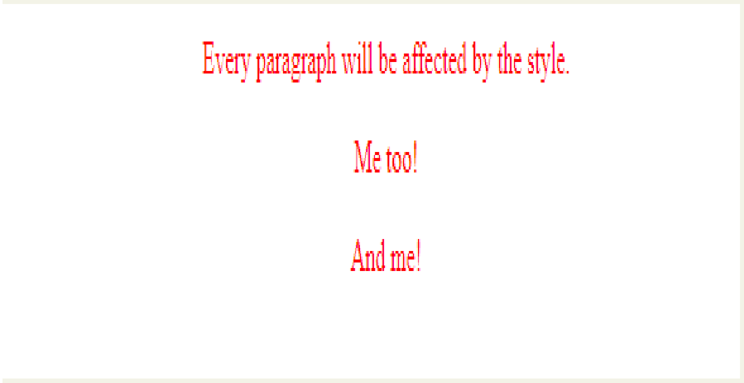
```
em { color: red; }
```

```
p {
     margin: 5px 0 10px 0;
     font-weight: bold;
     font-family: Arial, Helvetica, sans-serif;
}
```

Element Selectors



```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```



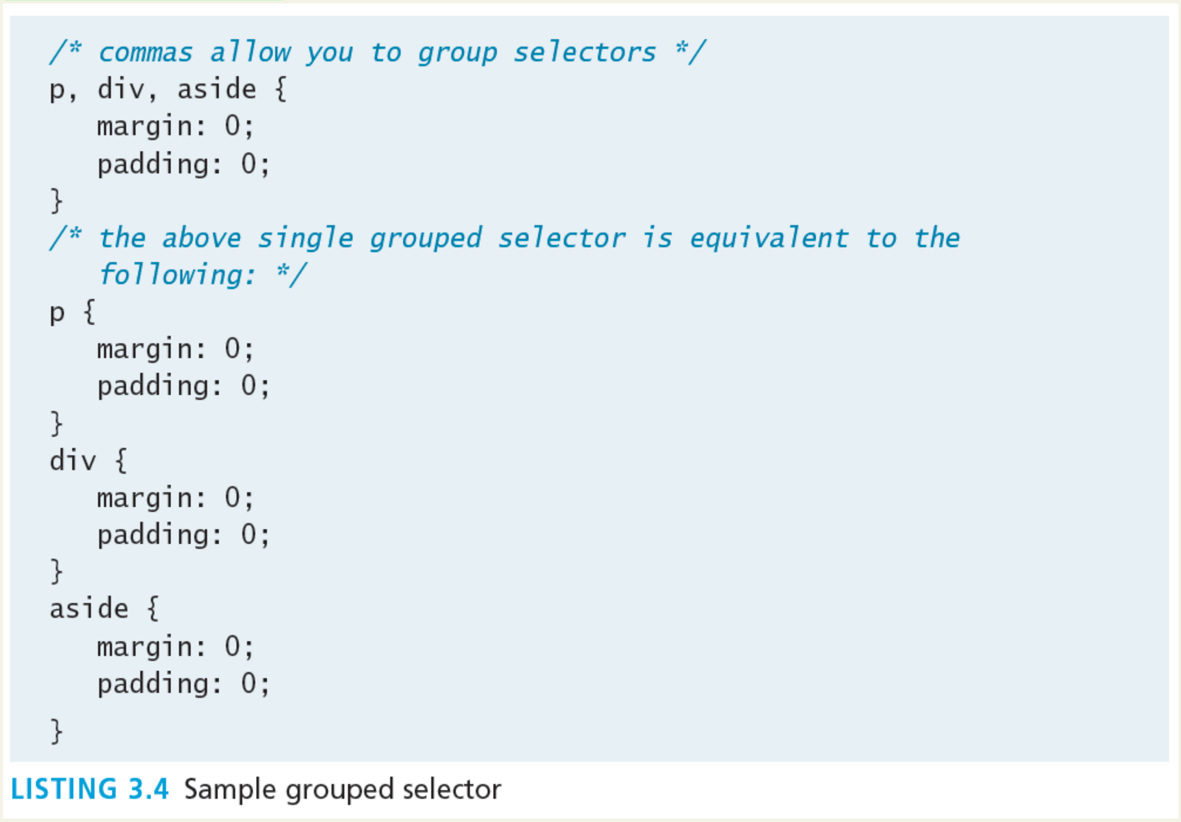
Every paragraph will be affected by the style.

Me too!

And me!

Grouped Selectors

The grouping selector selects all the HTML elements with the same style definitions.

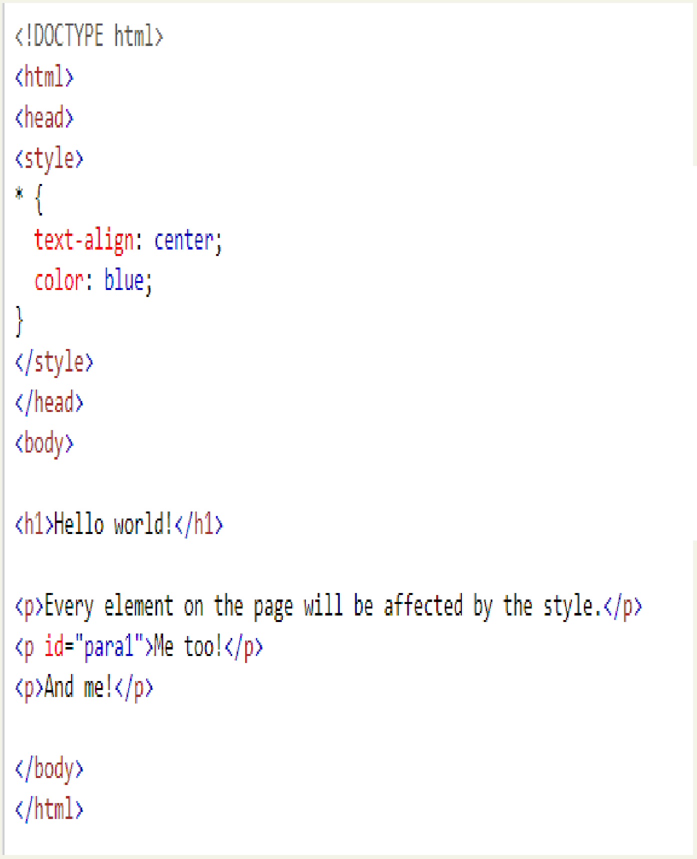


```
/* commas allow you to group selectors */
p, div, aside {
   margin: 0;
   padding: 0;
}
/* the above single grouped selector is equivalent to the
   following: */
p {
   margin: 0;
   padding: 0;
}
div {
   margin: 0;
   padding: 0;
}
aside {
   margin: 0;
   padding: 0;
}
```

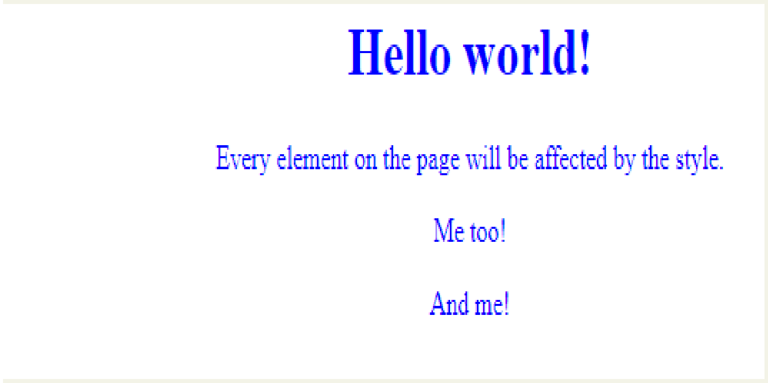
LISTING 3.4 Sample grouped selector

universal selector

The universal selector (*) selects all HTML elements on the page.



```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>

<h1>Hello world!</h1>

<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```



```
Hello world!

Every element on the page will be affected by the style.

Me too!

And me!
```

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.

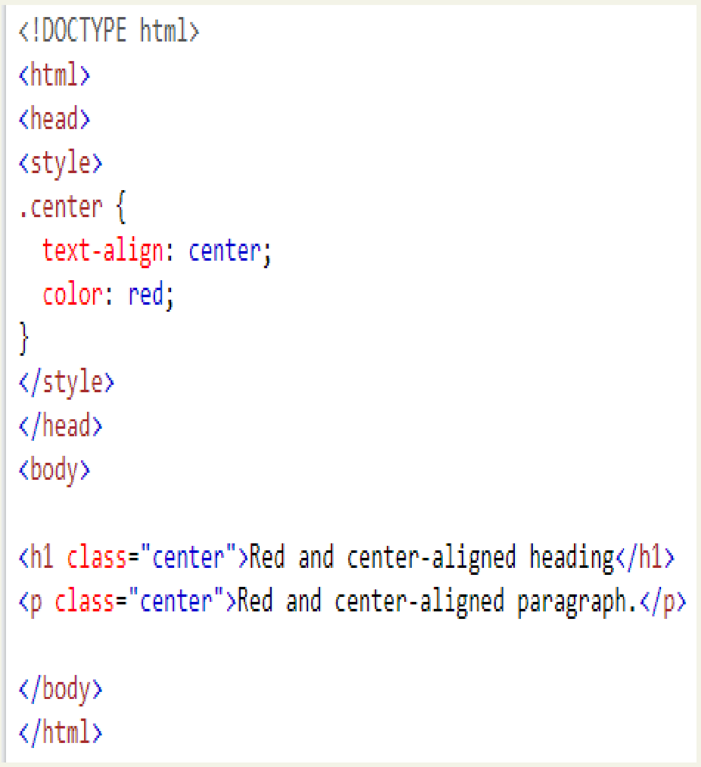
Class Selectors

Simultaneously target different HTML elements

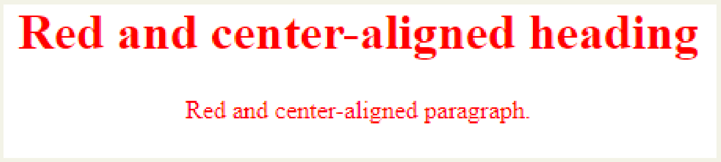
A **class selector** allows you to simultaneously target different HTML elements regardless of their position in the document tree.

To select elements with a specific class, write a period (.) character, followed by the class name.

Class Selectors



```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```



Red and center-aligned heading

Red and center-aligned paragraph.

Class Selectors

- You can also specify that only specific HTML elements should be affected by a class. In the example below, only <p> elements with class="center" will be center-aligned:

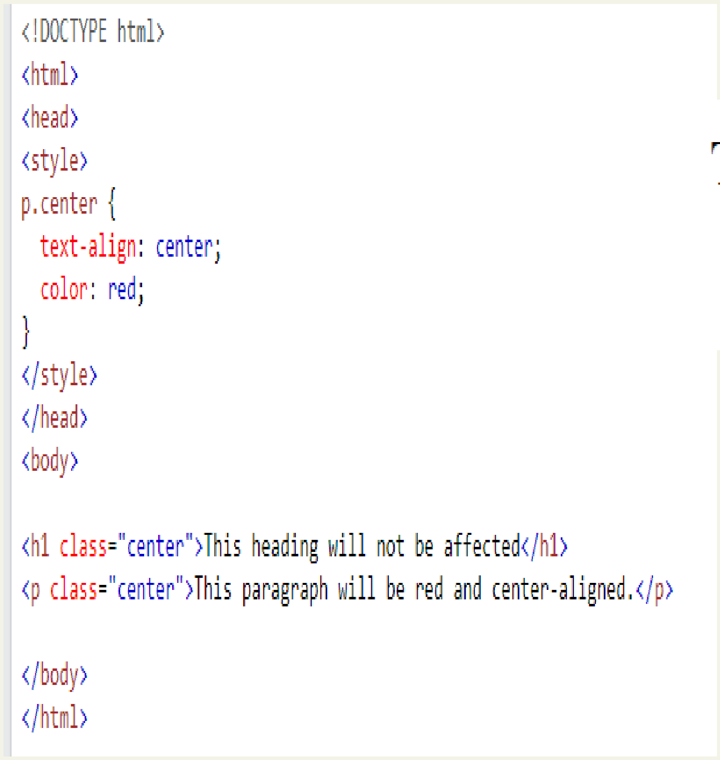
```
p.center {text-align: center;}
```

- HTML elements can also refer to more than one class. In the example below, the <p> element will be styled according to class="center" and to class="large":

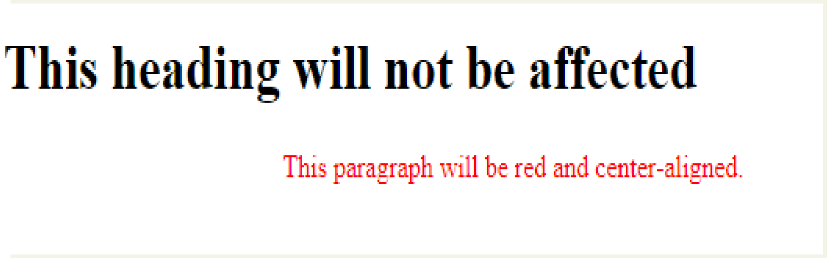
```
<p class="center large">This paragraph refers
to two classes.</p>
```

http://www.w3schools.com/css/css_syntax.asp

Class Selectors



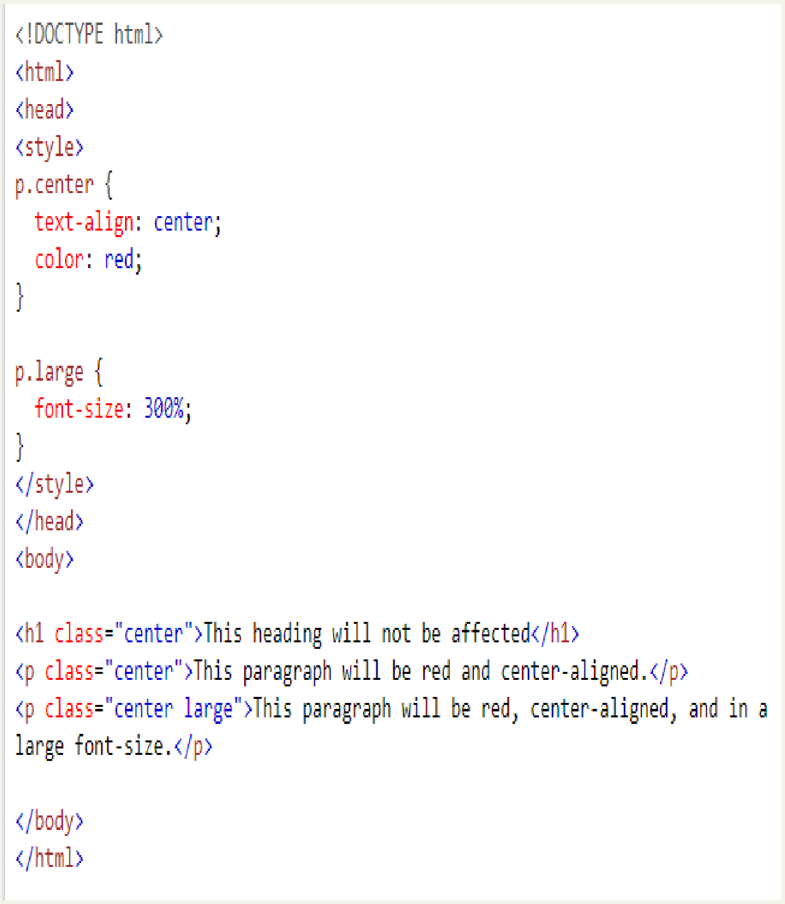
```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>

</body>
</html>
```



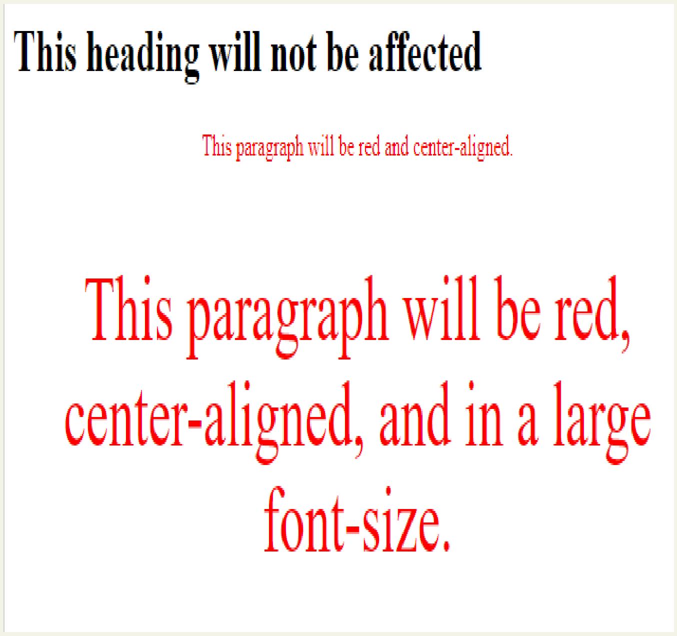
This heading will not be affected

This paragraph will be red and center-aligned.

Class Selectors



```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}

p.large {
  font-size: 300%;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a
large font-size.</p>

</body>
</html>
```



This heading will not be affected

This paragraph will be red and center-aligned.

This paragraph will be red, center-aligned, and in a large font-size.

Id Selectors

Target a specific element by its id attribute

The id selector uses the id attribute of an HTML element to select a specific element.

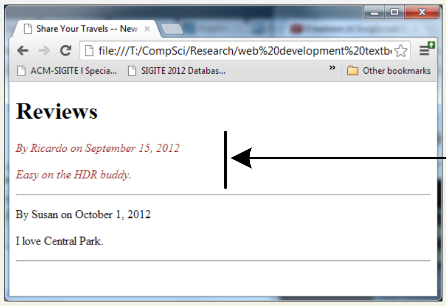
The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

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>
```


Share Your Travels -- New
file:///T:/CompSci/Research/web%20development%20textb
ACM-SIGITE I Specia...
SIGITE 2012 Databas...
Other bookmarks

Reviews

By Ricardo on September 15, 2012

Easy on the HDR buddy.

By Susan on October 1, 2012

I love Central Park.

```
#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.

Attribute Selectors

Selecting via presence of element attribute or by the value of an attribute

An **attribute selector** provides a way to select HTML elements by either the presence of an element attribute or by the value of an attribute.

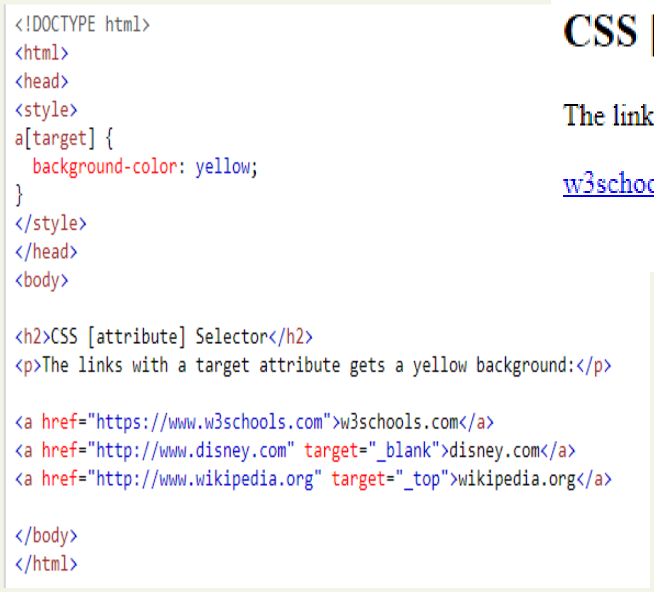
This can be a very powerful technique, but because of uneven support by some of the browsers, not all web authors have used them.

Attribute selectors can be a very helpful technique in the styling of hyperlinks and images.

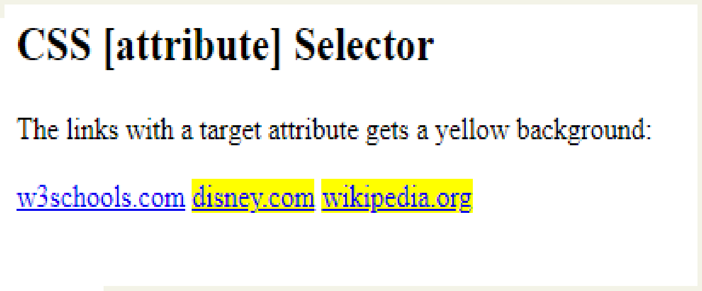
Attribute Selectors

The [attribute] selector is used to select elements with a specified attribute.

The following example selects all <a> elements with a target attribute:



```
<!DOCTYPE html>
<html>
<head>
<style>
a[target] {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute] Selector</h2>
<p>The links with a target attribute gets a yellow background:</p>

<a href="https://www.w3schools.com">w3schools.com</a>
<a href="http://www.disney.com" target="_blank">disney.com</a>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>

</body>
</html>
```



CSS [attribute] Selector

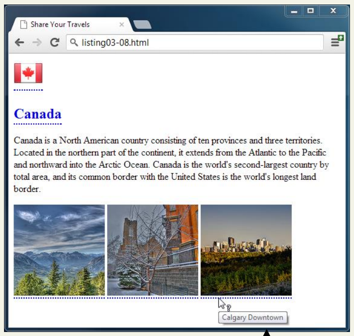
The links with a target attribute gets a yellow background:

w3schools.com disney.com wikipedia.org

Attribute Selectors

```
[title] {
   cursor: help;
   padding-bottom: 3px;
   border-bottom: 2px dotted blue;
   text-decoration: none;
}
```

```
<head lang="en">
   <meta charset="utf-8">
   <title>Share Your Travels</title>
     <style>
          [title] {
               cursor: help;
               padding-bottom: 3px;
               border-bottom: 2px dotted blue;
               text-decoration: none;
          }
     </style>
</head>
<body>
   <div>
      <img src="images/flags/CA.png" title="Canada Flag" />
       <h2><a href="countries.php?id=CA" title="see posts from Canada">
           Canada</a>
       </h2>
       <p>Canada is a North American country consisting of … </p>
       <div>
          <img src="images/square/6114907897.jpg" title="At top of Sulpher Mountain">
          <img src="images/square/6592317633.jpg" title="Grace Presbyterian Church">
          <img src="images/square/6592914823.jpg" title="Calgary Downtown">
       </div>
   </div>
</body>
```



Canada

Canada is a North American country consisting of ten provinces and three territories. Located in the northern part of the continent, it extends from the Atlantic to the Pacific and northward into the Arctic Ocean. Canada is the world's second-largest country by total area, and its common border with the United States is the world's longest land border.

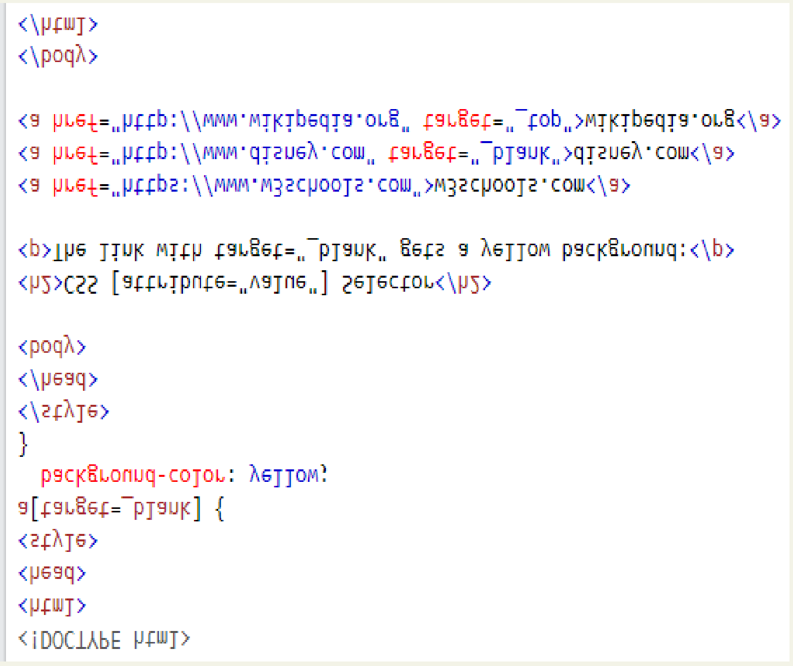
Selector	Matches	Example
[]	A specific attribute.	`[title]` Matches any element with a title attribute
[=]	A specific attribute with a specific value.	`a[title="posts from this country"]` Matches any `<a>` element whose title attribute is exactly `"posts from this country"`
[~=]	A specific attribute whose value matches at least one of the words in a space-delimited list of words.	`[title~="Countries"]` Matches any `title` attribute that contains the word `"Countries"`
[^=]	A specific attribute whose value begins with a specified value.	`a[href^="mailto"]` Matches any `<a>` element whose `href` attribute begins with `"mailto"`
[*=]	A specific attribute whose value contains a substring.	`img[src*="flag"]` Matches any `<img>` element whose `src` attribute contains somewhere within it the text `"flag"`
[$=]	A specific attribute whose value ends with a specified value.	`a[href$=".pdf"]` Matches any `<a>` element whose `href` attribute ends with the text `".pdf"`

TABLE 3.4 Attribute Selectors

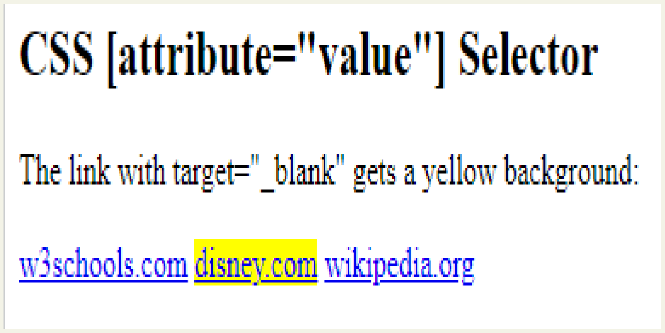
[attribute="value"] Selector

The [attribute="value"] selector is used to select elements with a specified attribute and value.

The following example selects all <a> elements with a target="_blank" attribute:



```
<!DOCTYPE html>
<html>
<head>
<style>
a[target=_blank] {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute="value"] Selector</h2>
<p>The link with target="_blank" gets a yellow background:</p>

<a href="https://www.w3schools.com">w3schools.com</a>
<a href="http://www.disney.com" target="_blank">disney.com</a>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>

</body>
</html>
```



CSS [attribute="value"] Selector

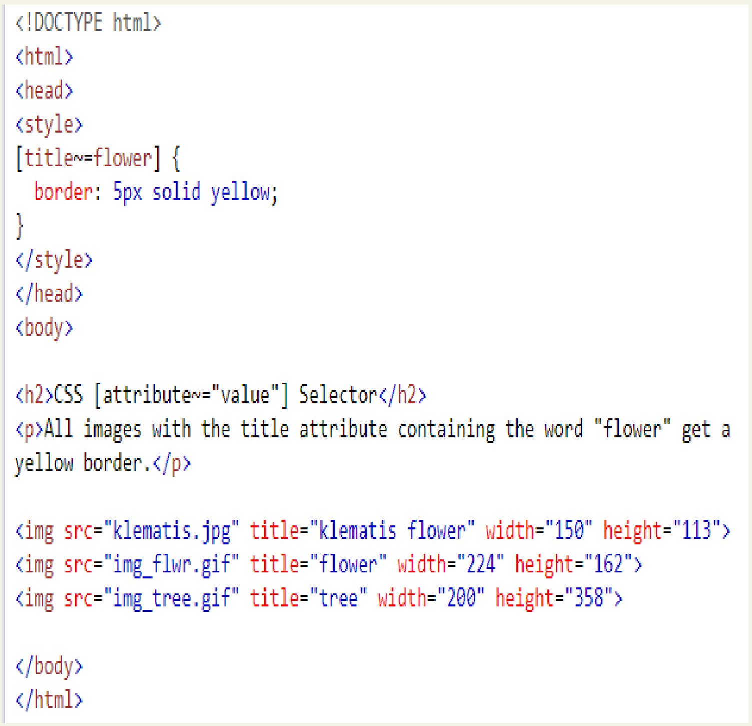
The link with target="_blank" gets a yellow background:

w3schools.com disney.com wikipedia.org

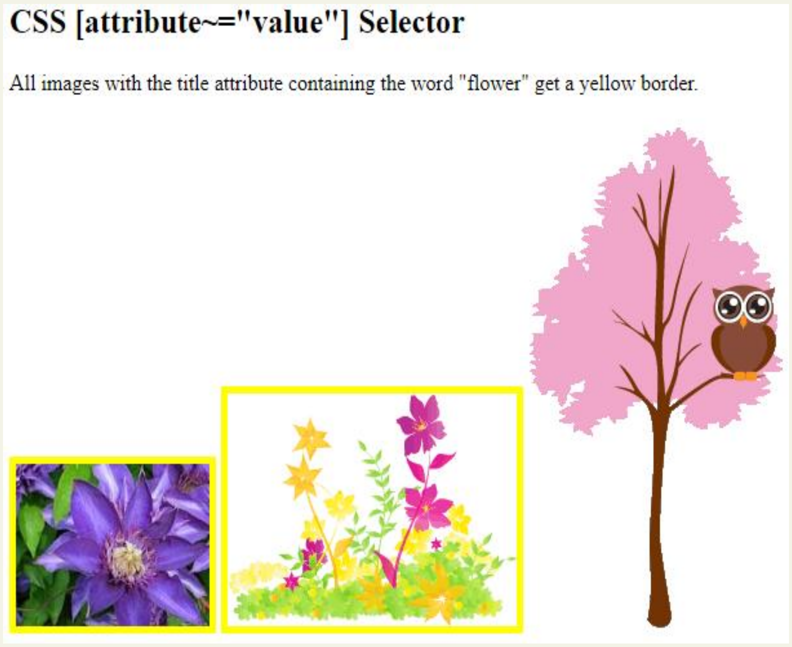
CSS [attribute~="value"] Selector

The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.

The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":



```
<!DOCTYPE html>
<html>
<head>
<style>
[title~=flower] {
  border: 5px solid yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute~="value"] Selector</h2>
<p>All images with the title attribute containing the word "flower" get a
yellow border.</p>

<img src="klematis.jpg" title="klematis flower" width="150" height="113">
<img src="img_flwr.gif" title="flower" width="224" height="162">
<img src="img_tree.gif" title="tree" width="200" height="358">

</body>
</html>
```



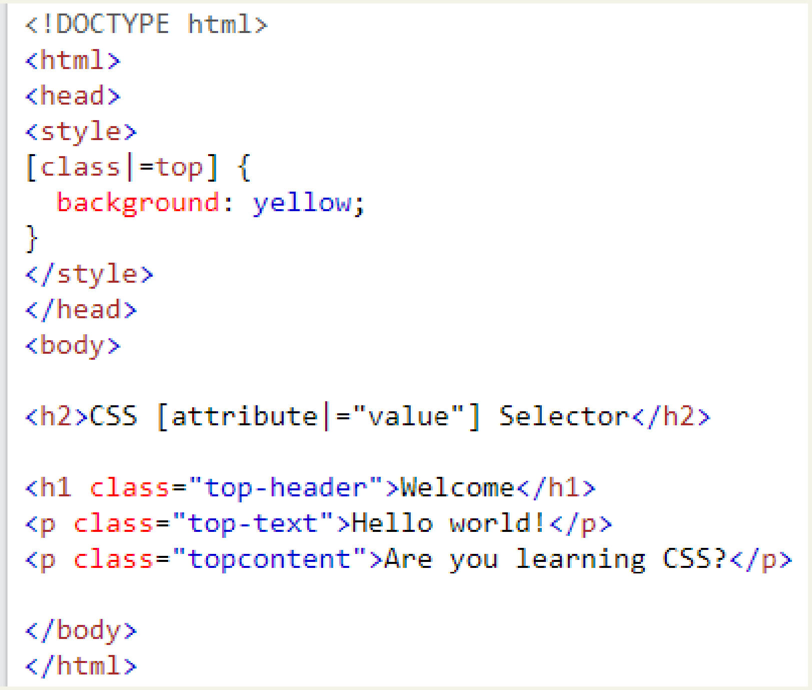
CSS [attribute~="value"] Selector

All images with the title attribute containing the word "flower" get a yellow border.

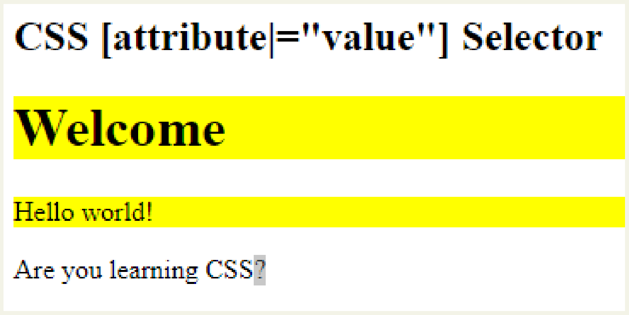
CSS [attribute|="value"] Selector

The [attribute|="value"] selector is used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-).

Note: The value has to be a whole word, either alone, like class="top", or followed by a hyphen(-), like class="top-text".



```
<!DOCTYPE html>
<html>
<head>
<style>
[class|=top] {
  background: yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute|="value"] Selector</h2>

<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>

</body>
</html>
```



CSS [attribute|="value"] Selector

Welcome

Hello world!

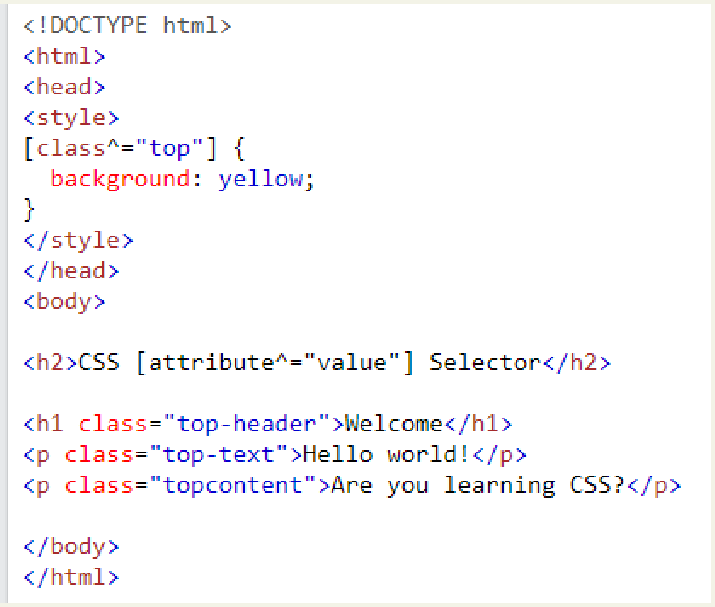
Are you learning CSS?

CSS [attribute^="value"] Selector

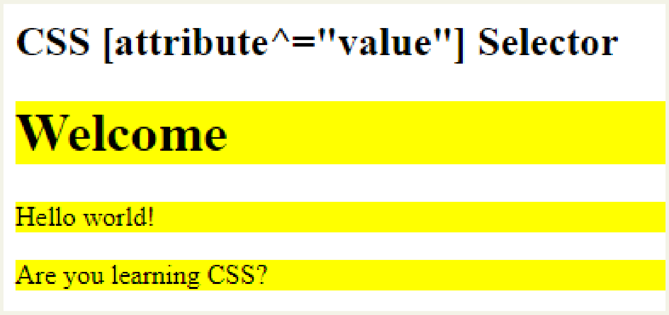
The [attribute^="value"] selector is used to select elements with the specified attribute, whose value starts with the specified value.

The following example selects all elements with a class attribute value that starts with "top":

Note: The value does not have to be a whole word!



```
<!DOCTYPE html>
<html>
<head>
<style>
[class^="top"] {
  background: yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute^="value"] Selector</h2>

<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>

</body>
</html>
```



CSS [attribute^="value"] Selector

Welcome

Hello world!

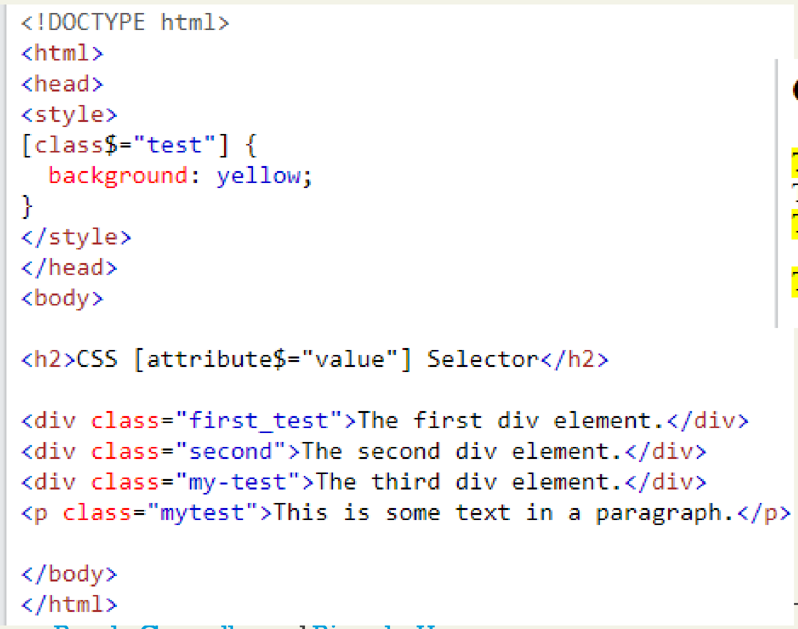
Are you learning CSS?

CSS [attribute$="value"] Selector

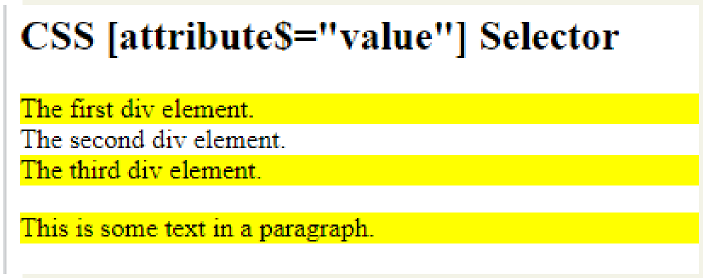
The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.

The following example selects all elements with a class attribute value that ends with "test":

Note: The value does not have to be a whole word!



```
<!DOCTYPE html>
<html>
<head>
<style>
[class$="test"] {
  background: yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute$="value"] Selector</h2>

<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>

</body>
</html>
```



CSS [attribute$="value"] Selector

The first div element.
The second div element.
The third div element.

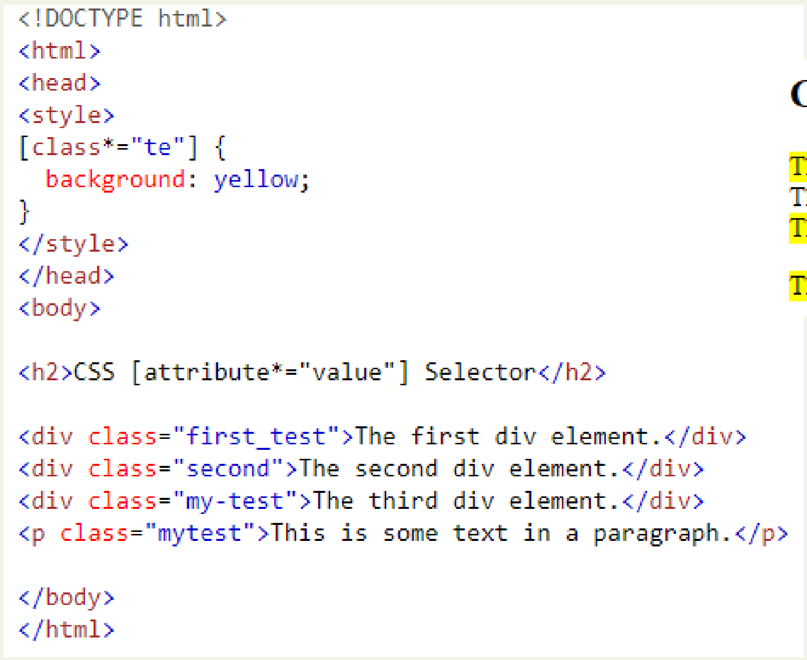
This is some text in a paragraph.

CSS [attribute*="value"] Selector

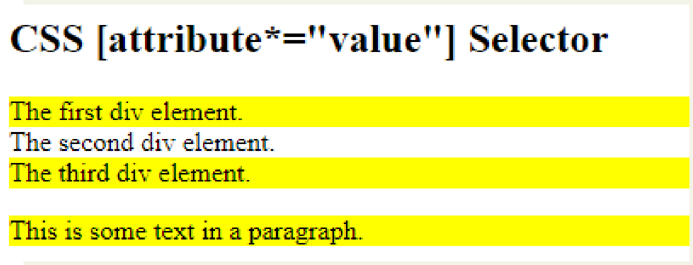
The [attribute*="value"] selector is used to select elements whose attribute value contains a specified value.

The following example selects all elements with a class attribute value that contains "te":

Note: The value does not have to be a whole word!



```
<!DOCTYPE html>
<html>
<head>
<style>
[class*="te"] {
  background: yellow;
}
</style>
</head>
<body>

<h2>CSS [attribute*="value"] Selector</h2>

<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>

</body>
</html>
```



CSS [attribute*="value"] Selector

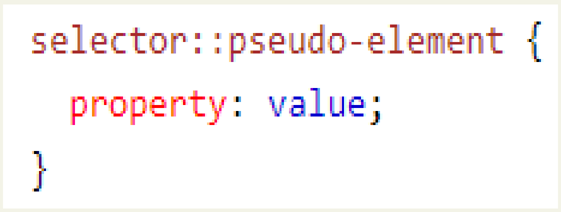
The first div element.
The second div element.
The third div element.

This is some text in a paragraph.

CSS Pseudo-element Selector

A CSS pseudo-element is used to style specified parts of an element.

Syntax:

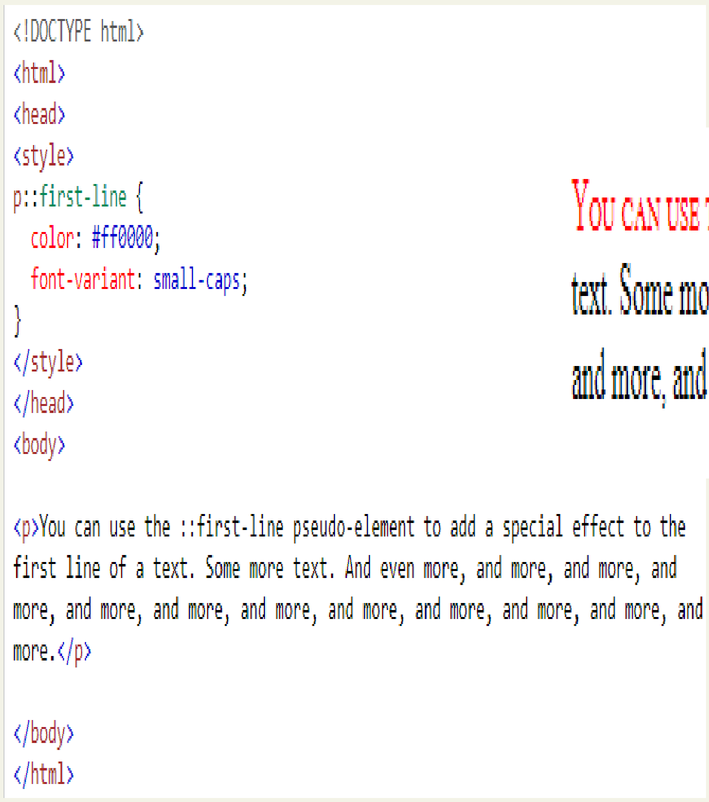


```
selector::pseudo-element {
  property: value;
}
```

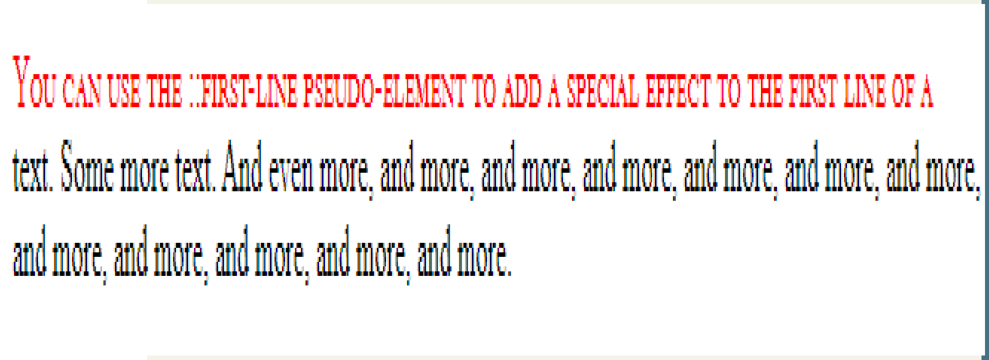
The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:



```
<!DOCTYPE html>
<html>
<head>
<style>
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}
</style>
</head>
<body>

<p>You can use the ::first-line pseudo-element to add a special effect to the
first line of a text. Some more text. And even more, and more, and more, and
more, and more, and more, and more, and more, and more, and more, and more, and
more.</p>

</body>
</html>
```



```
YOU CAN USE THE ::FIRST-LINE PSEUDO-ELEMENT TO ADD A SPECIAL EFFECT TO THE FIRST LINE OF A
text. Some more text. And even more, and more, and more, and more, and more, and more, and more,
and more, and more, and more, and more, and more.
```

The ::first-line Pseudo-element

Note: The ::first-line pseudo-element can only be applied to block-level elements.

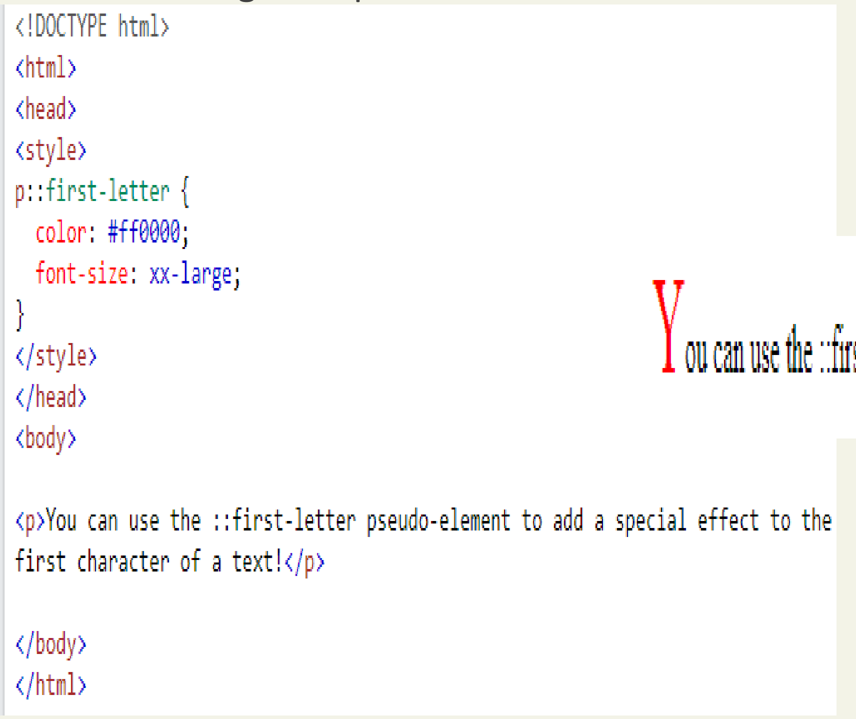
The following properties apply to the ::first-line pseudo-element:

- **font properties**
- **color properties**
- **background properties**
- **word-spacing**
- **letter-spacing**
- **text-decoration**
- **vertical-align**
- **text-transform**
- **line-height**
- **clear**

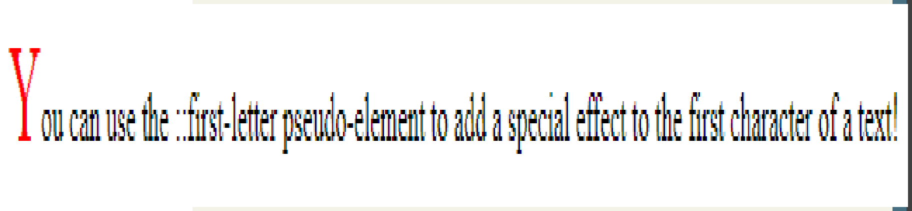
The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all <p> elements:



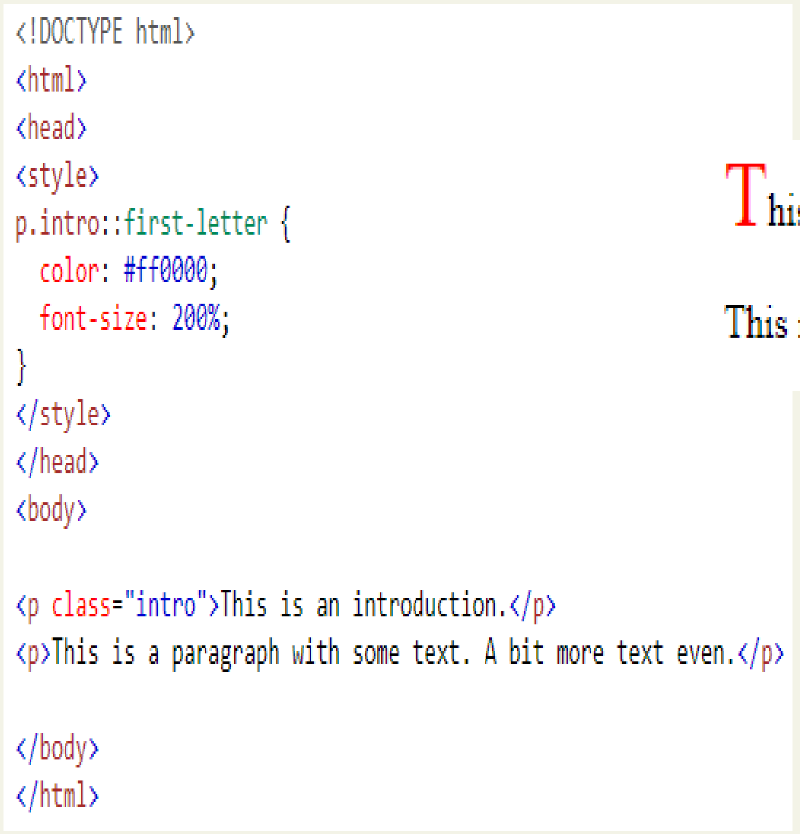
```
<!DOCTYPE html>
<html>
<head>
<style>
p::first-letter {
  color: #ff0000;
  font-size: xx-large;
}
</style>
</head>
<body>

<p>You can use the ::first-letter pseudo-element to add a special effect to the
first character of a text!</p>

</body>
</html>
```



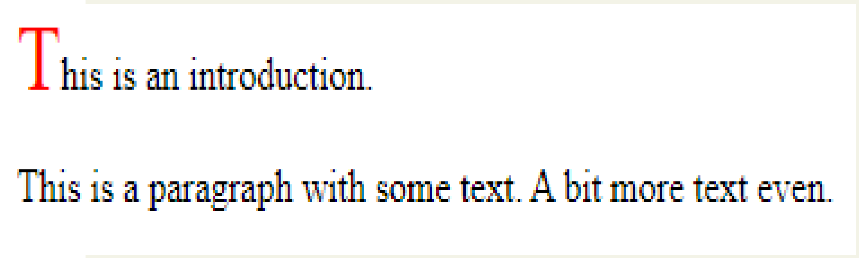
You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!

Pseudo class selectors

Pseudo-elements can be combined with HTML classes:



```
<!DOCTYPE html>
<html>
<head>
<style>
p.intro::first-letter {
  color: #ff0000;
  font-size: 200%;
}
</style>
</head>
<body>

<p class="intro">This is an introduction.</p>
<p>This is a paragraph with some text. A bit more text even.</p>

</body>
</html>
```



This is an introduction.

This is a paragraph with some text. A bit more text even.

Contextual Selectors

Select elements based on their ancestors, descendants, or siblings

A **contextual selector** (in CSS3 also called **combinators**) allows you to select elements based on their ancestors, descendants, or siblings.

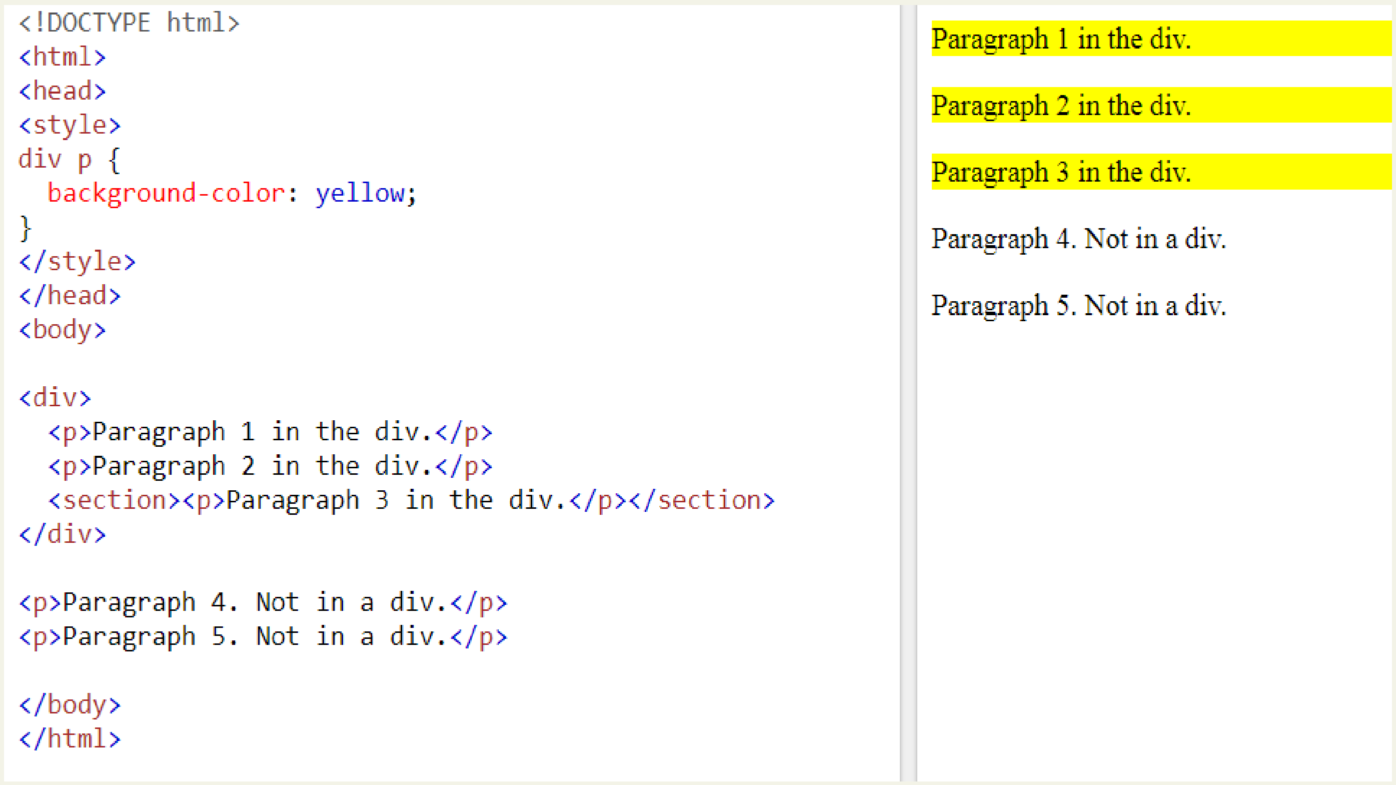
That is, it selects elements based on their context or their relation to other elements in the document tree.

Contextual Selectors

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

Descendant Selector :

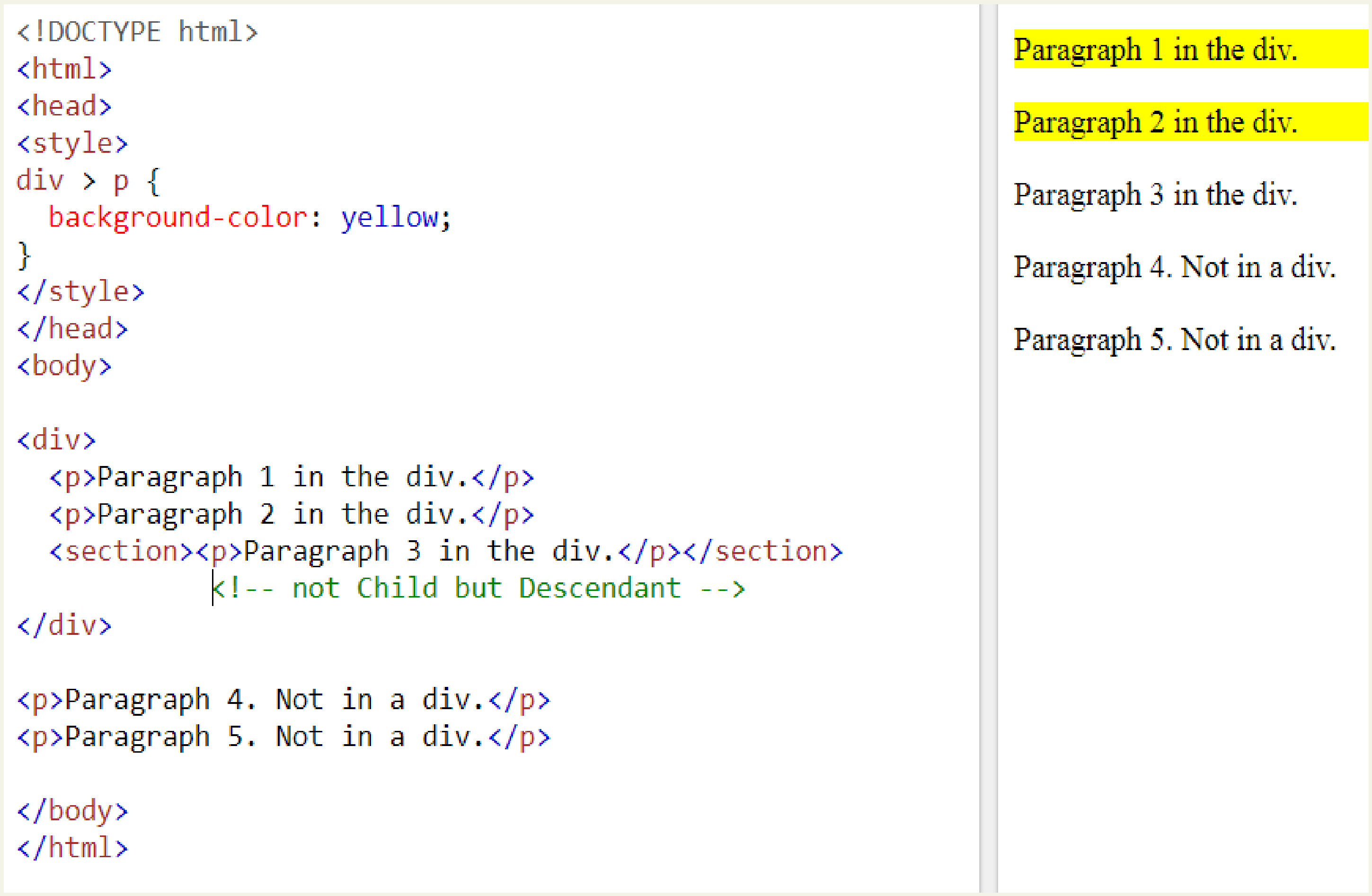
The descendant selector matches all elements that are descendants of a specified element. The following example selects all <p> elements inside <div> elements:



```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

```
Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.
```

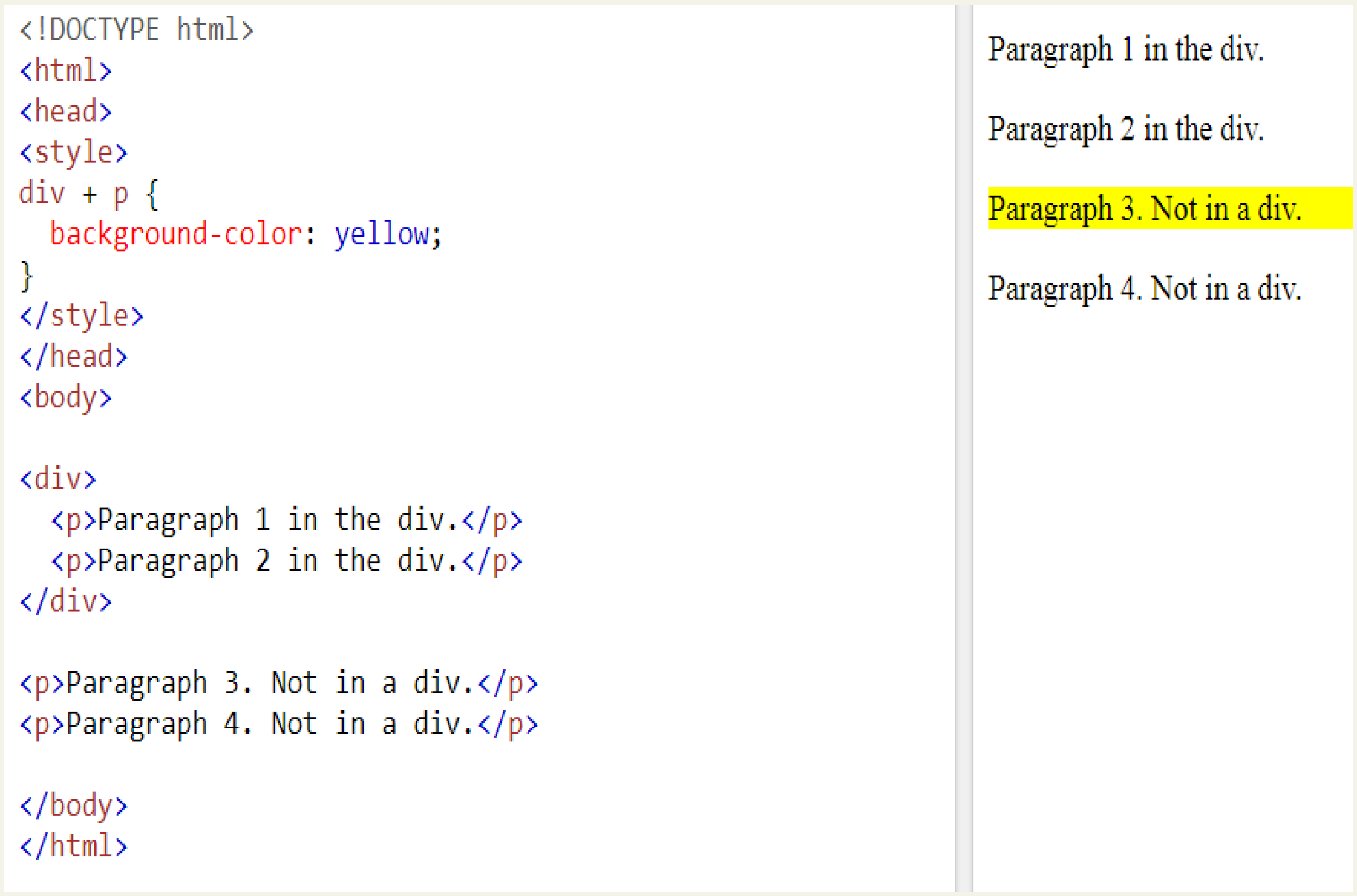
Child Selector



```
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
            <!-- not Child but Descendant -->
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

```
Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.
```

Adjacent Sibling Selector



```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. Not in a div.</p>
<p>Paragraph 4. Not in a div.</p>

</body>
</html>
```

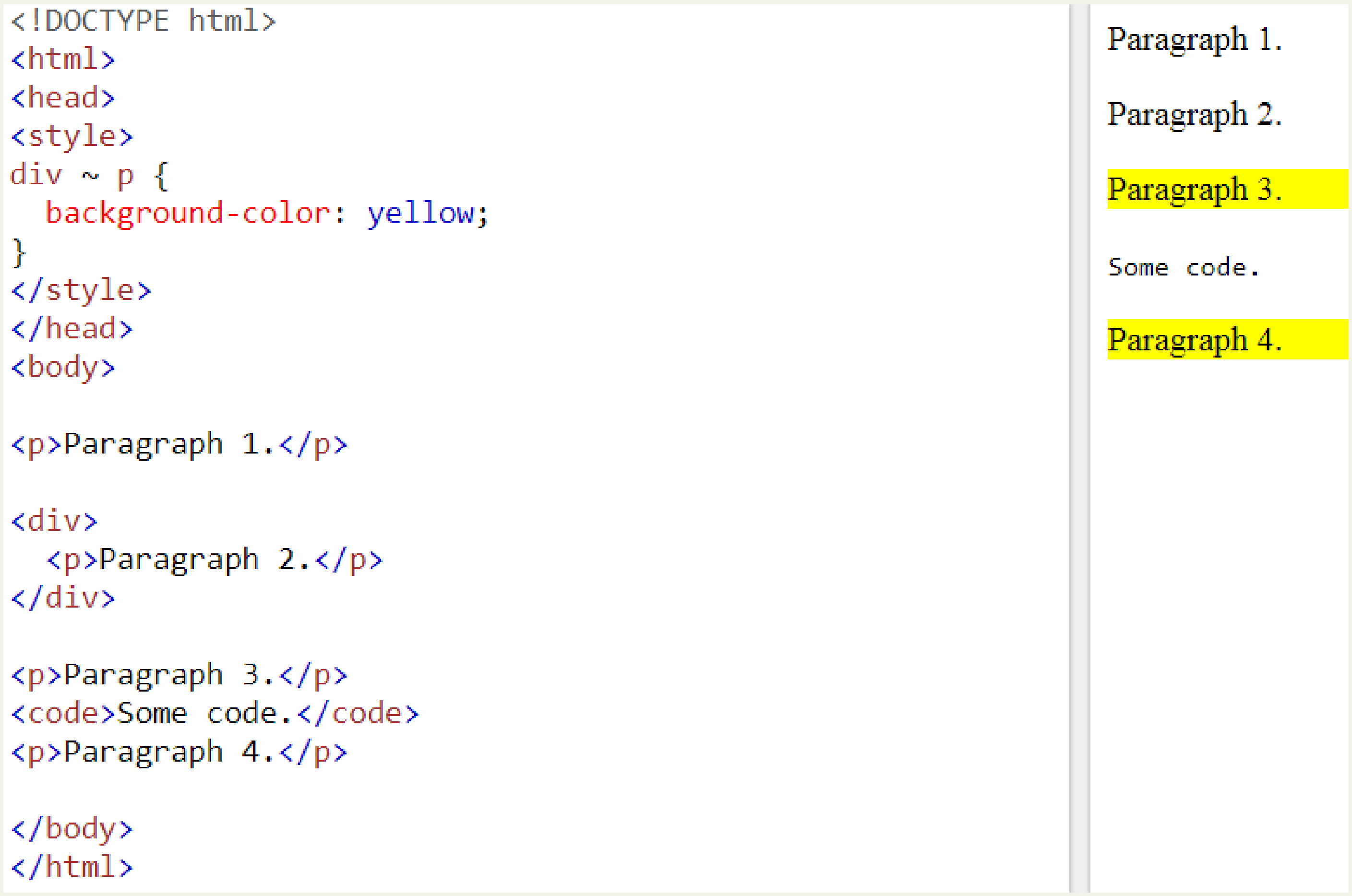
Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

General Sibling Selector



```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>

<p>Paragraph 1.</p>

<div>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>

</body>
</html>
```

```
Paragraph 1.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.
```

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 style sheets 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.

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

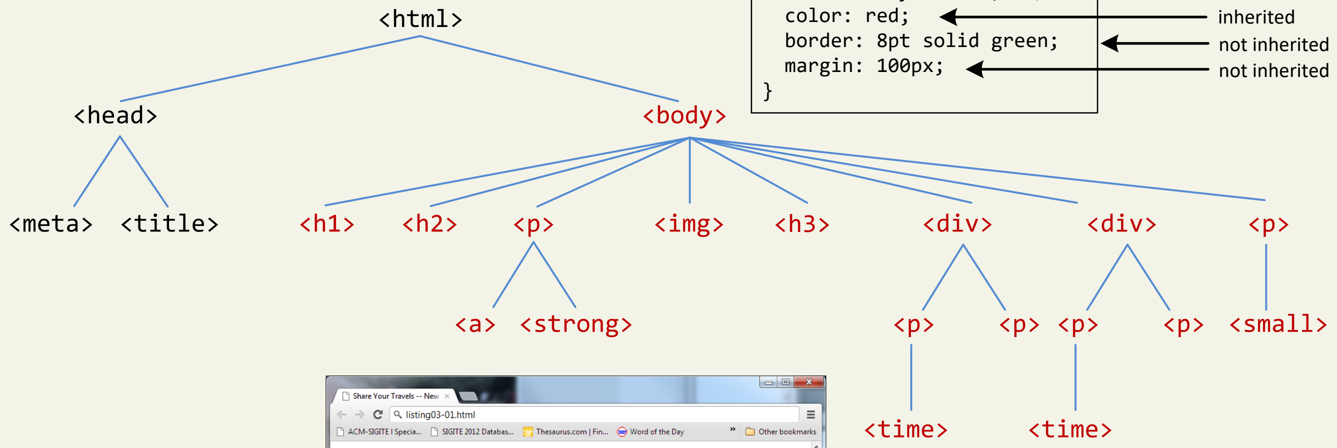
Many (but not all) CSS properties affect not only themselves but their descendants as well.

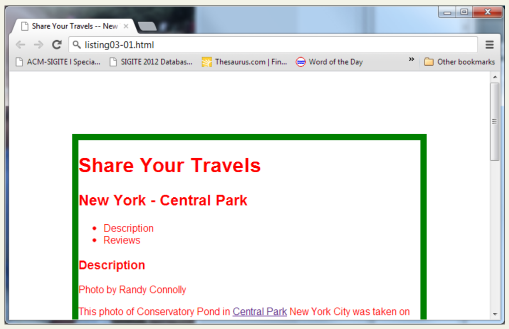
Font, color, list, and text properties are inheritable.

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

Inheritance

```
body {
  font-family: Arial;        ← inherited
  color: red;                ← inherited
  border: 8pt solid green;   ← not inherited
  margin: 100px;             ← not inherited
}
```





Share Your Travels

New York - Central Park

- Description
- Reviews

Description

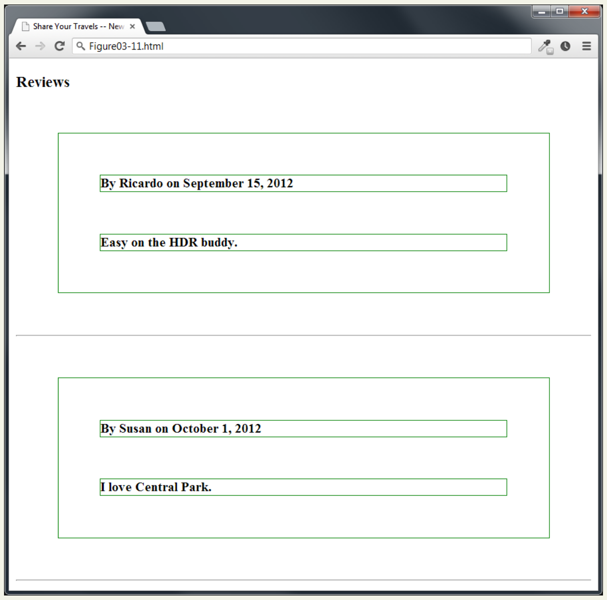
Photo by Randy Connolly

This photo of Conservatory Pond in Central Park New York City was taken on

Inheritance

How to force inheritance

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



Reviews

By Ricardo on September 15, 2012

Easy on the HDR buddy.

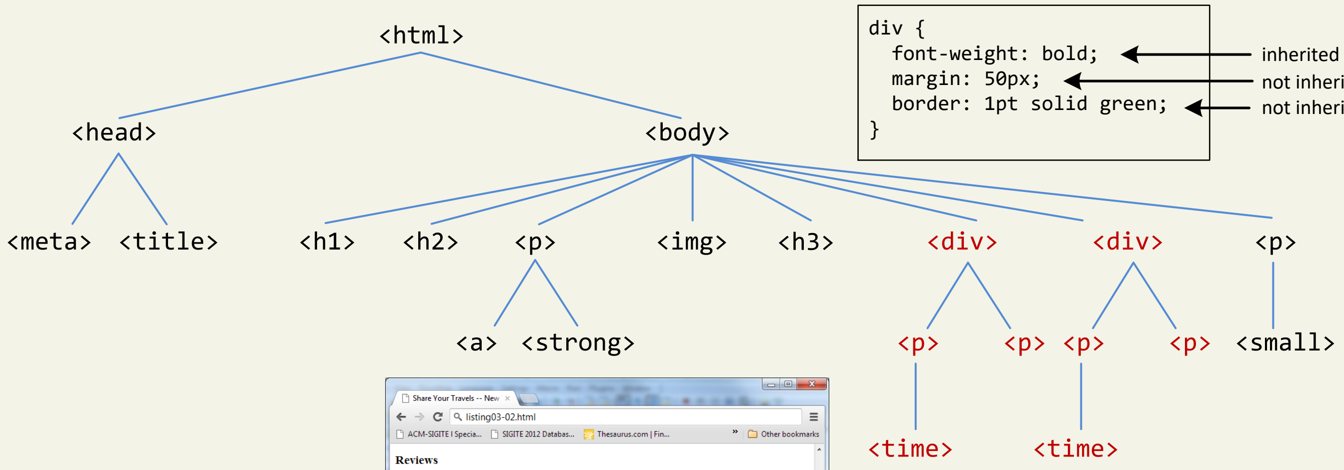
By Susan on October 1, 2012

I love Central Park.

```
div {
  font-weight: bold;
  margin: 50px;
  border: 1pt solid green;
}
p {
  border: inherit;
  margin: inherit;
}
```

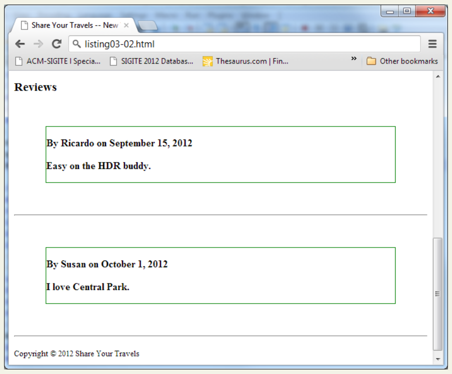
```
<h3>Reviews</h3>
<div>
   <p>By Ricardo on <time>September 15, 2012</time></p>
   <p>Easy on the HDR buddy.</p>
</div>
<hr/>

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

Inheritance



```
div {
  font-weight: bold;
  margin: 50px;
  border: 1pt solid green;
}
```

inherited
not inherited
not inherited



Reviews

By Ricardo on September 15, 2012

Easy on the HDR buddy.

By Susan on October 1, 2012

I love Central Park.

Copyright © 2012 Share Your Travels

Specificity

Cascade Principle #2

Specificity is how the browser determines which style rule takes precedence when more than one style rule could be applied to the same element.

The more *specific* the selector, the more it takes precedence (i.e., overrides the previous definition).

Specificity

How it works

The way that specificity works in the browser is that the browser assigns a weight to each style rule.

When several rules apply, the one with the greatest weight takes precedence.

Specificity

These color and font-weight properties are inheritable and thus potentially applicable to all the child elements contained within the body.

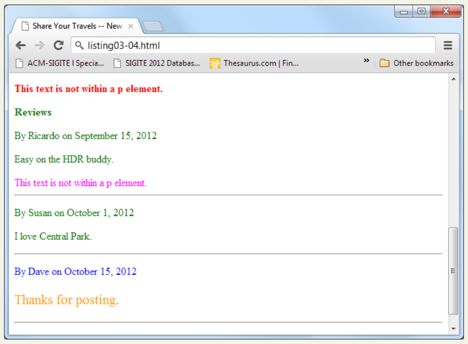
However, because the <div> and <p> elements also have the same properties set, they *override* the value defined for the <body> element because their selectors (div and p) are more specific.

Class selectors are more specific than element selectors, and thus take precedence and override element selectors.

Id selectors are more specific than class selectors, and thus take precedence and override class selectors.

```
body {
  font-weight: bold;
  color: red;
}

div {
  font-weight: normal;
  color: magenta;
}

p {
  color: green;
}

.last {
  color: blue;
}

#verylast {
  color: orange;
  font-size: 16pt;
}
```

```
This text is not within a p element.
<p>Reviews</p>
<div>
   <p>By Ricardo on <time>September 15, 2012</time
   <p>Easy on the HDR buddy.</p>
   This text is not within a p element.
</div>
<hr/>

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

<div>
   <p class="last">By Dave on <time>October 15, 20
   <p class="last" id="verylast">Thanks for postin
</div>
<hr/>
```



This text is not within a p element.

Reviews

By Ricardo on September 15, 2012

Easy on the HDR buddy.

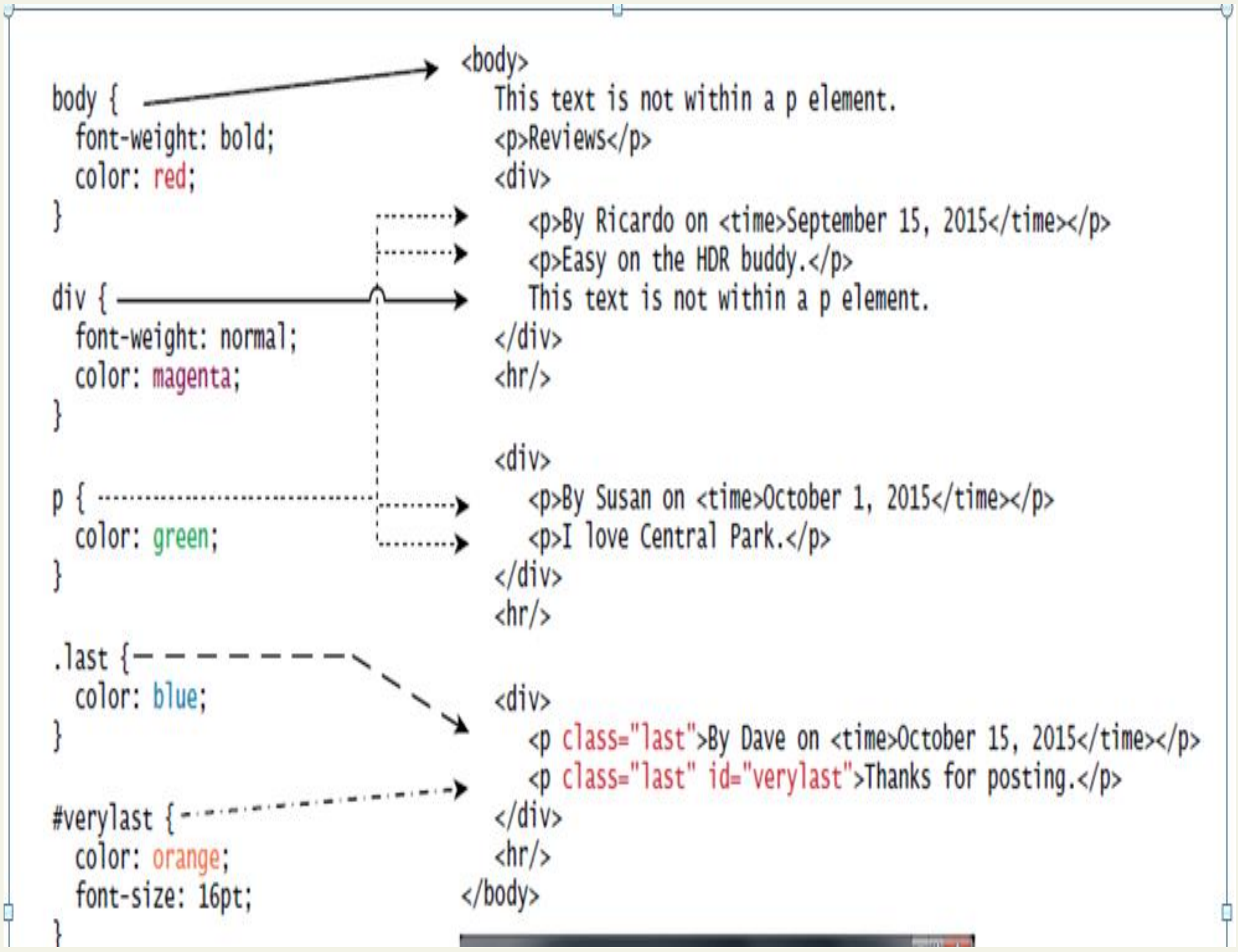
This text is not within a p element.

By Susan on October 1, 2012

I love Central Park.

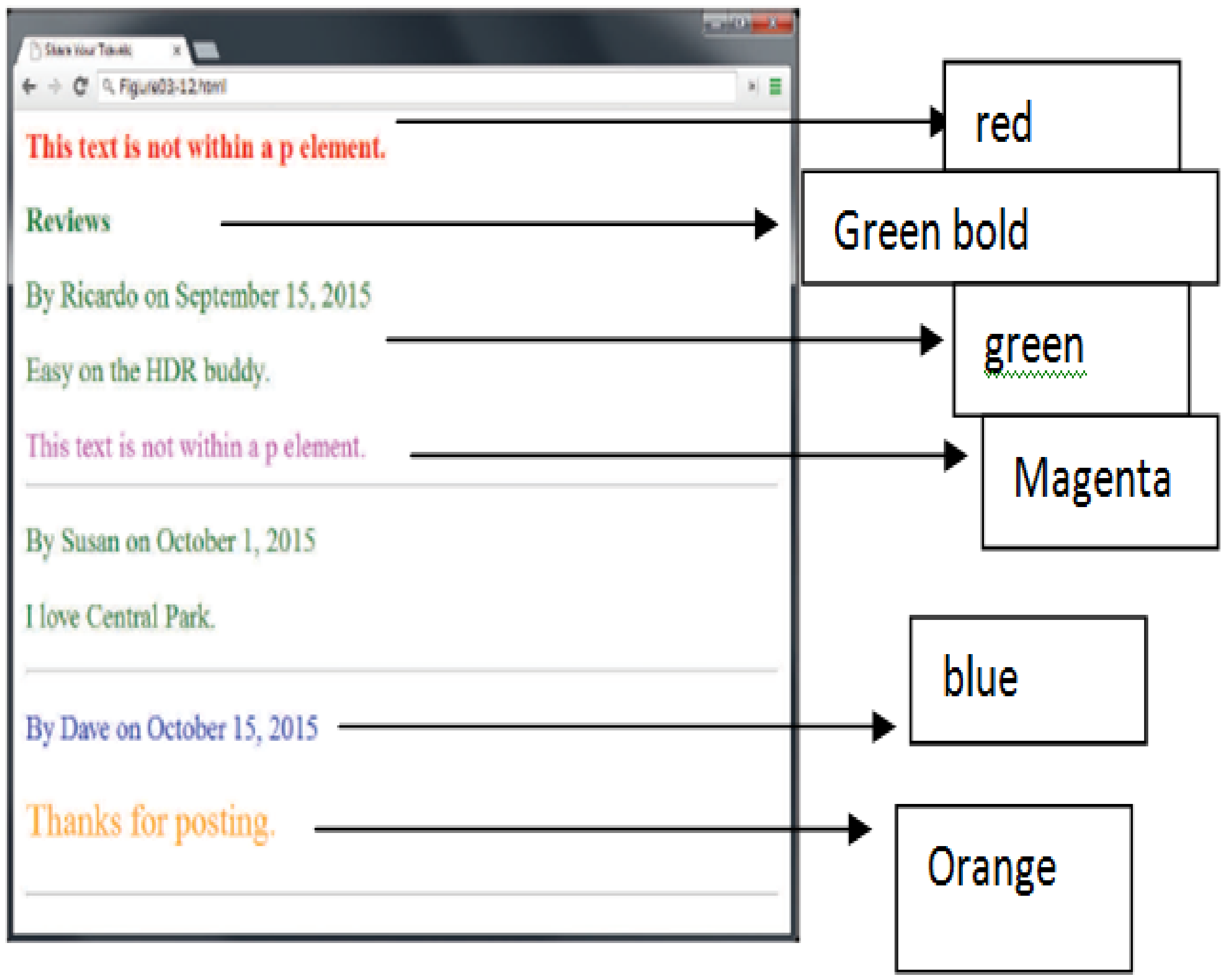
By Dave on October 15, 2012

Thanks for posting.



```
body {
  font-weight: bold;
  color: red;
}

div {
  font-weight: normal;
  color: magenta;
}

p {
  color: green;
}

.last {
  color: blue;
}

#verylast {
  color: orange;
  font-size: 16pt;
}
```

```
<body>
  This text is not within a p element.
  <p>Reviews</p>
  <div>
    <p>By Ricardo on <time>September 15, 2015</time></p>
    <p>Easy on the HDR buddy.</p>
    This text is not within a p element.
  </div>
  <hr/>

  <div>
    <p>By Susan on <time>October 1, 2015</time></p>
    <p>I love Central Park.</p>
  </div>
  <hr/>

  <div>
    <p class="last">By Dave on <time>October 15, 2015</time></p>
    <p class="last" id="verylast">Thanks for posting.</p>
  </div>
  <hr/>
</body>
```



This text is not within a p element.

Reviews

By Ricardo on September 15, 2015

Easy on the HDR buddy.

This text is not within a p element.

By Susan on October 1, 2015

I love Central Park.

By Dave on October 15, 2015

Thanks for posting.

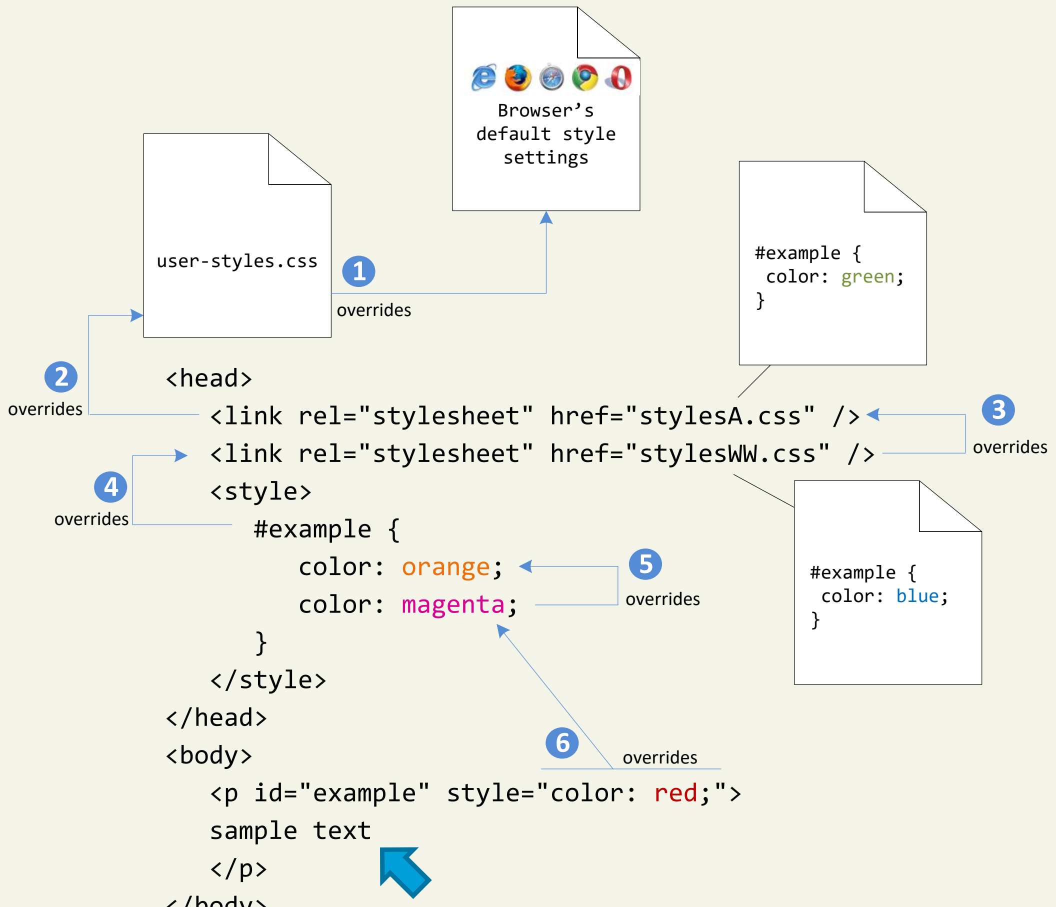
Location

Cascade Principle #3

When inheritance and specificity cannot determine style precedence, the principle of **location** will be used.

The principle of location is that when rules have the same specificity, then the latest are given more weight.

Location



```
#example {
 color: green;
}
```

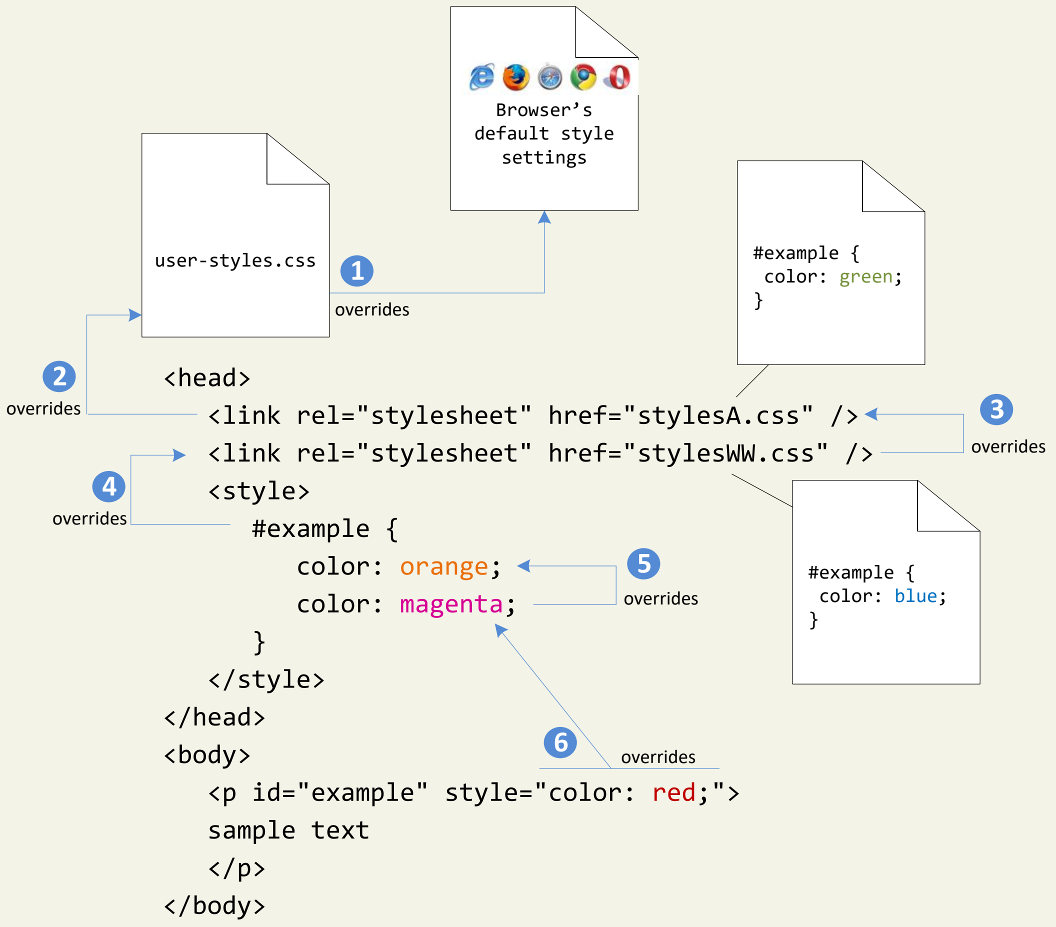
```
#example {
 color: blue;
}
```

```
<head>
   <link rel="stylesheet" href="stylesA.css" />
   <link rel="stylesheet" href="stylesWW.css" />
   <style>
      #example {
         color: orange;
         color: magenta;
      }
   </style>
</head>
<body>
   <p id="example" style="color: red;">
   sample text
   </p>
</body>
```

Can you guess what will be the color of the sample text ?

Location

What color would the sample text be if there wasn't an inline style definition?



```
#example {
 color: green;
}
```

```
#example {
 color: blue;
}
```

```
<head>
   <link rel="stylesheet" href="stylesA.css" />
   <link rel="stylesheet" href="stylesWW.css" />
   <style>
      #example {
         color: orange;
         color: magenta;
      }
   </style>
</head>
<body>
   <p id="example" style="color: red;">
   sample text
   </p>
</body>
```

Section 6 of 7

THE BOX MODEL

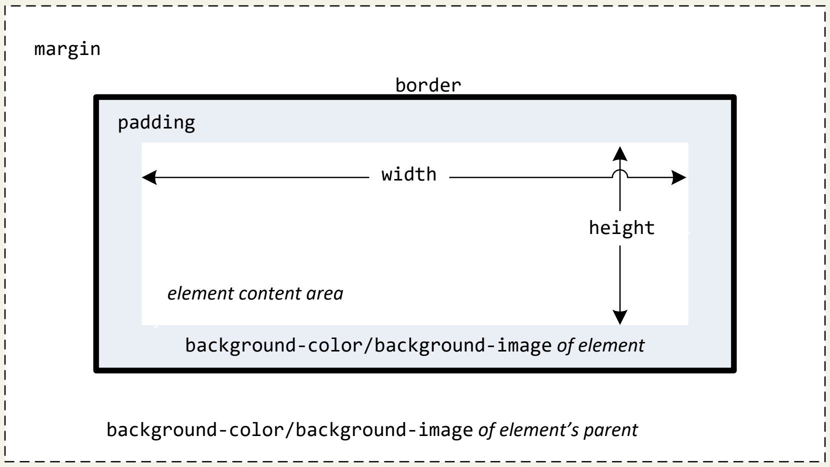
The Box Model

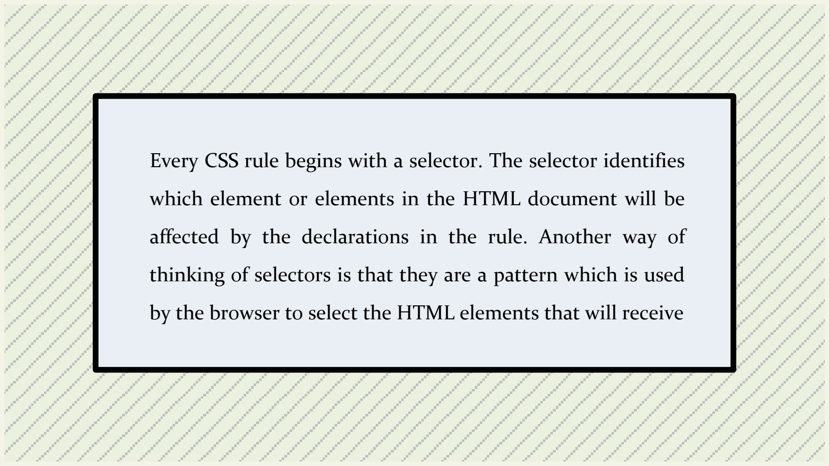
Time to think inside the box

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

It is absolutely essential that you familiarize yourself with the terminology and relationship of the CSS properties within the element box.

The Box Model


margin
border
padding
width
height
element content area
background-color/background-image *of element*
background-color/background-image *of element's parent*


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

Box Model Property #1

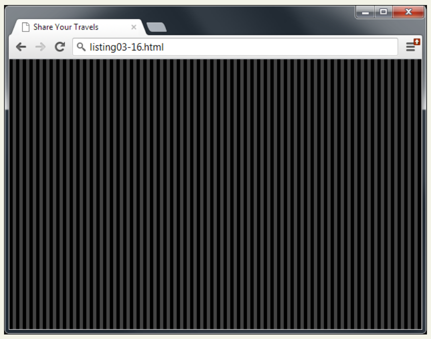
The background color or image of an element fills an element out to its border (if it has one that is).

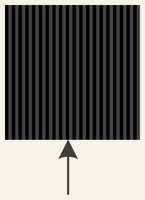
In web design, it has become extremely common too use CSS to display purely presentational images rather than using the <img> element.

Background Properties

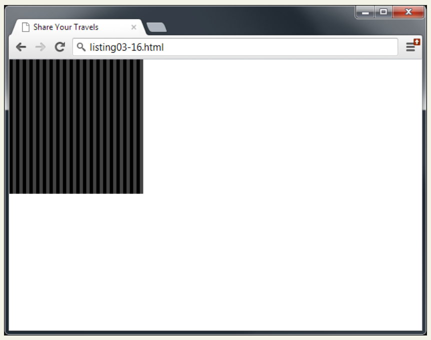
Property	Description
background	A combined short-hand property that allows you to set the background values in one property. While you can omit properties with the short-hand, do remember that any omitted properties will be set to their default value.
background-attachment	Specifies whether the background image scrolls with the document (default) or remains fixed. Possible values are: fixed, scroll.
background-color	Sets the background color of the element.
background-image	Specifies the background image (which is generally a jpeg, gif, or png file) for the element. Note that the URL is relative to the CSS file and not the HTML. CSS3 introduced the ability to specify multiple background images.
background-position	Specifies where on the element the background image will be placed. Some possible values include: bottom, center, left, and right. You can also supply a pixel or percentage numeric position value as well. When supplying a numeric value, you must supply a horizontal/vertical pair; this value indicates its distance from the top left corner of the element.
background-repeat	Determines whether the background image will be repeated. This is a common technique for creating a tiled background (it is in fact the default behavior). Possible values are: repeat, repeat-x, repeat-y, and no-repeat.
background-size	New to CSS3, this property lets you modify the size of the background image.

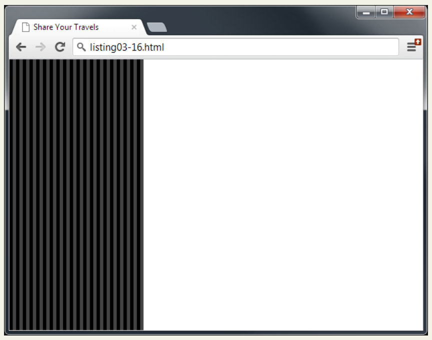
Background Repeat

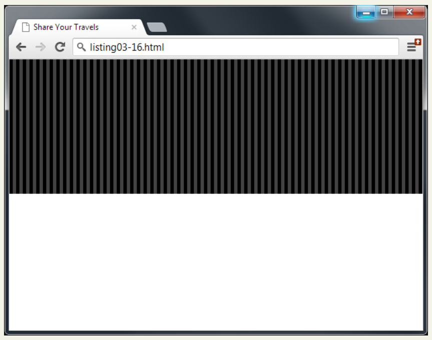




```
background-image: url(../images/backgrounds/body-background-tile.gif);
background-repeat: repeat;
```





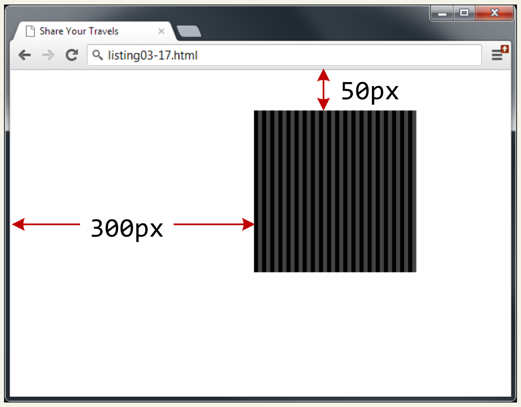


```
background-repeat: no-repeat;
```

```
background-repeat: repeat-y;
```

```
background-repeat: repeat-x;
```

Background Position



```
body {
        background: white url(../images/backgrounds/body-background-tile.gif) no-repeat;
        background-position: 300px 50px;
}
```

Borders

Box Model Property #2

Borders provide a way to visually separate elements.

You can put borders around all four sides of an element, or just one, two, or three of the sides.

Borders

Property	Description
border	A combined short-hand property that allows you to set the style, width, and color of a border in one property. The order is important and must be: **border-style border-width border-color**
border-style	Specifies the line type of the border. Possible values are: solid, dotted, dashed, double, groove, ridge, inset, and outset.
border-width	The width of the border in a unit (but not percents). A variety of keywords (thin, medium, etc) are also supported.
border-color	The color of the border in a color unit.
border-radius	The radius of a rounded corner.
border-image	The URL of an image to use as a border.

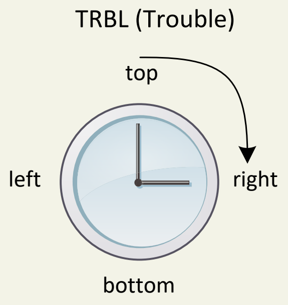
Shortcut notation

TRBL

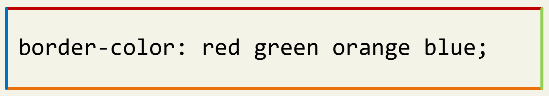
With border, margin, and padding properties, there are long-form and shortcut methods to set the 4 sides

```
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 */

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

border-color: red green orange blue;     /* sets all four sides differently */
```

When using this multiple values shortcut, they are applied in clockwise order starting at the top.
Thus the order is: **top right bottom left.**



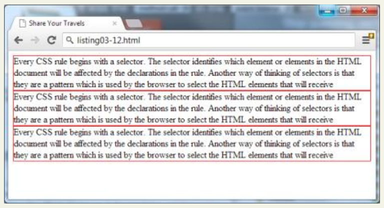
```
border-color: top right bottom left;
```



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

Margins and Padding

Box Model Properties #3 and #4

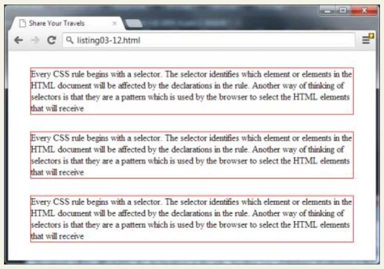


Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML elements that will receive

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

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

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

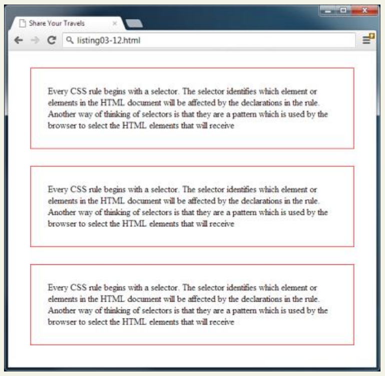


Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML elements that will receive

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

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

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



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

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

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

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

Width and Height

Box Model Properties #5 and #6

The width and height properties specify the size of the element's content area.

Perhaps the only rival for collapsing margins in troubling our students, box dimensions have a number of potential issues.

Width and Height

Potential Problem #1

Only block-level elements and non-text inline elements such as images have a **width** and **height** that you can specify.

By default the width of and height of elements is the actual size of the content.

For text,

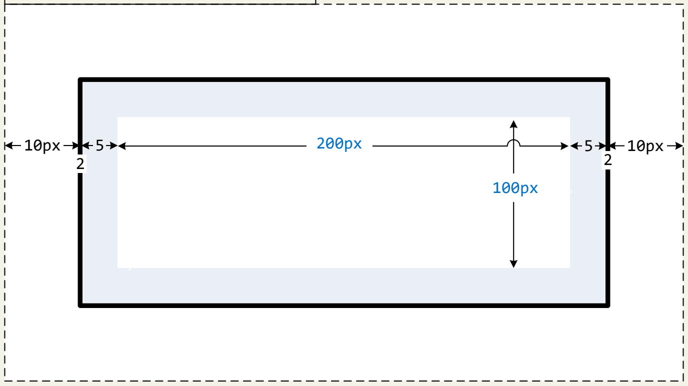
- this is determined by the font size and font face;

For images,

- the width and height of the actual image in pixels determines the element box's dimensions.

```
div {
  box-sizing: content-box;
  width: 200px;
  height: 100px;
  padding: 5px;
  margin: 10px;
  border: solid 2pt black;
}
```

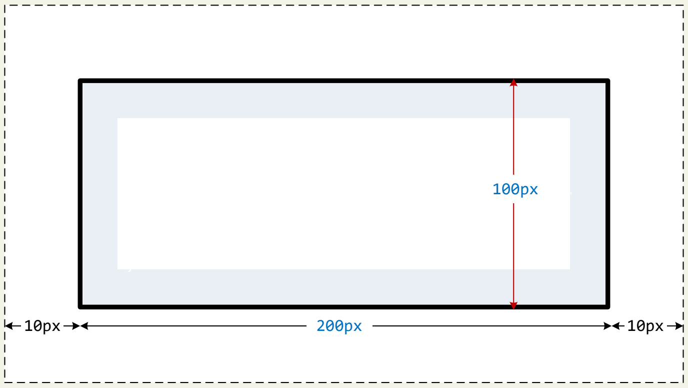
True element width = 10 + 2 + 5 + 200 + 5 + 2 + 10 = 234 px
True element height = 10 + 2 + 5 + 100 + 5 + 2 + 10 = 134 px



Default

```
div {
  ...
  box-sizing: border-box;
}
```

True element width = 10 + 200 + 10 = 220 px
True element height = 10 + 100 + 10 = 120 px



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 text in a similar way no matter which font is being used.

Font-Family

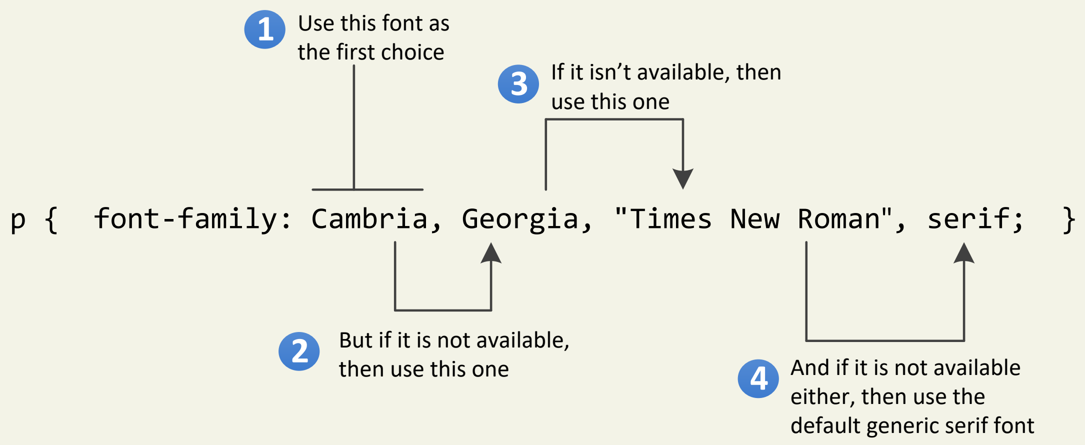
A few issues here

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 in not on the user's computer.

Specifying the Font-Family



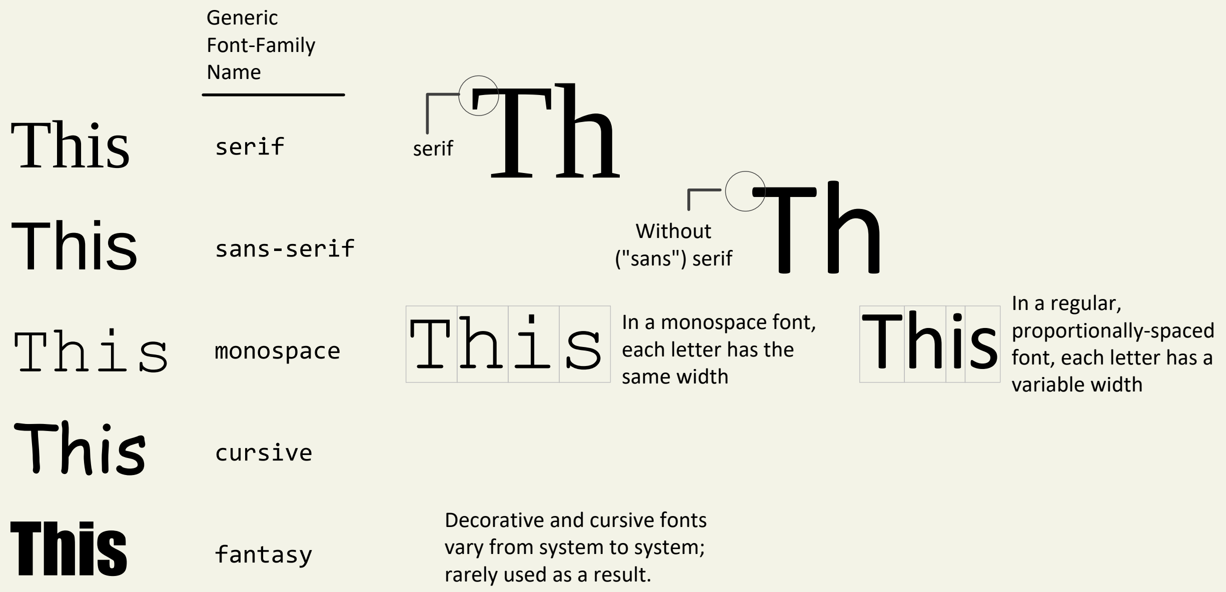
```
p {  font-family: Cambria, Georgia, "Times New Roman", serif;  }
```

1. Use this font as the first choice
2. But if it is not available, then use this one
3. If it isn't available, then use this one
4. And if it is not available either, then use the default generic serif font

Generic Font-Family

The font-family property supports five different generic families.

The browser supports a typeface from each family.



Generic Font-Family Name
This
serif
This
sans-serif
This
monospace
This
cursive
This
fantasy
serif
Th
Without ("sans") serif
Th
This
In a monospace font, each letter has the same width
This
In a regular, proportionally-spaced font, each letter has a variable width
Decorative and cursive fonts vary from system to system; rarely used as a result.

Font Sizes

Welcome ems and percents again

If we wish to create web layouts that work well on different devices, we should learn to use relative units such as **em** units or **percentages** for our font sizes (and indeed for other sizes in CSS as well).

One of the principles of the web is that the user should be able to change the size of the text if he or she so wishes to do so.

Using percentages or em units ensures that this user action will work.

How to use ems and percents

When used to specify a font size, both em units and percentages are relative to the parent's font size.

How to use ems and percents

`<body>`	Browser's default text size is usually 16 pixels
`<p>`	100% or 1em is 16 pixels
`<h3>`	125% or 1.125em is 18 pixels
`<h2>`	150% or 1.5em is 24 pixels
`<h1>`	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>
```

How to use ems and percents

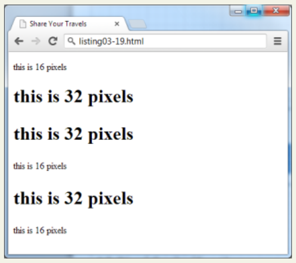
It might seem easy … but it's not …

While this looks pretty easy to master, things unfortunately can quickly become quite complicated.

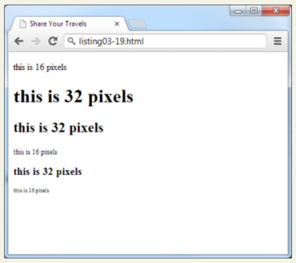
Remember that percents and em units are relative to their parents, so if the parent font size changes, this affects all of its contents.

ems and percents

```
<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>
```



```
/* using 16px scale */

body { font-size: 100%; }
p    { font-size: 1em; }      /* 1 x 16 = 16px */
h1   { font-size: 2em; }      /* 2 x 16 = 32px */
```



```
/* using 16px scale */

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

article { font-size: 75% }   /* h1 = 2 * 16 * 0.75 = 24px
                                 p = 1 * 16 * 0.75 = 12px   */

div  { font-size: 75% }      /* h1 = 2 * 16 * 0.75 * 0.75 = 18px
                                 p = 1 * 16 * 0.75 * 0.75 = 9px    */
```

What you’ve learned

1 What is CSS?

2 CSS Syntax

3 Location of Styles

4 Selectors

5 The Cascade: How Styles Interact

6 The Box Model

7 CSS Text Styling