CS4032 Web Programming

Cascading Stylesheets (CSS)

Contents

zeshan.khan@nu.edu.pk

CSS

- 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
- Most commonly, in a separate text file that contains only CSS.
- Can also be in-line (Not recommended)

Advantages

- Let HTML focus on the layout only
- The code is more easily maintainable with CSS taking care of presentation
- Sites with centralized CSS files are quicker to load they support multiple HTML pages
- CSS can be used to adopt a page for different output mediums.

Version

- W3C published the CSS Level 1 Recommendation in 1996
- CSS Level 2 was published in 1997
- CSS2.1 was published officially in 2011
- While CSS2.1 was developing CSS3 was popularized by another group at W3C

Browsers

- Not all browsers support all CSS features (Check on multiple browsers)
- Not all versions of a browser support the same CSS features (Verify with multiple versions of a browser)
- Can validate CSS through the W3C validator as well.

Syntax

- A CSS document consists of one or more style rules.
- A rule has a selector (to be effected), a property (to modify), and value (to modify with)
- Rule1:
  - Selector {
    - Property: value;
    - Property2: value2
  - }

- p{
  - color: #000;
- }
- h1{
  - margin : 0px;
  - color : red;
- }

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

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

Property values

- Each CSS declaration also contains a **value** for a property.
- Possible value may be chosen from a predefined list
- Numeric values are supported with Units
  - Absolute Units (pt, px, in)
  - Relative Units (%, em)

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

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

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;

Comments in CSS

- It is often helpful to add comments to your style sheets. Comments take the form:
- ***/* comment goes here */***

zeshan.khan@nu.edu.pk

CSS Style Locations

- The browser style sheet defines the default styles the browser uses for each HTML element.
- 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.
- Author-created style sheets (what we are learning in this presentation).

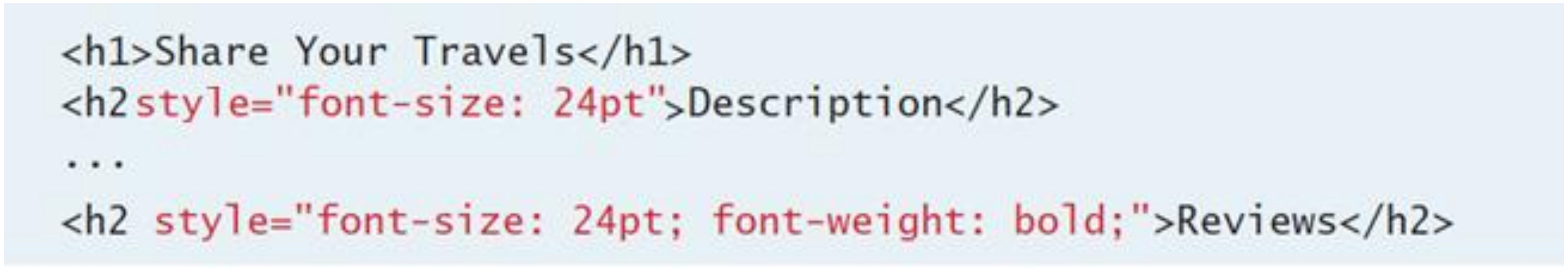
Style Locations

CSS style rules can be located in three different locations.

- Inline
- Embedded
- External

You can combine all 3

Inline Styles



```
<h1>Share Your Travels</h1>
<h2style="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

Its use is discouraged

Has highest priority

Embedded Style Sheet

```
<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

Better than inline but discourage by large

Has second highest priority

External Style Sheet

```
<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

Commonly used and highly recommended approach

Has lowest priority after inline and embedded

- Provides a single point to update the style of multiple HTML documents (Generalized solution)

- The browser is able to cache the external style sheet which can improve the performance of the site

Example Here

Selectors

- 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.

Element Selectors

Uses the HTML element name.

```
p{

    color: #000;

}

h1{

    margin : 0px;

    color : red;

}
```

Grouped Selectors

- 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.

- P, span, h1 {
  - color: #000;
- }
- h1{
  - margin : 0px;
- }
- Div, aside {
  - Margin:0;
  - Padding:0;
- }

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

- A class selector allows you to simultaneously target different HTML elements regardless of their position in the document tree.
- If a series of HTML element 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 Selector Example

HTML

- <header class=“header-data”>
    - <h1>Header of the Page</h1>
    - <nav><span>Navigation</span></nav>
- </header>
- <section class=“page-data”>
    - <h2>First Level H2</h2>
    - <div>
        - <h2>Second level H2</h2>
    - </div>
- </section>
- <section class=“page-data”>
    - <p>Some text here.<p>
- </section>

CSS

- .page-data{
    - color:#000;
- }
- .page-data{
    - Font-style:italic;
- }
- .page-data{
    - Font-style:bold;
- }

Id Selectors

- An id selector 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, which takes the form: pound/hash (#) followed by the id name.
- Note: You should only be using an id once per page

zeshan.khan@nu.edu.pk

Class Selector Example

HTML

- <header id=“header-data”>
  - <h1>Header of the Page</h1>
  - <nav><span>Navigation</span></nav
    >
- </header>
- <section id=“page-data”>
  - <h2>First Level H2</h2>
  - <div>
    - <h2>Second level H2</h2>
  - </div>
- </section>
- <section id=“page-data2”>
  - <p>Some text here.<p>
- </section>

CSS

- #page-data{
  - color:#000;
- }
- #page-data{
  - Font-style:italic;
- }
- #page-data2 {
  - Font-style:bold;
- }

Id versus Class Selectors

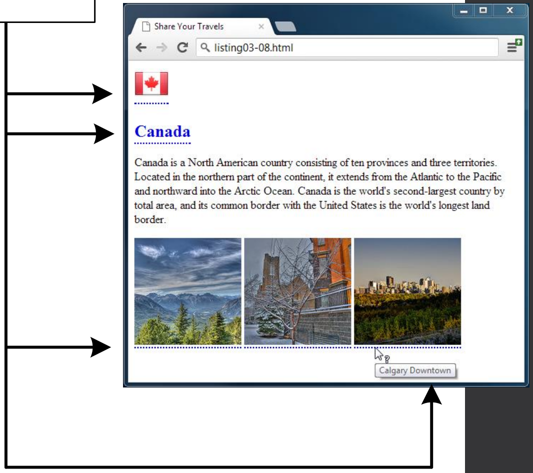
- Id selectors should only be used when referencing a single HTML element since an id attribute can only be assigned to a single HTML element.
- Class selectors should be used when (potentially) referencing several related elements.

Attribute Selectors

- 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.

```
<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>
```

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



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.

Pseudo Selectors

- A pseudo-element selector is a way to select something that does not exist explicitly as an element in the HTML document tree but which is still a recognizable selectable object.
- A 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

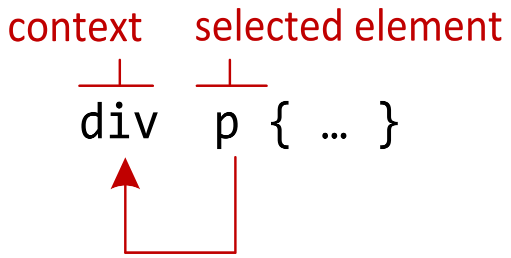
Contextual Selectors

- 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.

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>.

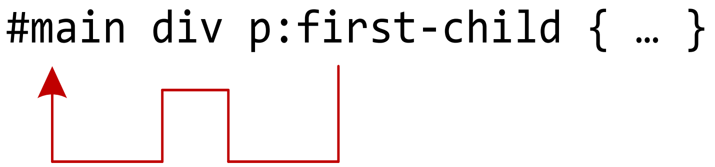
Descendant Selector

- A descendant selector matches all elements that are contained within another element. The character used to indicate descendant selection is the space character.



```
div  p { … }
```

Selects a `<p>` element somewhere
within a `<div>` element



```
#main div p:first-child { … }
```

Selects the first `<p>` element somewhere within a `<div>` element that is somewhere within an element with an `id="main"`

Example Here

Cascade: Conflict

- 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 Principles

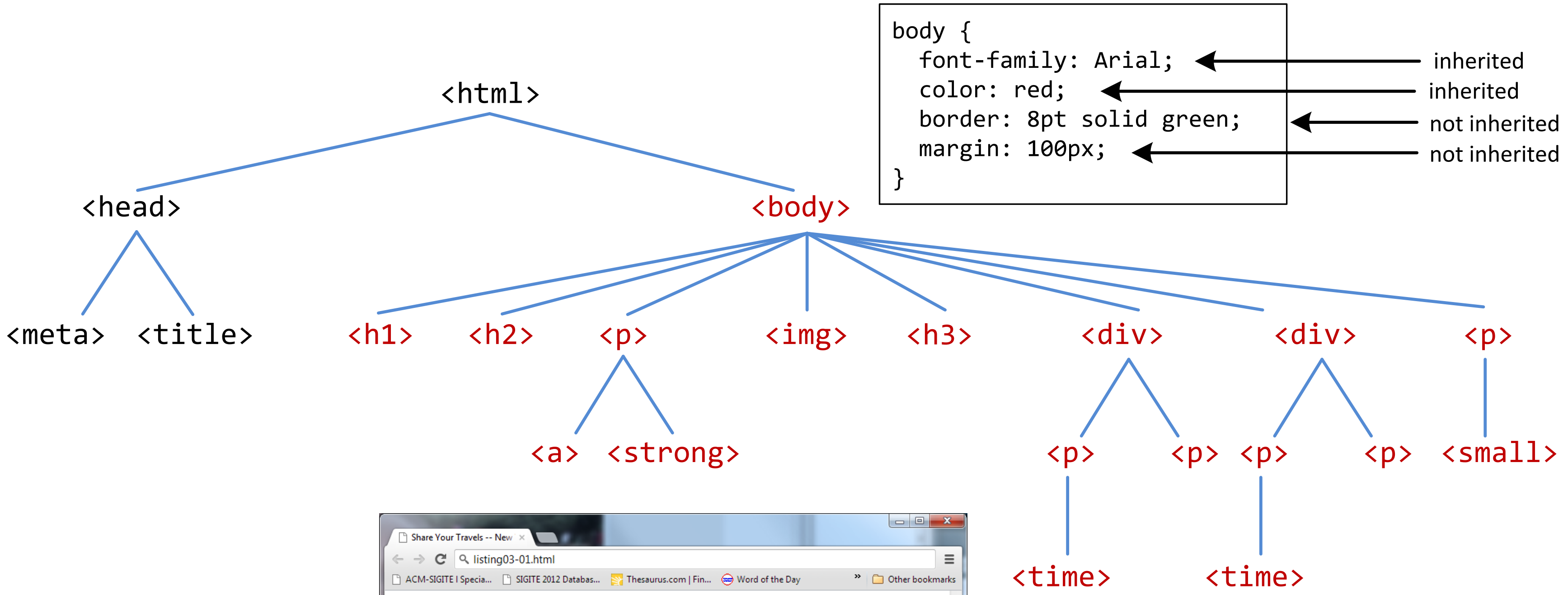
- CSS uses the following cascade principles to help it deal with conflicts:
- Inheritance
- Specificity (Precedence)
- Location (Where style is defined)

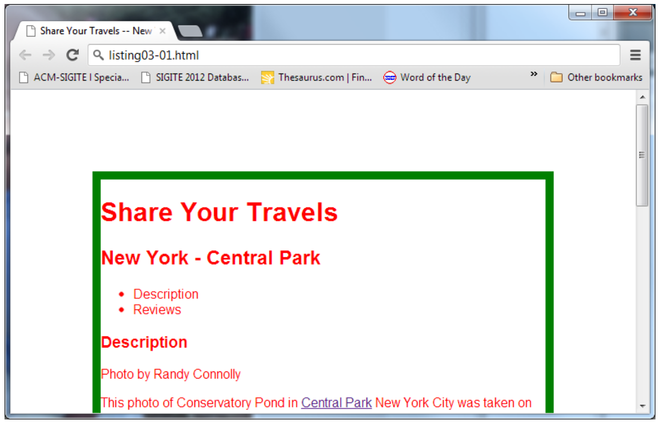
Inheritance

- 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;
  color: red;
  border: 8pt solid green;
  margin: 100px;
}
```





Share Your Travels

New York - Central Park

- Description
- Reviews

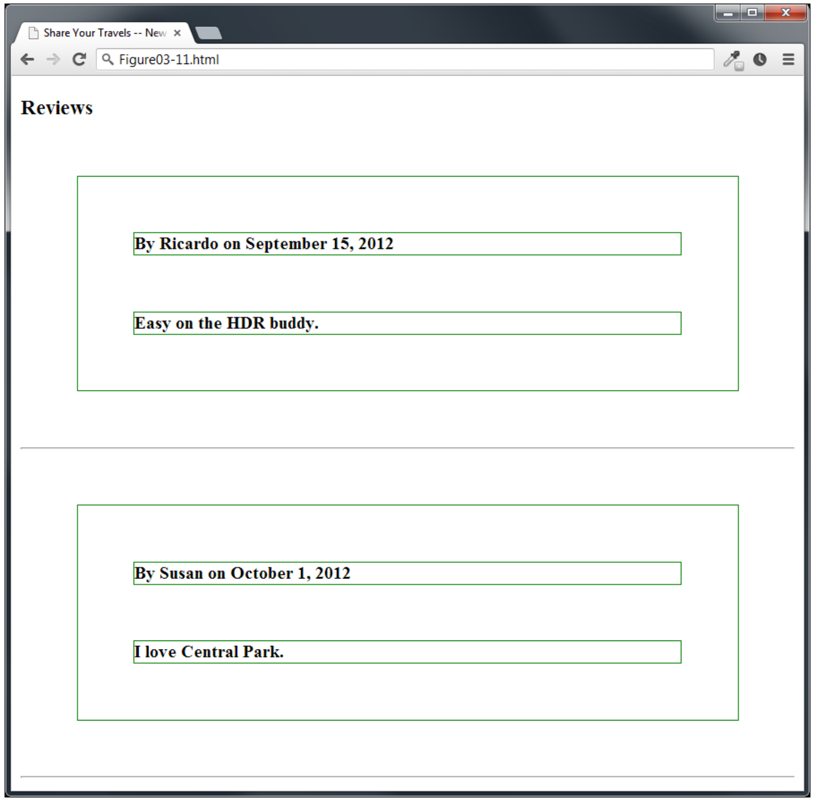
Description

Photo by Randy Connolly

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

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

Inheritance



Reviews

By Ricardo on September 15, 2012

Easy on the HDR buddy.

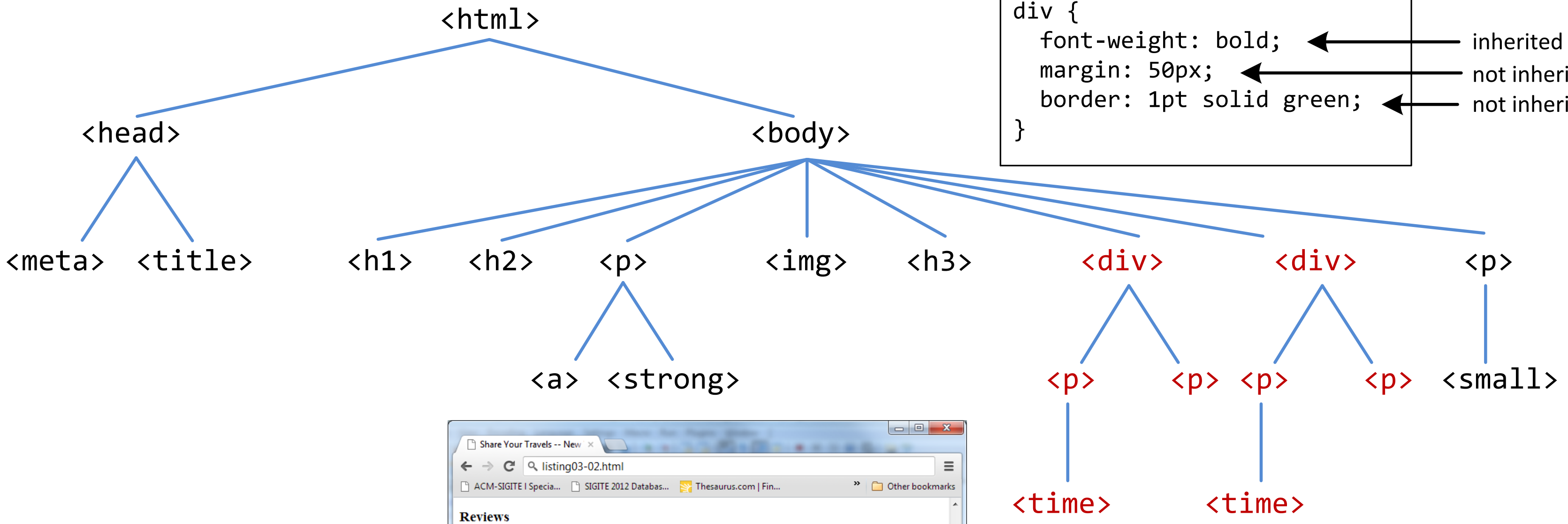
By Susan on October 1, 2012

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/>
```

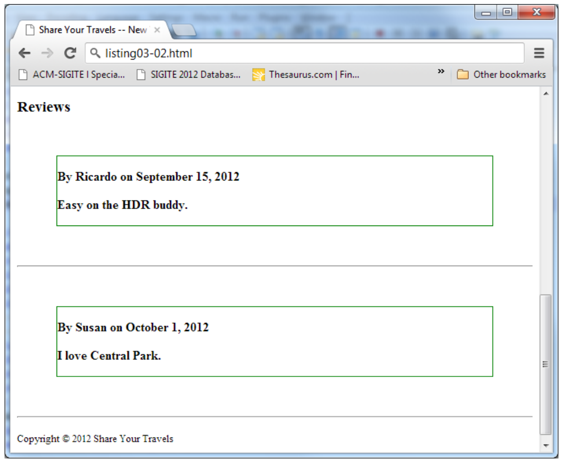
zeshan.khan@nu.edu.pk

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

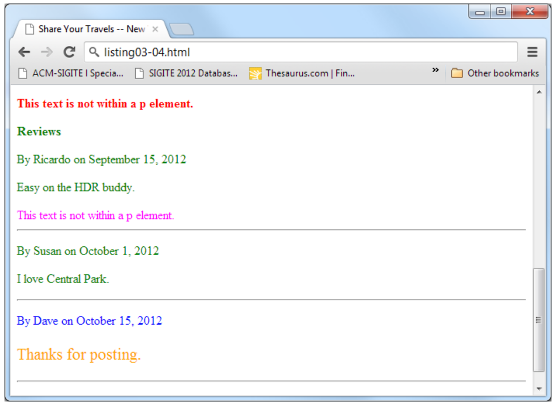
Specificity

- 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

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

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

p {
  color: green;
}

.last {
  color: blue;
}

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

```html
This text is not within a p element.
<p>Reviews</p>
<div>
   <p>By Ricardo on <time>September 15, 2012</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, 2012</time></p>
   <p>I love Central Park.</p>
</div>
<hr/>

<div>
   <p class="last">By Dave on <time>October 15, 2012</time></p>
   <p class="last" id="verylast">Thanks for posting.</p>
</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.

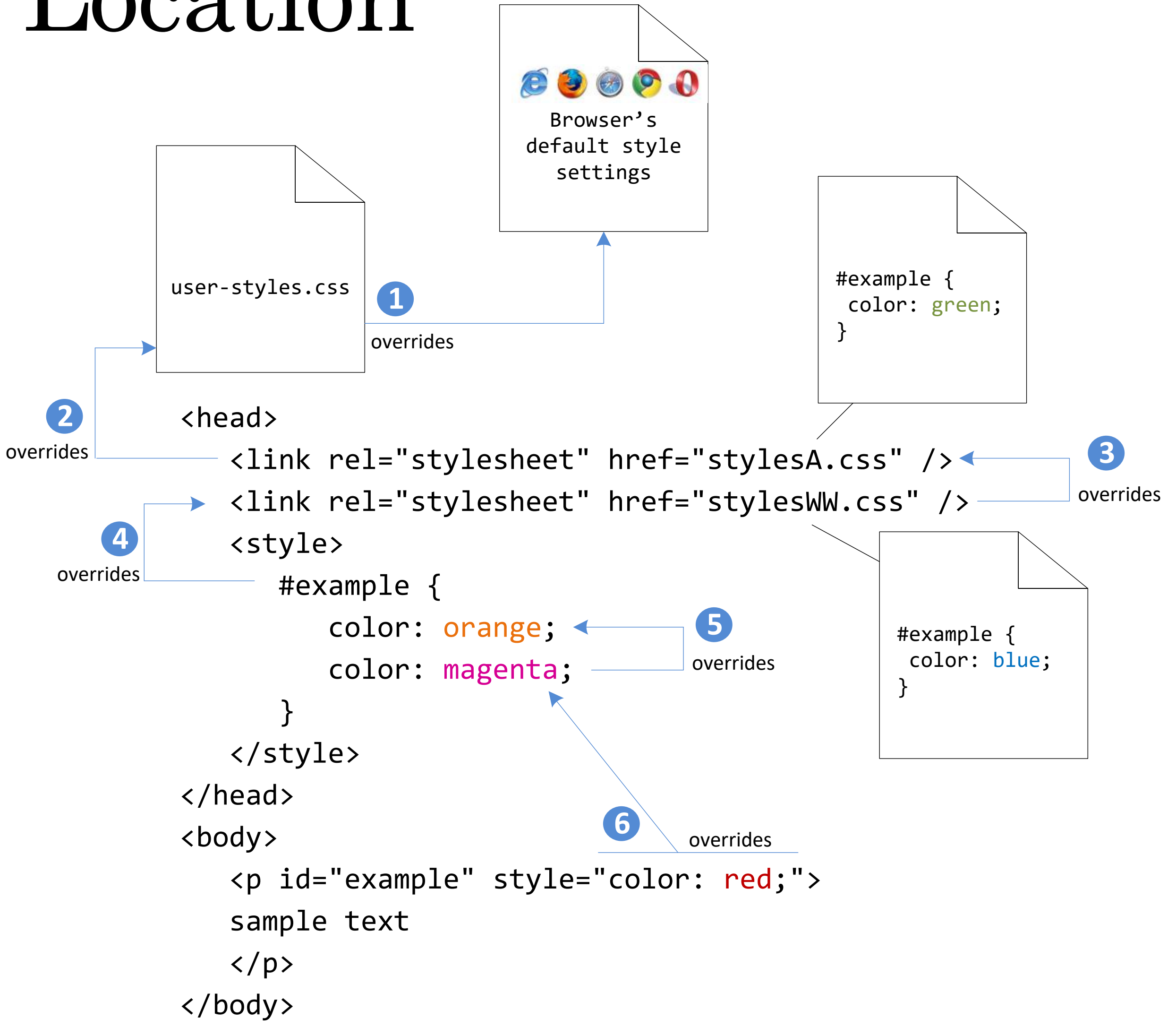
By Dave on October 15, 2012

Thanks for posting.

Location

- 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

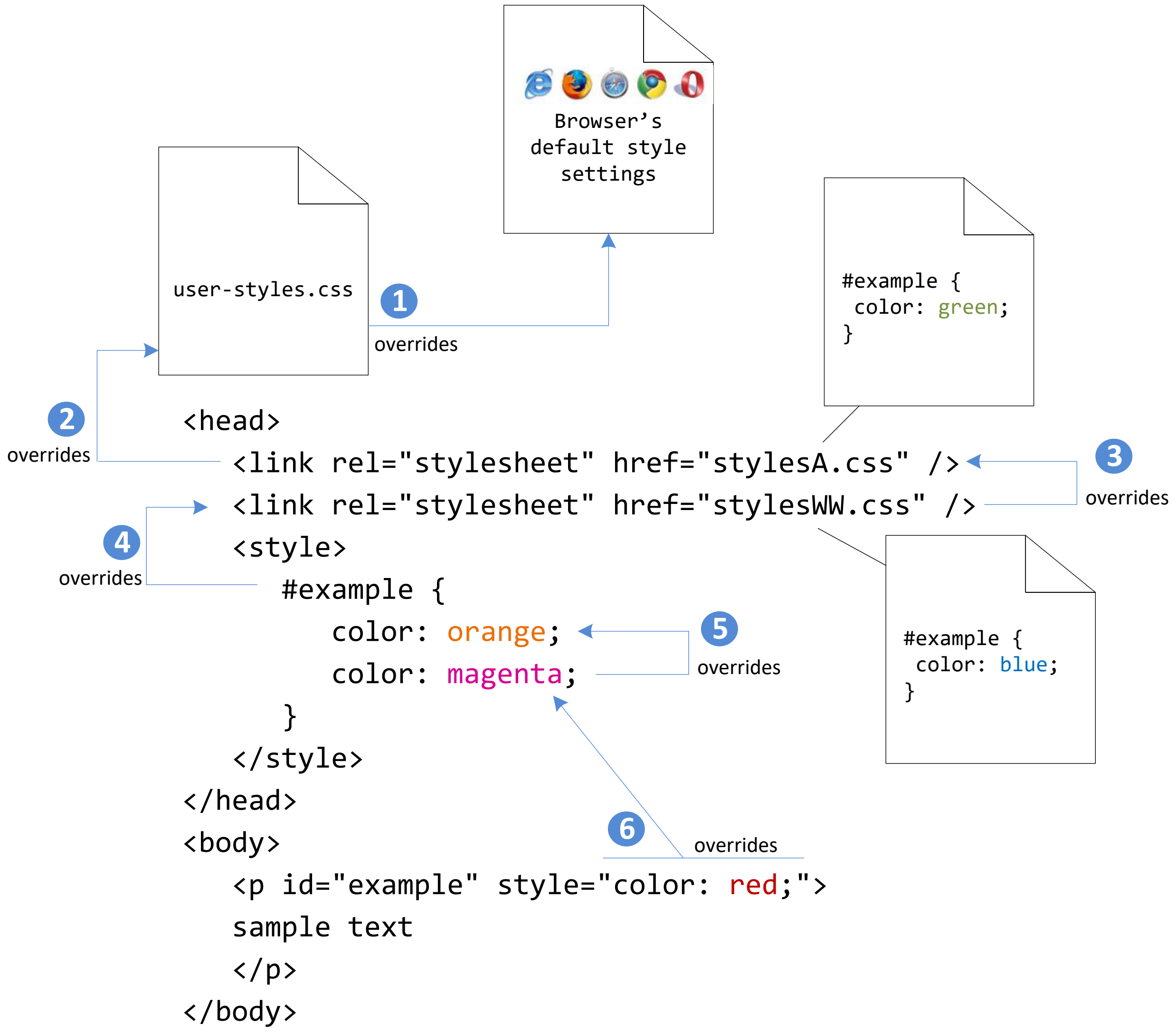


```
#example {
 color: green;
}
```

```
#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>
```

Location



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

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

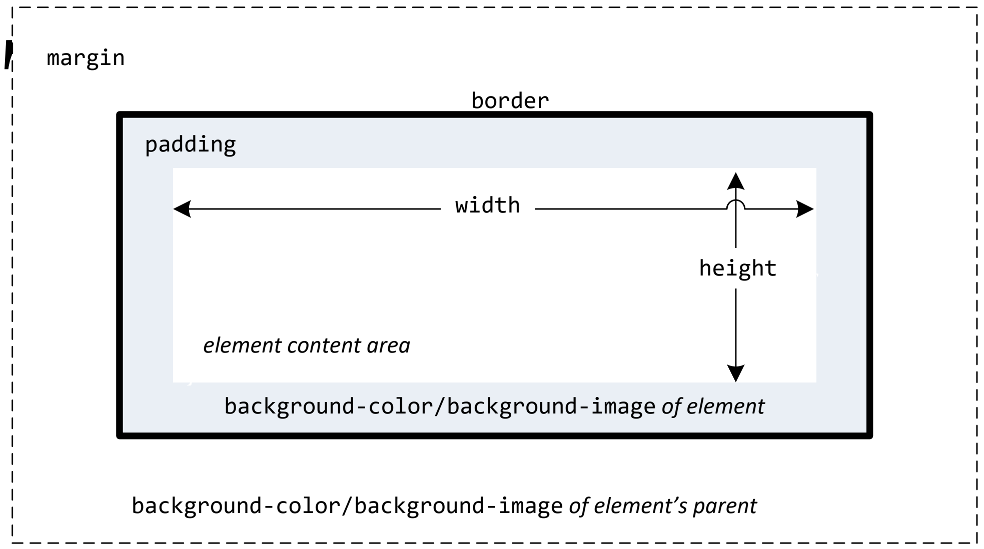
```
<head>
   <link rel="stylesheet" href="stylesA.css" />
   <link rel="stylesheet" href="stylesWW.css" />
   <style>
      #example {
         color: orange;
         color: magenta;
      }
   </style>
</head>
<body>
   <p id="example" style="color: red;">
   sample text
   </p>
</body>
```

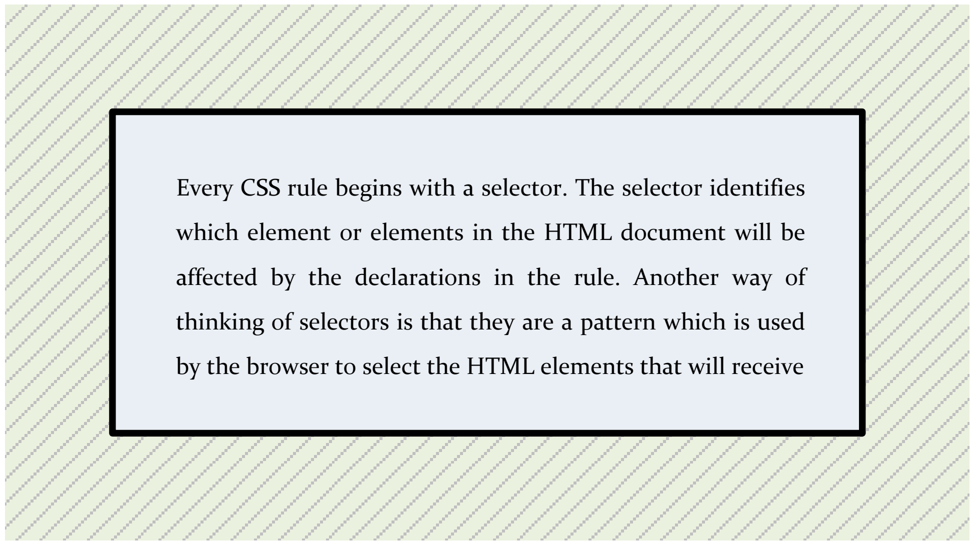
Location

- There is one exception to the principle of location.
- If a property is marked with !important in an author-created style rule, then it will override any other author-created style regardless of its location.
- The only exception is a style marked with !important in an user style sheet; such a rule will override all others.

The Box Model

- 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.


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

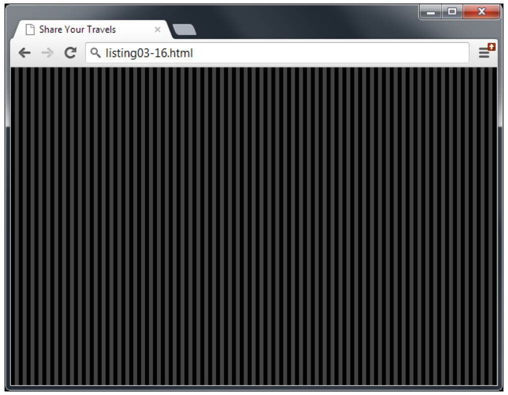
Example Here

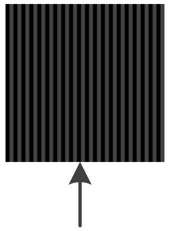
Background

- The background color or image of an element fills an element out to its border (if it has one that is).
- In contemporary web design, it has become extremely common too use CSS to display purely presentational images (such as background gradients and patterns, decorative images, etc) rather than using the <img> element.

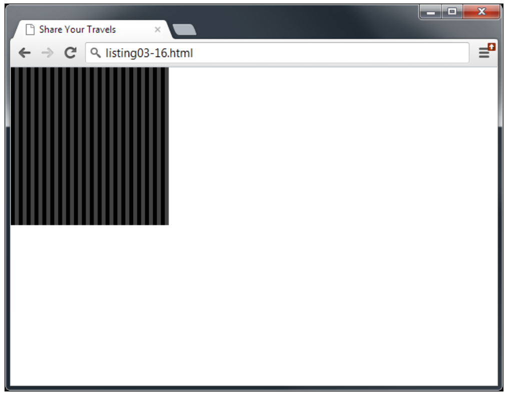
Property	Description
background	A combined short-hand property that allows you to set the background values in one property. While you can omit properties with the short-hand, do remember that any omitted properties will be set to their default value.
background-attachment	Specifies whether the background 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

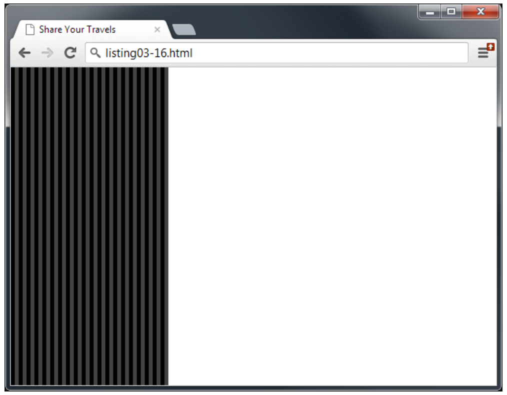




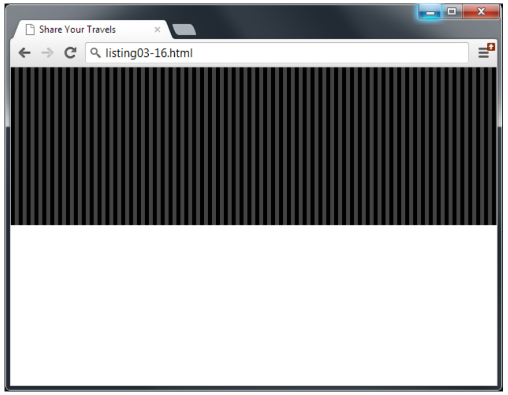
```
background-image: url(../images/backgrounds/body-background-tile.gif);
background-repeat: repeat;
```



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

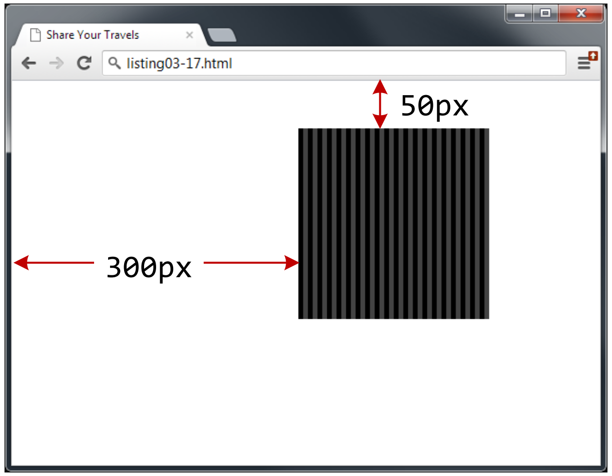


```
background-repeat: repeat-y;
```



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

Background Position



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

zeshan.khan@nu.edu.pk

Example Here

zeshan.khan@nu.edu.pk

Borders

- 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.

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.

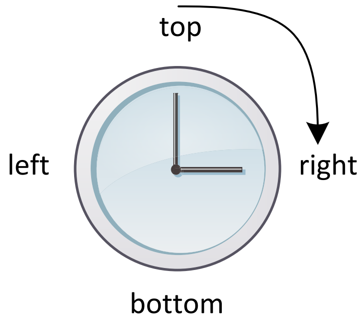
Shortcut notation

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.**

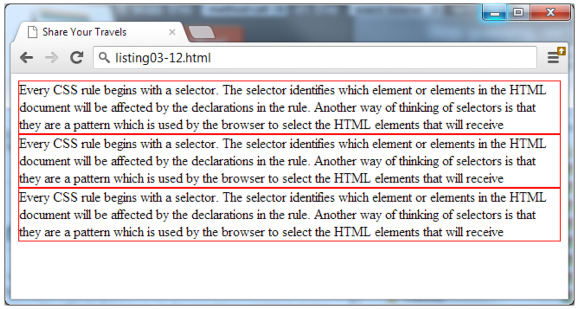
TRBL (Trouble)



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

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

Margins and Padding

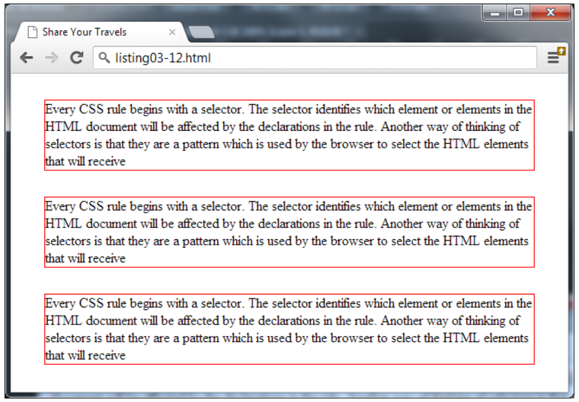

Share Your Travels
listing03-12.html

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;
}
```

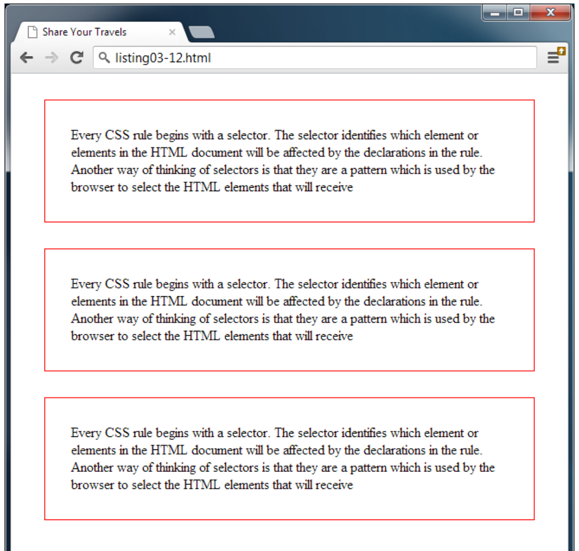

Share Your Travels
listing03-12.html

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;
}
```


Share Your Travels
listing03-12.html

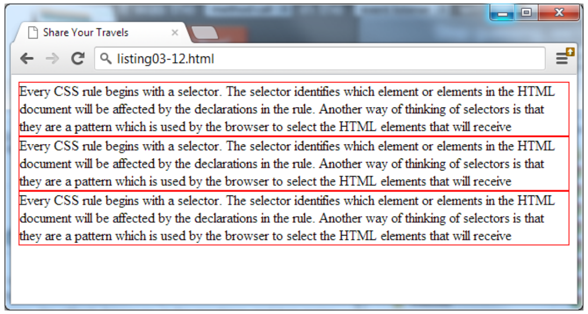
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML elements that will receive

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;
}
```

Margins

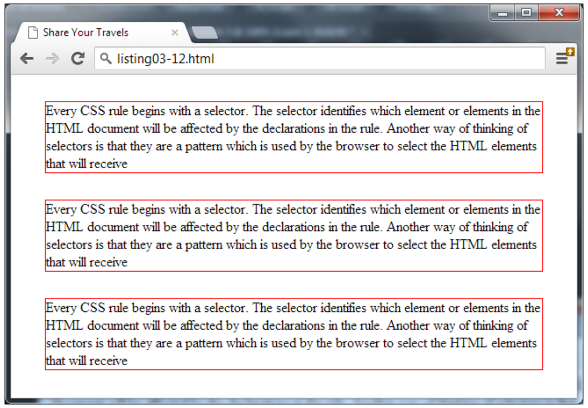

Share Your Travels
listing03-12.html

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;
}
```

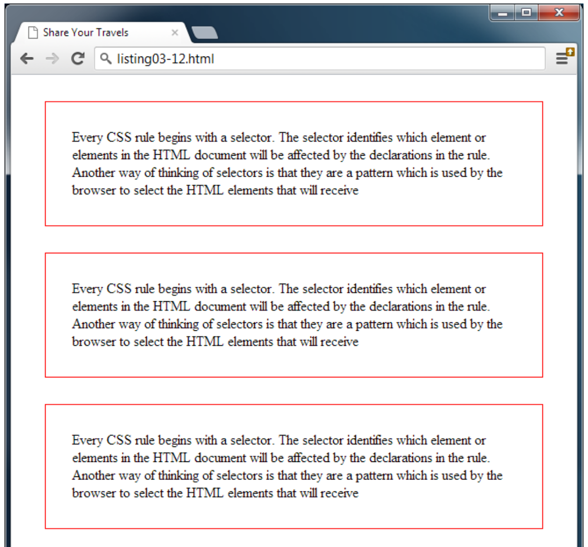

Share Your Travels
listing03-12.html

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;
}
```


Share Your Travels
listing03-12.html

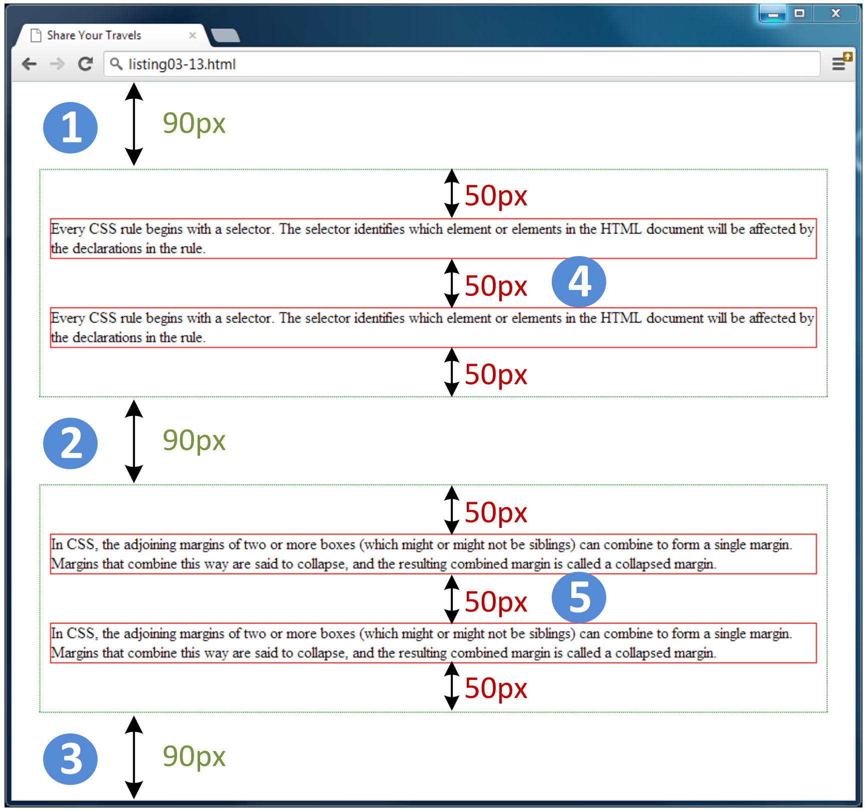
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML elements that will receive

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;
}
```

Collapsing Margins



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.

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.

In CSS, the adjoining margins of two or more boxes (which might or might not be siblings) can combine to form a single margin. Margins that combine this way are said to collapse, and the resulting combined margin is called a collapsed margin.

In CSS, the adjoining margins of two or more boxes (which might or might not be siblings) can combine to form a single margin. Margins that combine this way are said to collapse, and the resulting combined margin is called a collapsed margin.

```
<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;
}
```

2

If overlapping margins did not collapse, then margin space for wo4d be 50p (90pixels for the bottom margin of the first <div> + 90 pixels for the top margin of the second <div>), while the margins and for would be 100px.

However, as you can see this is not the case.

zeshan.khan@nu.edu.pk

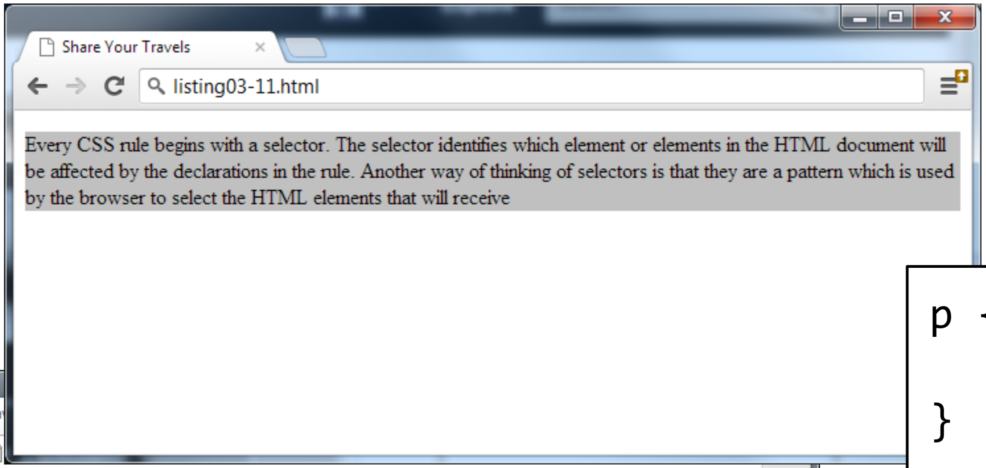
Collapsing Margins

- 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.
- To complicate matters even further, there are a large number of special cases in which adjoining vertical margins do not collapse.

Width and Height

- The width and height properties specify the size of the element’s content area.
- Defined for block elements

Width and Height

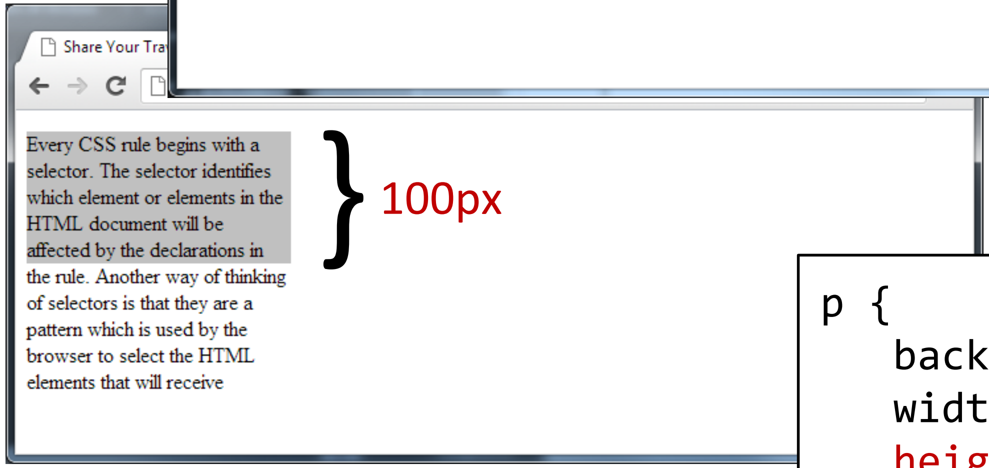


Share Your Travels

listing03-11.html

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 {
    background-color: silver;
}
```



Share Your Tra

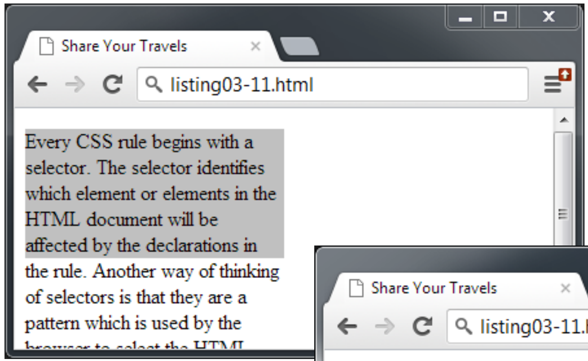
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML elements that will receive

100px

```
p {
    background-color: silver;
    width: 200px;
    height: 100px;
}
```

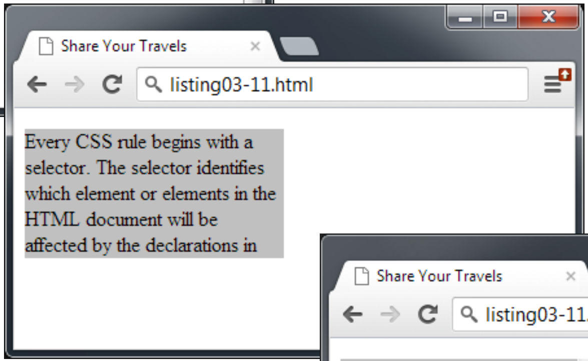
Overflow Property

`overflow: visible;`



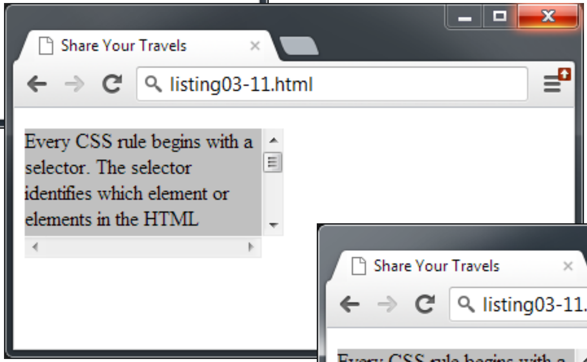
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern which is used by the browser to select the HTML

`overflow: hidden;`



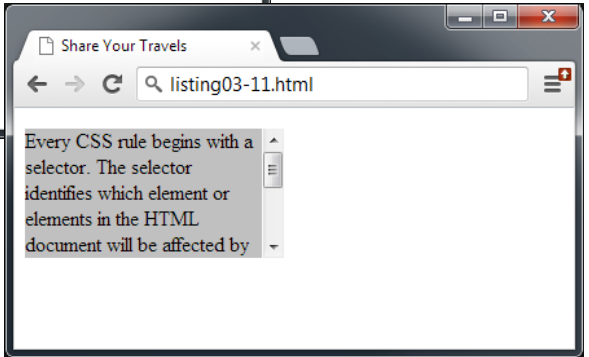
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in

`overflow: scroll;`



Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML

`overflow: auto;`



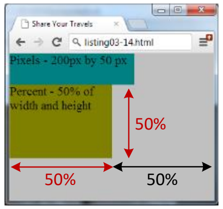
Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by

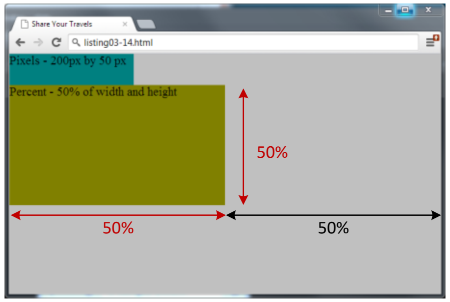
Sizing Elements

- When you use %, the size is relative to the size of the parent element.
- When you use ems, the size of the box is relative to the size of the text within it.
- The rationale behind using these relative measures is to make one's design scalable to the size of the browser or device that is viewing it.

```
<style>
  html,body {
     margin:0;
     width:100%;
     height:100%;
     background: silver;
  }
  .pixels {
     width:200px;
     height:50px;
     background: teal;
  }
  .percent {
     width:50%;
     height:50%;
     background: olive;
  }
```

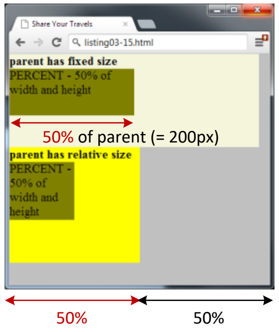
```
<body>
   <div class="pixels">
     Pixels - 200px by 50 px
   </div>
   <div class="percent">
     Percent - 50% of width and height
   </div>
</body>
```

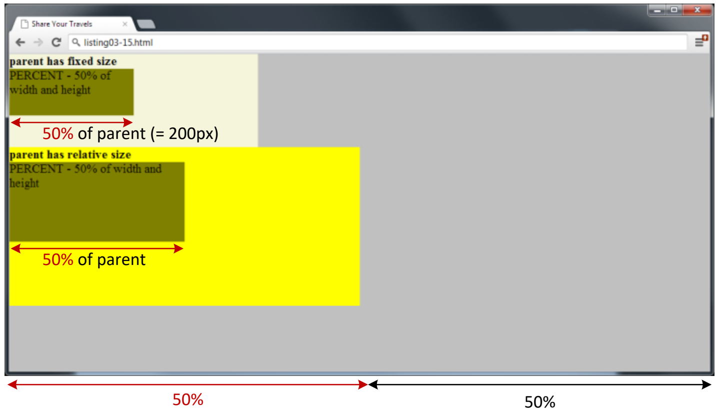




```
  .parentFixed {
     width:400px;
     height:150px;
     background: beige;
  }
  .parentRelative {
     width:50%;
     height:50%;
     background: yellow;
  }
</style>
```

```
<body>
<div class="parentFixed">
   <strong>parent has fixed size</strong>
   <div class="percent">
      PERCENT - 50% of width and height
   </div>
</div>
<div class="parentRelative">
   <strong>parent has relative size</strong>
   <div class="percent">
      PERCENT - 50% of width and height
   </div>
</div>
</body>
```



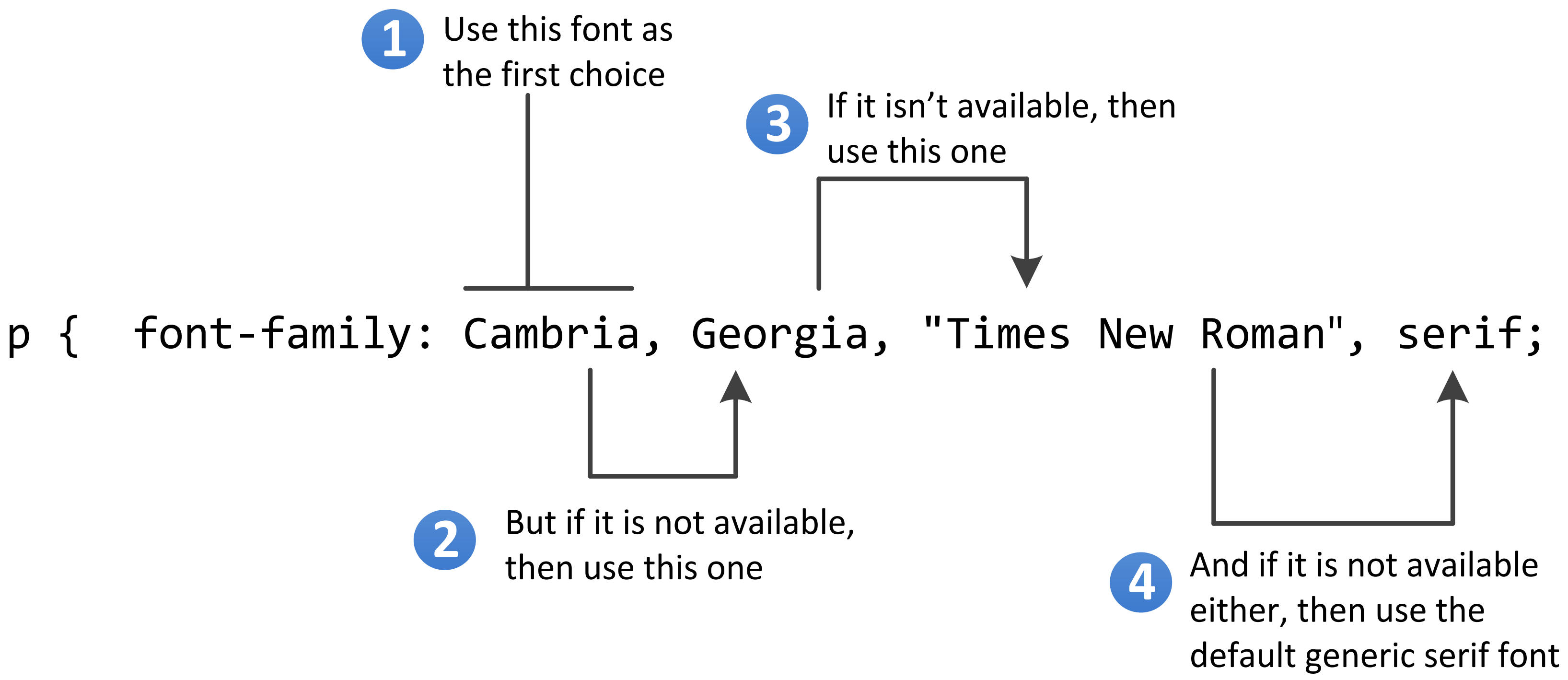


Example Here

Text Properties

- 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.

Specifying the Font-Family



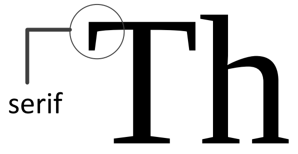
```
p {  font-family: Cambria, Georgia, "Times New Roman", serif;  }
```

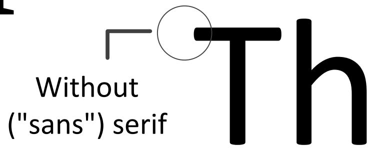
1. Use this font as the first choice
2. But if it is not available, then use this one
3. If it isn't available, then use this one
4. And if it is not available either, then use the default generic serif font

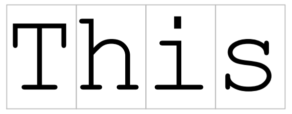
Generic Font-Family

- The font-family property supports five different generic families.
- The browser supports a typeface from each family.

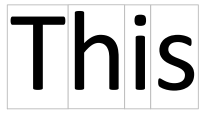
	Generic Font-Family Name
This	`serif`
This	`sans-serif`
This	`monospace`
This	`cursive`
This	`fantasy`







In a monospace font, each letter has the same width



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-face

- Over the past few years, the most recent browser versions have begun to support the **@font-face** selector in CSS.
- This selector allows you to use a font on your site even if it is not installed on the end user's computer.
- Due to the on-going popularity of open source font sites such as Google Web Fonts (http://www.google.com/webfonts) and Font Squirrel (http://www.fontsquirrel.com/), @font-face seems to have gained a critical mass of widespread usage.

Example Here

zeshan.khan@nu.edu.pk