

How Styles Interact

The Cascade

33

The Cascade: How Styles Interact

CSS

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

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

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

- inheritance,
- specificity, and
- location

34

The Cascade: How Styles Interact

Inheritance

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

Are inheritable:

- Font,
- color,
- list, and
- text properties

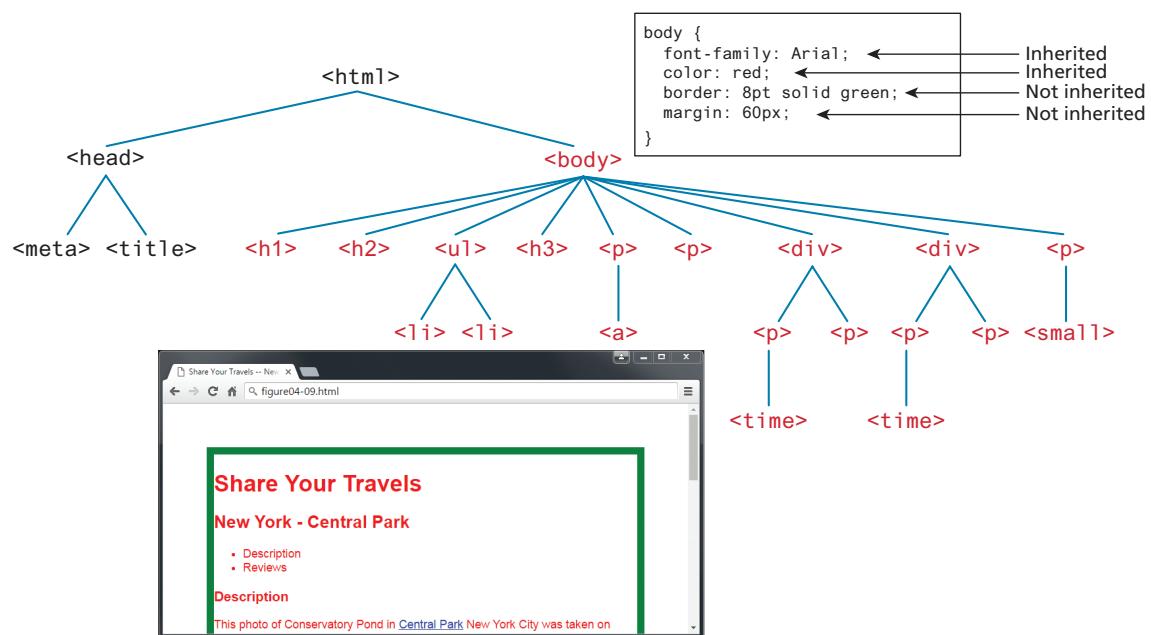
Not inheritable:

- layout,
- sizing,
- border,
- background, and
- spacing properties

35

The Cascade: How Styles Interact

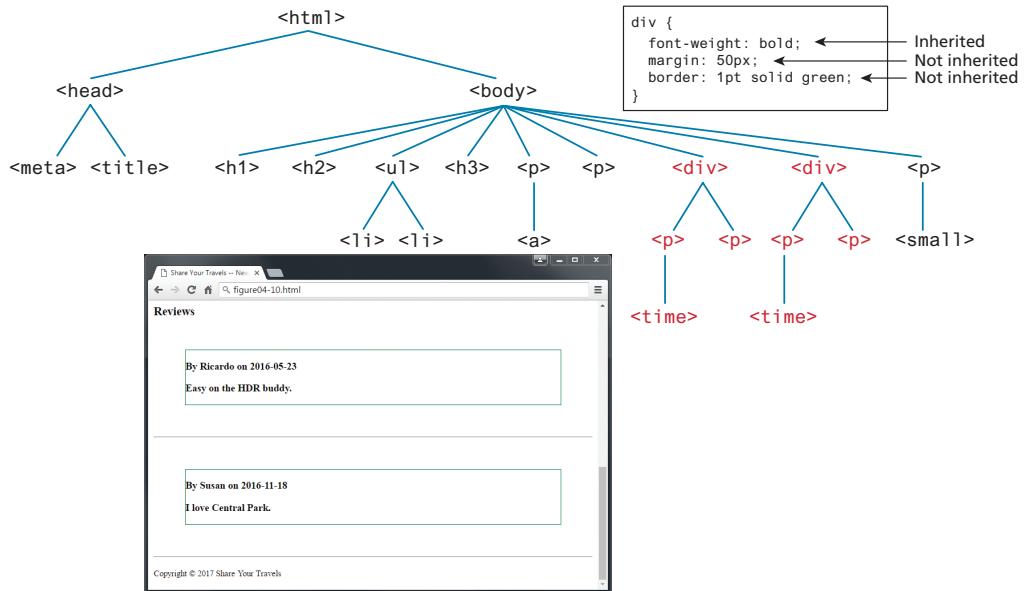
Inheritance



36

The Cascade: How Styles Interact

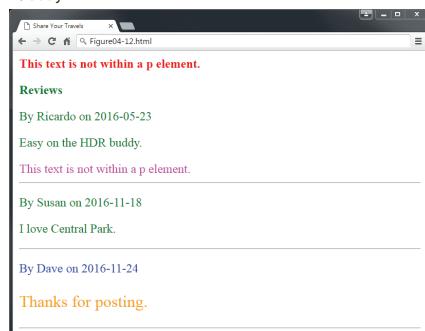
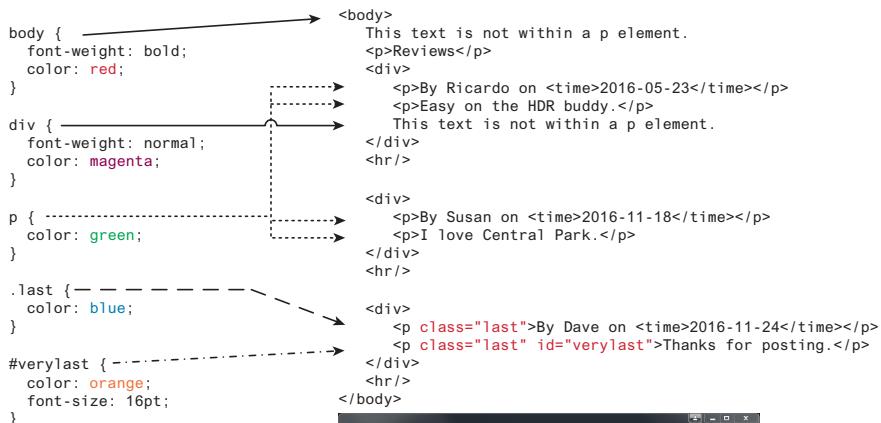
Inheritance



37

The Cascade: How Styles Interact

Specificity



38

The Cascade

Specificity Algorithm

div { color: green; }	Specificity Value 0001
-----------------------------	---------------------------

div form { color: orange; }	0002
-----------------------------------	------

.example { color: blue; } a[href\$=".pdf"] { color: blue; }	0010
--	------

#firstExample { color: magenta; }	0100
---	------

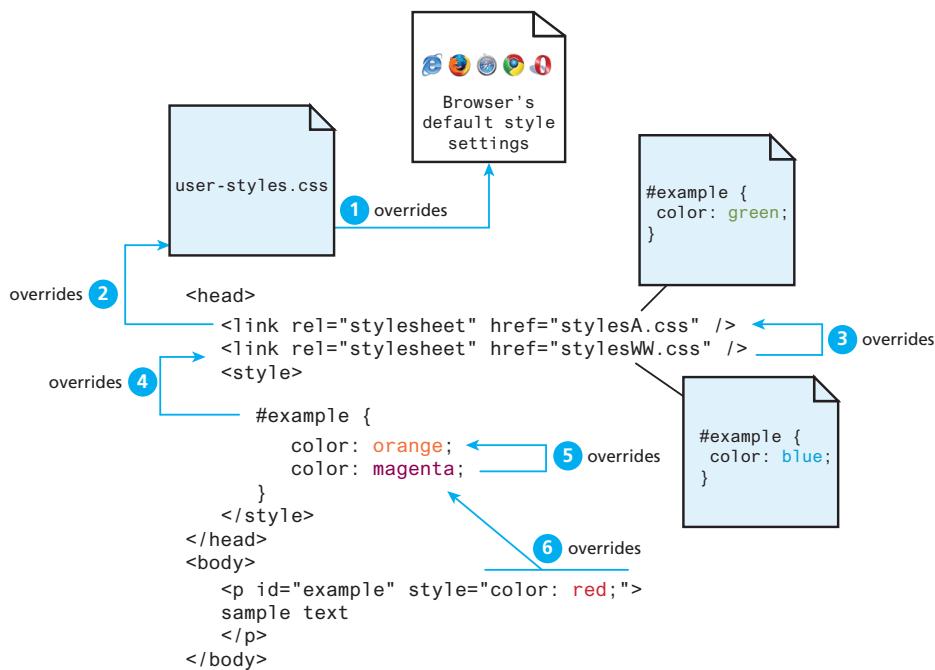
div #firstExample { color: grey; }	0101
--	------

<div style="color: red;">	1000
---------------------------	------

39

The Cascade: How Styles Interact

Location

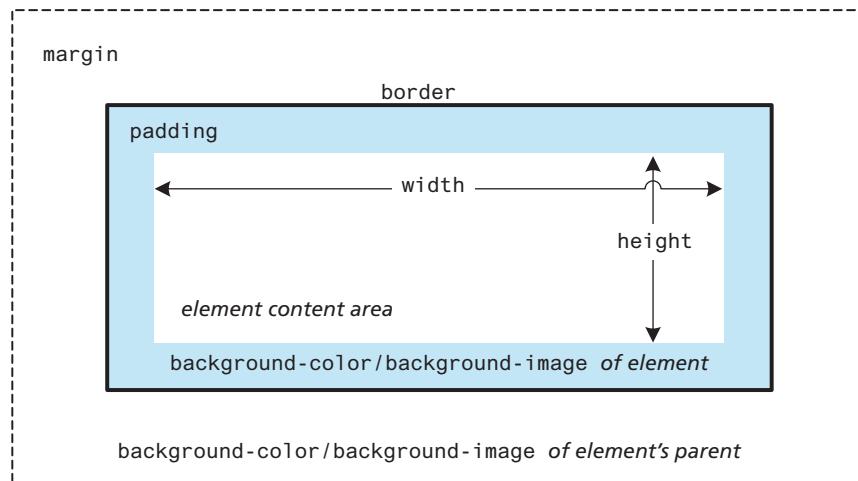


The Box Model

41

The Box Model

Overview



42

The Box Model

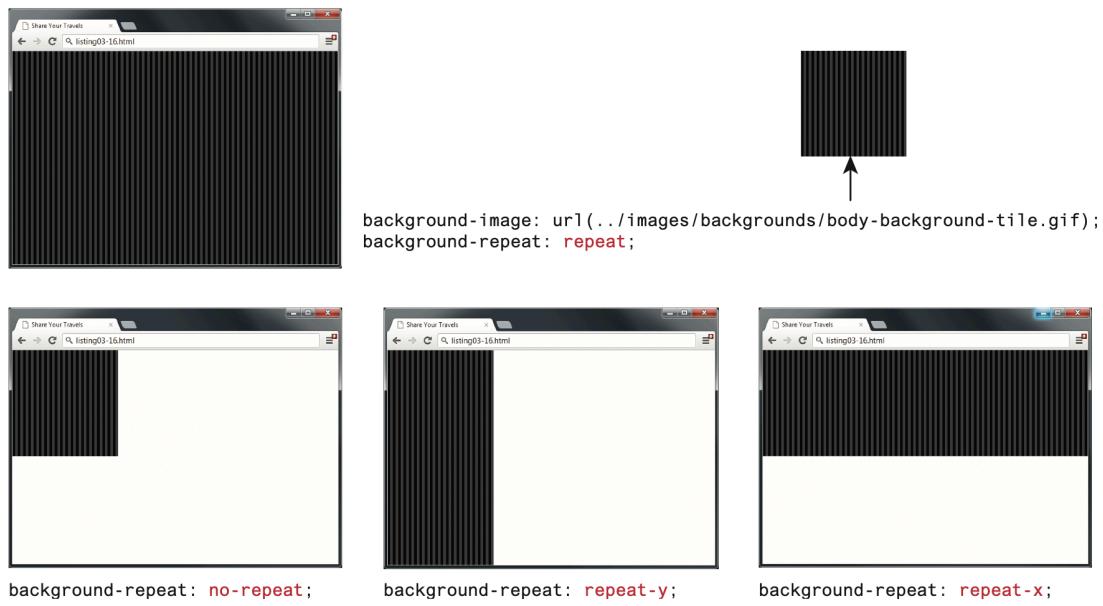
Background

- Background
- Background-attachment
- Background-color
- Background-image
- Background-position
- Background-repeat
- Background-size

43

The Box Model

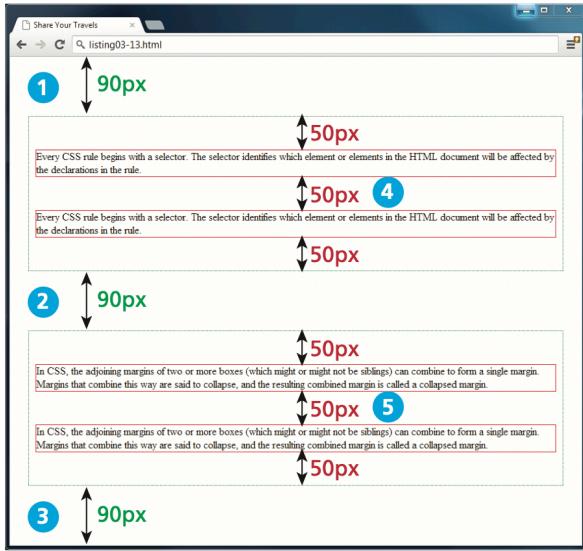
Background



44

The Box Model

Collapsing Margins



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

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

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

45

The Box Model

CSS TRBL Shortcut

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

Alternately, we can set all four sides at once:

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

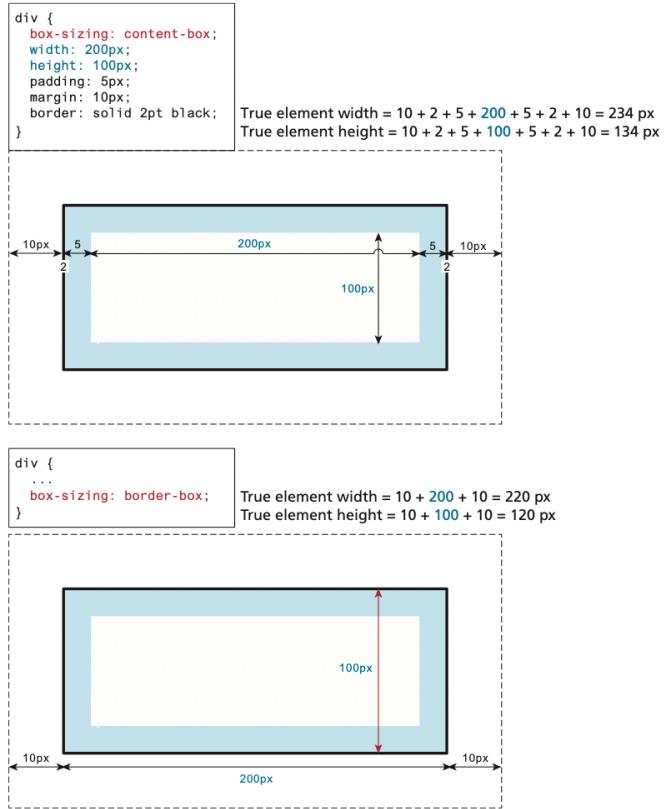
```
border-color: red green orange blue; /* sets 4 colors */
```



46

The Box Model

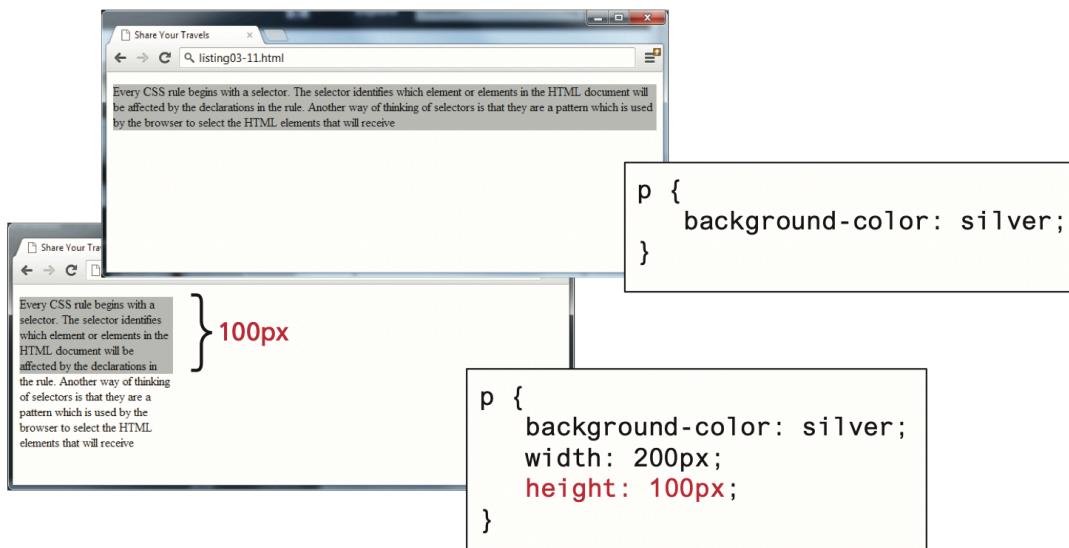
Box Dimensions



47

The Box Model

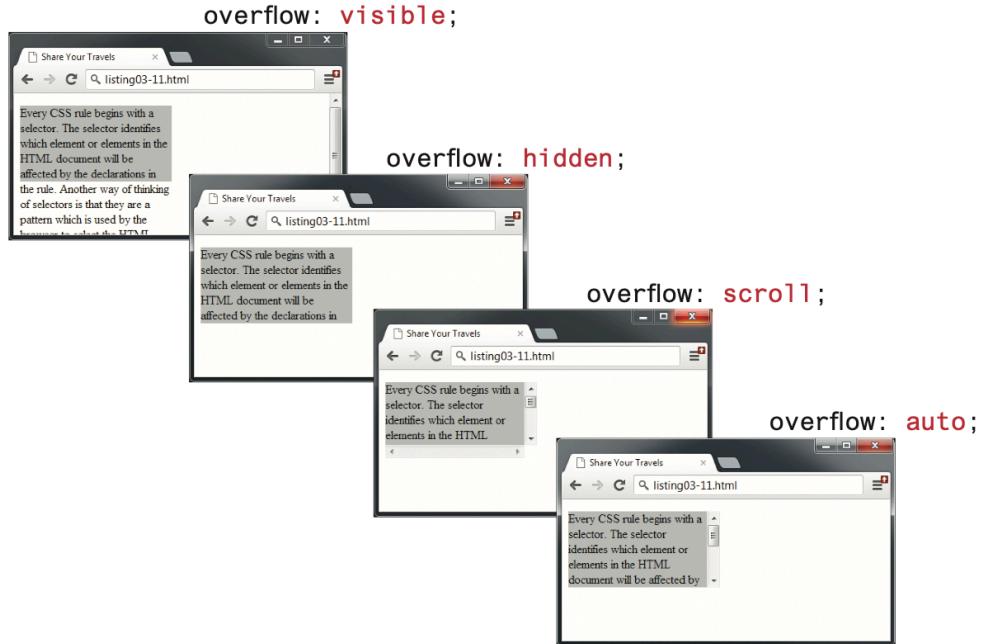
Limitations of Height Property



48

The Box Model

Overflow Property



49

CSS Text Styling

50

CSS Text Styling

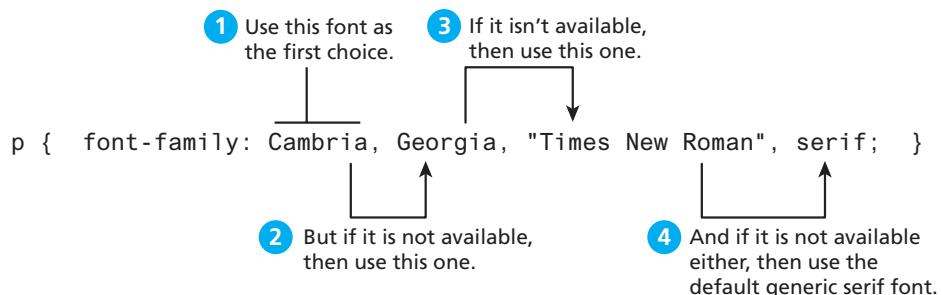
Font Family

Property	Description
font	A combined shorthand property that allows you to set the family, style, size, variant, and weight in one property. style weight variant size font-family
font-family	Specifies the typeface/font to use. More than one can be specified.
font-size	The size of the font in one of the measurement units
font-style	Specifies whether italic, oblique, or normal
font-variant	Specifies either small-caps text or none
font-weight	Specifies either normal, bold, bolder, lighter, or a value between 100 and 900 in multiples of 100, where larger number represents weightier (i.e., bolder) text.

51

CSS Text Styling

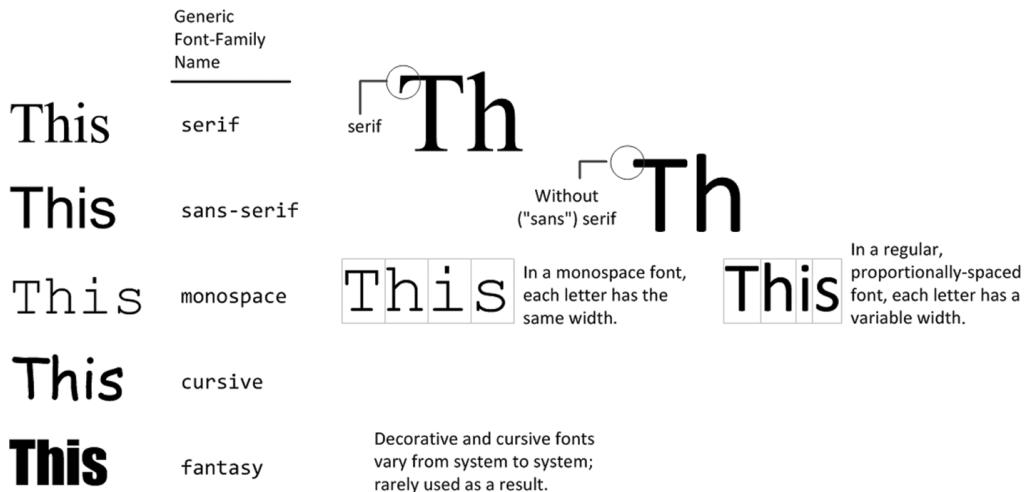
Specifying the Font Family



52

CSS Text Styling

Different Font Families



53

CSS Text Styling

Font Sizes

<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%; }           /* 1.0 x 16 = 16 */
p { font-size: 1em; }               /* 1.0 x 16 = 16 */
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>
  Browser's default text size is usually 16 pixels
  <p>100% or 1em is 16 pixels</p>
  <h3>125% or 1.125em is 18 pixels</h3>
  <h2>150% or 1.5em is 24 pixels</h2>
  <h1>200% or 2em is 32 pixels</h1>
</body>
```

54

CSS Text Styling

Paragraph Properties

Just as there are properties that affect the font in CSS, there are also a range of CSS properties that affect text independently of the font.

- letter-spacing
- line-height
- text-align
- text-decoration
- text-direction
- text-shadow
- ...

55

Normal Flow

56

Normal Flow

To understand CSS positioning and layout, it is essential that we understand this distinction as well as the idea of **normal flow**:

how the browser will normally display block-level elements and inline elements from left to right and from top to bottom

57

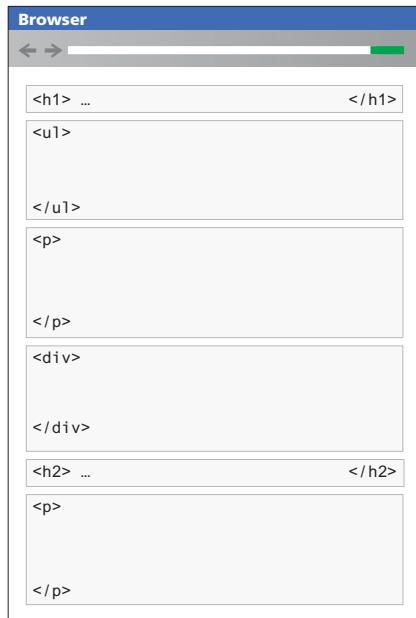
Normal Flow

- **Block-level elements** such as `<p>`, `<div>`, `<h2>`, ``, and `<table>` are each contained on their own line.
- **Inline elements** do not form their own blocks but instead are displayed within lines.

58

Normal Flow

Block-Level Elements



Each block exists on its own line and is displayed in normal flow from the browser window's top to its bottom.

By default each block-level element fills up the entire width of its parent (in this case, it is the <body>, which is equivalent to the width of the browser window).

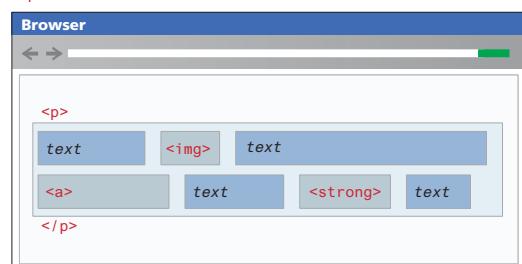
You can use CSS box model properties to customize, for instance, the width of the box and the margin space between other block-level elements.

59

Normal Flow

Inline Elements

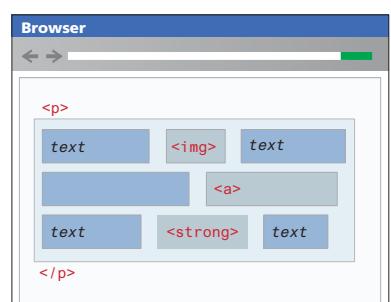
```
<p>
This photo  of Conservatory Pond in
<a href="http://www.centralpark.com/">Central Park</a> New York City
was taken on October 22, 2015 with a <strong>Canon EOS 30D</strong>
camera.
</p>
```



Inline content is laid out horizontally left to right within its container.

Once a line is filled with content, the next line will receive the remaining content, and so on.

Here the content of this <p> element is displayed on two lines.



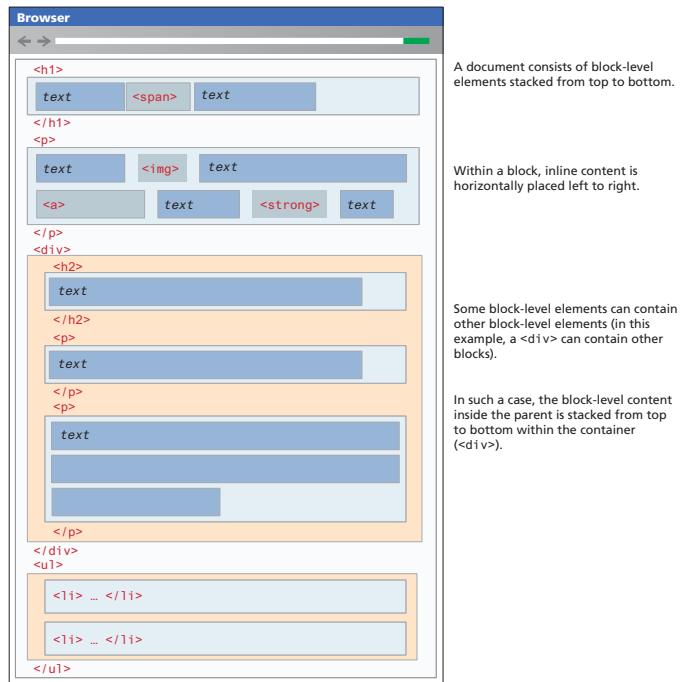
If the browser window resizes, then inline content will be “reflowed” based on the new width.

Here the content of this <p> element is now displayed on three lines.

60

Normal Flow

Block and Inline Elements



A document consists of block-level elements stacked from top to bottom.

Within a block, inline content is horizontally placed left to right.

Some block-level elements can contain other block-level elements (in this example, a <div> can contain other blocks).

In such a case, the block-level content inside the parent is stacked from top to bottom within the container (<div>).

61

Positioning Elements

62

Positioning Elements

- **absolute** The element is removed from normal flow and positioned in relation to its nearest positioned ancestor.
- **fixed** The element is fixed in a specific position in the window even when the document is scrolled.
- **relative** The element is moved relative to where it would be in the normal flow.
- **static** The element is positioned according to the normal flow. This is the default.

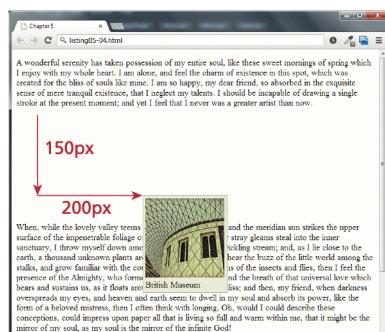
63

Positioning Elements

Relative Positioning



```
<p>A wonderful serenity has taken possession of my ...</p>
<figure>
  
  <figcaption>British Museum</figcaption>
</figure>
<p>When, while the lovely valley ...</p>
```

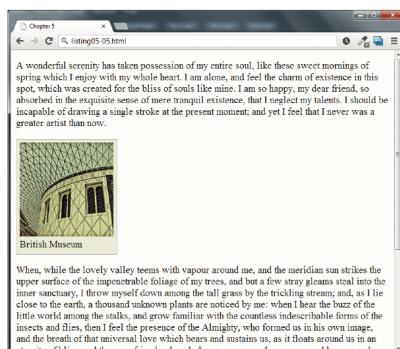


```
figure {
  border: 1px solid #A8A8A8;
  background-color: #EDEDED;
  padding: 5px;
  width: 150px;
  position: relative;
  top: 150px;
  left: 200px;
}
```

64

Positioning Elements

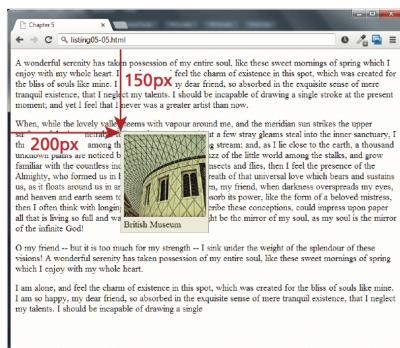
Absolute Positioning



```
<p>A wonderful serenity has taken possession of my ...
```

```
<figure>
  
  <figcaption>British Museum</figcaption>
</figure>
```

```
<p>When, while the lovely valley ...
```



```
figure {
  margin: 0;
  border: 1px solid #A8A8A8;
  background-color: #EDEDED;
  padding: 5px;
  width: 150px;
  position: absolute;
  top: 150px;
  left: 200px;
}
```

65

Positioning Elements

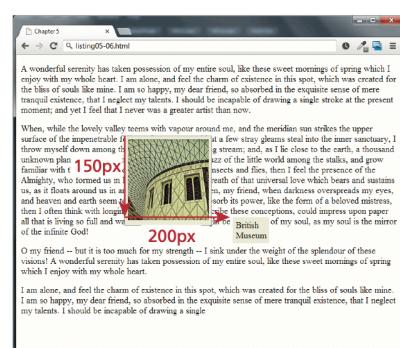
Absolute Positioning is relative to nearest positioned ancestor



```
<p>A wonderful serenity has taken possession of my ...
```

```
<figure>
  
  <figcaption>British Museum</figcaption>
</figure>
```

```
<p>When, while the lovely valley ...
```



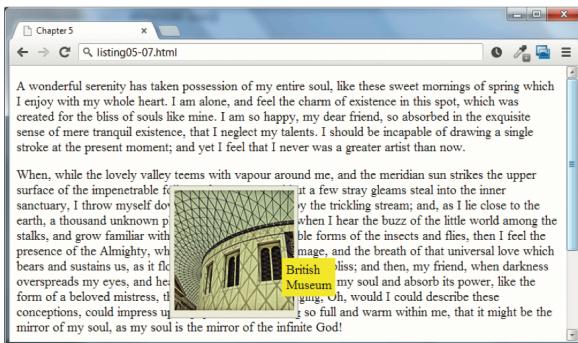
```
figure {
  margin: 0;
  border: 1px solid #A8A8A8;
  background-color: #EDEDED;
  padding: 5px;
  width: 150px;
  position: absolute;
  top: 150px;
  left: 200px;
}

figcaption {
  background-color: #EDEDED;
  padding: 5px;
  position: absolute;
  top: 150px;
  left: 200px;
}
```

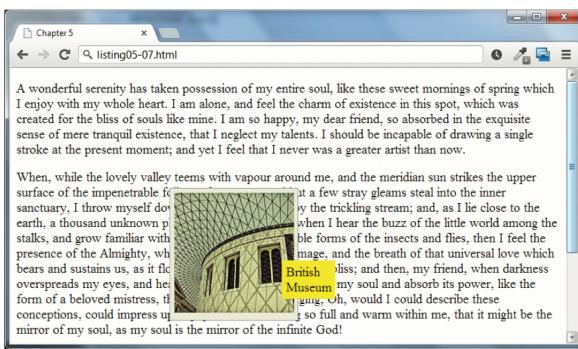
66

Positioning Elements

Z-Index



```
figure {  
    position: absolute;  
    top: 150px;  
    left: 200px;  
}  
figcaption {  
    position: absolute;  
    top: 90px;  
    left: 140px;  
}
```

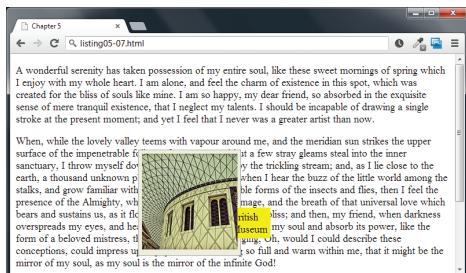


```
figure {  
    ...  
    z-index: 5;  
}  
figcaption {  
    ...  
    z-index: 1;  
}
```

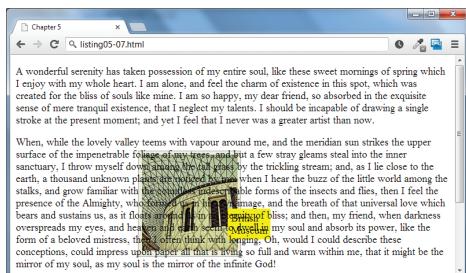
67

Positioning Elements

Z-Index



```
figure {  
    ...  
    z-index: 1;  
}  
figcaption {  
    ...  
    z-index: -1;  
}
```



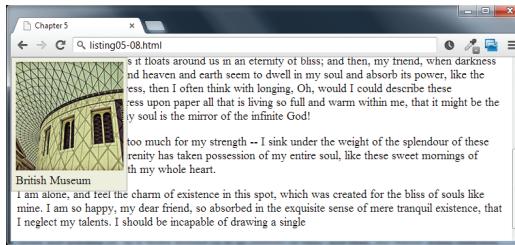
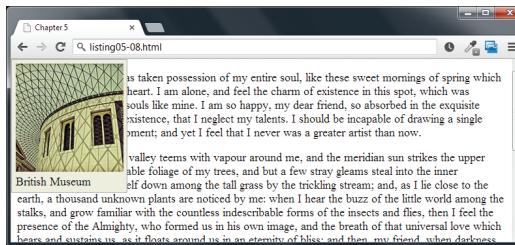
```
figure {  
    ...  
    z-index: -1;  
}  
figcaption {  
    ...  
    z-index: 1;  
}
```

68

Positioning Elements

Fixed Position

```
figure {  
    ...  
    position: fixed;  
    top: 0;  
    left: 0;  
}
```



69

Floating Elements

70

Floating Elements

It is possible to displace an element out of its position in the normal flow via the CSS **float** property

- An element can be floated to the left or floated to the right .
- it is moved all the way to the far left or far right of its containing block and the rest of the content is “reflowed” around the floated element

71

Floating Elements

The figure block has a width of 150px.

Notice that a floated block-level element should have a width specified.

figure {
width: 150px;
float: left;

figure {
width: 150px;
float: right;
margin: 10px;

72

Floating Elements

Floating Multiple Items Side by Side

The first screenshot shows a page with six floated images (Westminster, Emirates Stadium, Albert Hall, British Museum, Wellington Monument, and Lewes Castle) and their captions. The second screenshot shows the same layout but with the first two images removed. The third screenshot shows the full layout again.

```
<article>
<figure>

<figcaption>Westminster</figcaption>
</figure>
<figure>

<figcaption>Emirates Stadium</figcaption>
</figure>
<figure>

<figcaption>Albert Hall</figcaption>
</figure>
<figure>

<figcaption>British Museum</figcaption>
</figure>
<figure>

<figcaption>Wellington Monument</figcaption>
</figure>
<figure>

<figcaption>Lewes Castle</figcaption>
</figure>
<p>When, while the lovely valley teems ..</p>
```

```
figure {
    ...
    width: 150px;
    float: left;
}
```

73

Floating Elements

Floating Multiple Items Side by Side

Stop elements from flowing around a floated element by using the **clear** property

A red arrow points to the 'British Museum' image, which is marked with a red border. Another red arrow points to the 'Wellington Monument' image, which is also marked with a red border. A red bracket on the left side of the page groups these two images together, indicating they are being cleared.

```
.first { clear: left; }
```

```
<article>
<figure>

<figcaption>Westminster</figcaption>
</figure>
<figure>

<figcaption>Emirates Stadium</figcaption>
</figure>
<figure>

<figcaption>Albert Hall</figcaption>
</figure>
<figure class="first">

<figcaption>British Museum</figcaption>
</figure>
<figure>

<figcaption>Wellington Monument</figcaption>
</figure>
<figure>

<figcaption>Lewes Castle</figcaption>
</figure>
<p class="first">When, while the lovely ...</p>
```

74

Floating Elements

Clear property

- **left** The left-hand edge of the element cannot be adjacent to another element.
- **right** The right-hand edge of the element cannot be adjacent to another element.
- **both** Both the left-hand and right-hand edges of the element cannot be adjacent to another element.
- **none** The element can be adjacent to other elements.

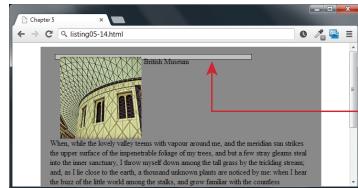
75

Floating Elements

Containing Floats

Another problem that can occur with floats is when an element is floated within a containing block that contains only floated content. In such a case, the containing block essentially disappears

```
<article>
  <figure>
    
    <figcaption>British Museum</figcaption>
  </figure>
  <p class="first">When, while the lovely valley ...</p>
</article>
```

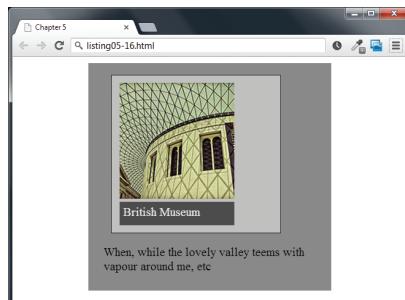


```
figure img {
  width: 170px;
  margin: 0 5px;
  float: left;
}
figure figcaption {
  width: 100px;
  float: left;
}
figure {
  border: 1px solid #262626;
  background-color: #c1c1c1;
  padding: 5px;
  width: 400px;
  margin: 10px;
}
.first { clear: left; }
```

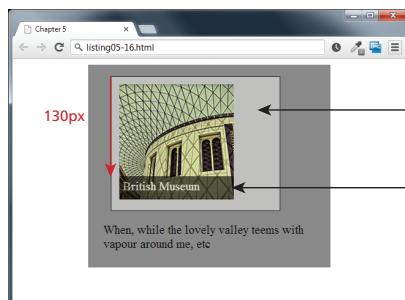
76

Floating Elements

Overlaying and Hiding Element



```
figure {  
    border: 1pt solid #262626;  
    background-color: #c1c1c1;  
    padding: 10px;  
    width: 200px;  
    margin: 10px;  
}  
  
figcaption {  
    background-color: black;  
    color: white;  
    opacity: 0.6;  
    width: 140px;  
    height: 20px;  
    padding: 5px;  
}
```



```
figure {  
    ...  
    position: relative;  
}  
figcaption {  
    ...  
    position: absolute;  
    top: 130px;  
    left: 10px;  
}
```

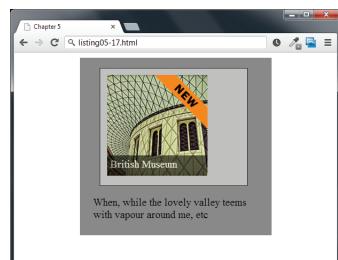
This creates the positioning context.
This does the actual move.

77

Floating Elements

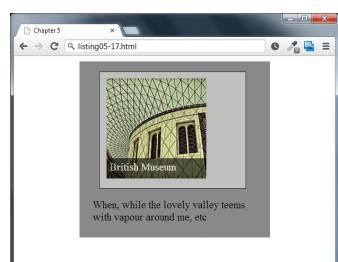
Using display

```
<figure>  
      
    <figcaption>British Museum</figcaption>  
      
</figure>
```



```
.overlaid {  
    position: absolute;  
    top: 10px;  
    left: 10px;  
}  
  
new-banner.png
```

Transparent area



```
.overlaid {  
    position: absolute;  
    top: 10px;  
    left: 10px;  
    display: none;  
}  
  
.hide {  
    display: none;  
}  
  
<img ... class="overlaid hide"/>
```

This hides the overlaid image.
This is the preferred way to hide: by adding this additional class to the element. This makes it clear in the markup that the element is not visible.

78

Floating Elements

Comparing visibility with display



```
figure {  
...  
display: auto;  
}
```

```
figure {  
...  
display: none;  
}
```

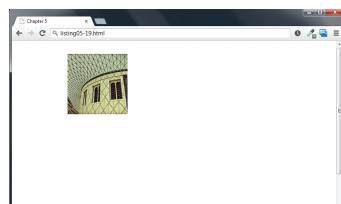
```
figure {  
...  
visibility: hidden;  
}
```

79

Floating Elements

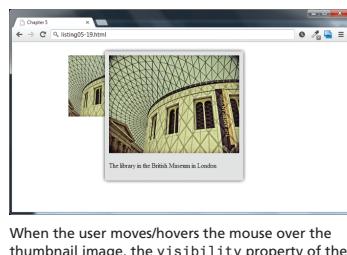
Using Hover with display

```
<figure class="thumbnail">  
    
  <figcaption class="popup">  
      
    <p>The library in the British Museum in London</p>  
  </figcaption>  
</figure>
```



When the page is displayed, the larger version of the image, which is within the `<figcaption>` element, is hidden.

```
figcaption.popup {  
  padding: 10px;  
  background: #e1e1e1;  
  position: absolute;  
  
  /* add a drop shadow to the frame */  
  box-shadow: 0 0 15px #A9A9A9;  
  
  /* hide it until there is a hover */  
  visibility: hidden;  
}
```



When the user moves/hovers the mouse over the thumbnail image, the `visibility` property of the `<figcaption>` element is set to `visible`.

```
figure.thumbnail:hover figcaption.popup {  
  position: absolute;  
  top: 0;  
  left: 100px;  
  
  /* display image upon hover */  
  visibility: visible;  
}
```

80

Styling Tables

81

Styling Tables

Table Borders

The figure consists of three screenshots of a web browser window titled "figure05-08.html". Each screenshot displays a table with the following data:

19th Century French Paintings		
Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Total Number of Paintings		4

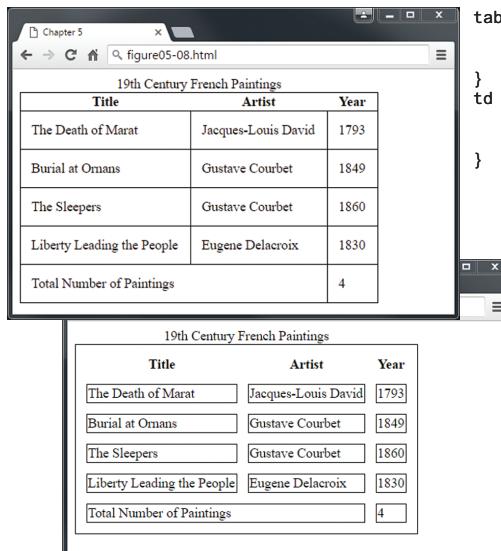
The first screenshot shows a solid black border around the entire table. The second screenshot shows a solid black border only on the top and bottom rows, with no border between the columns. The third screenshot shows a collapse border, where the borders between the cells are removed, creating a more open appearance.

```
table {  
    border: solid 1pt black;  
}  
  
table {  
    border: solid 1pt black;  
}  
td {  
    border: solid 1pt black;  
}  
  
table {  
    border: solid 1pt black;  
    border-collapse: collapse;  
}  
td {  
    border: solid 1pt black;  
}
```

82

Styling Tables

Table Borders

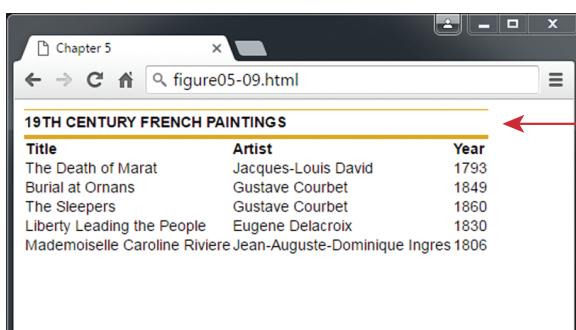


```
table {  
    border: solid 1pt black;  
    border-collapse: collapse;  
}  
td {  
    border: solid 1pt black;  
    padding: 10pt;  
}  
  
table {  
    border: solid 1pt black;  
    border-spacing: 10pt;  
}  
td {  
    border: solid 1pt black;  
}
```

83

Styling Tables

Boxes



```
caption {  
    font-weight: bold;  
    padding: 0.25em 0 0.25em 0;  
    text-align: left;  
    text-transform: uppercase;  
    border-top: 1px solid #DCA806;  
}  
  
table {  
    font-size: 0.8em;  
    font-family: Arial, sans-serif;  
    border-collapse: collapse;  
    border-top: 4px solid #DCA806;  
    border-bottom: 1px solid white;  
    text-align: left;  
}
```

84

Styling Tables

Boxes

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
thead tr {  
    background-color: #CACACA;  
}  
th {  
    padding: 0.75em;  
}
```

85

Styling Tables

Boxes

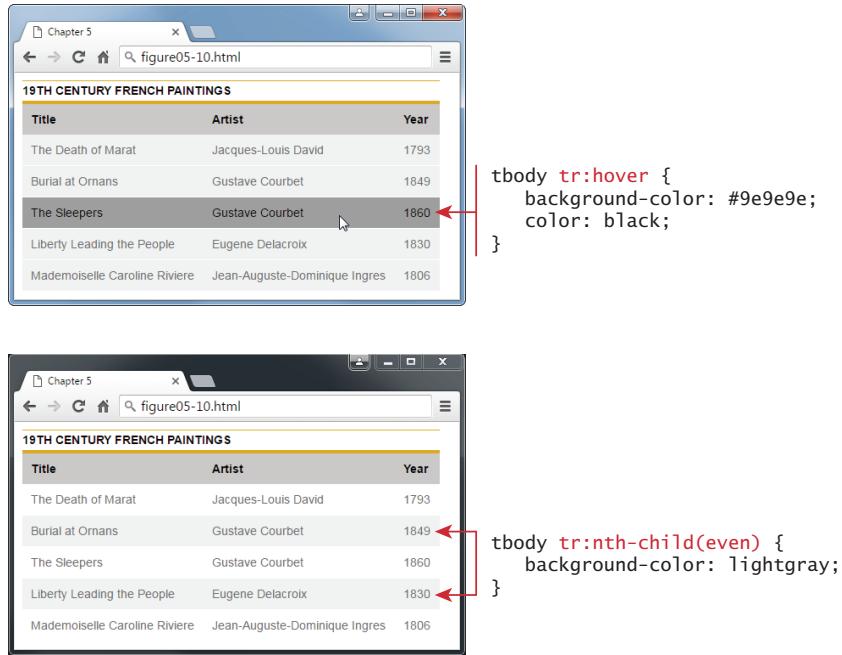
Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
tbody tr {  
    background-color: #F1F1F1;  
    border-bottom: 1px solid white;  
    color: #6E6E6E;  
}  
tbody td {  
    padding: 0.75em;  
}
```

86

Styling Tables

Hover & Zebras



The screenshots illustrate two different CSS approaches for styling table rows:

Top Screenshot (tbody tr:hover):

19TH CENTURY FRENCH PAINTINGS		
Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

Bottom Screenshot (tbody tr:nth-child(even)):

19TH CENTURY FRENCH PAINTINGS		
Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

CSS Rules:

tbody tr:hover {
background-color: #9e9e9e;
color: black;
}

tbody tr:nth-child(even) {
background-color: lightgray;
}

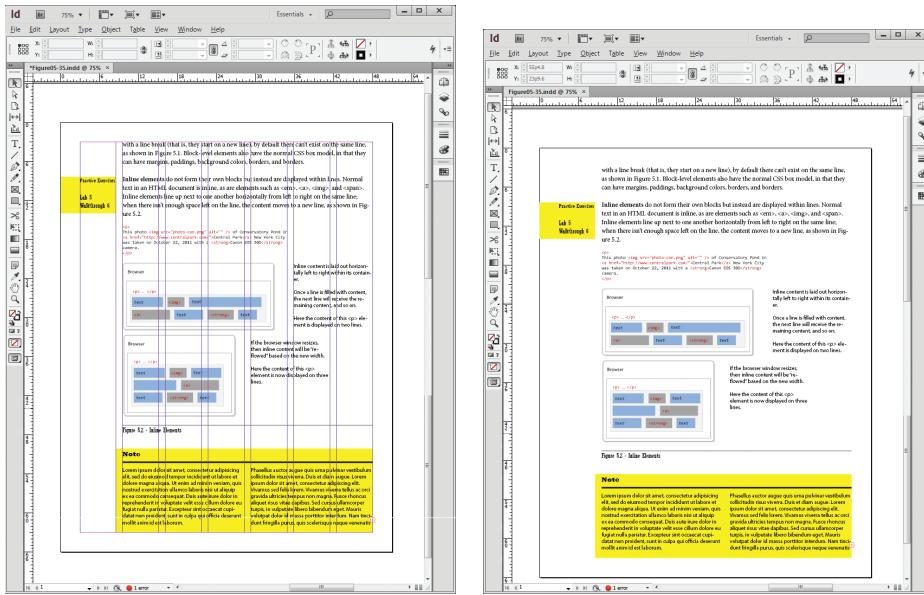
87

CSS Frameworks

88

CSS Frameworks and Preprocessors

Grid in print design



Most page design begins with a grid. In this case, a seven-column grid is being used to layout page elements in Adobe InDesign.

Without the gridlines visible, the elements on the page do not look random, but planned and harmonious.

89

CSS Frameworks and Preprocessors

Using Bootstrap

```
<head>
  <link href="bootstrap.css" rel="stylesheet">
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-2">
        left column
      </div>
      <div class="col-md-7">
        main content
      </div>
      <div class="col-md-3">
        right column
      </div>
    </div>
  </div>
</body>
```

90

CSS Frameworks and Preprocessors

CSS Preprocessors

