



Member Login

User Name:



Password:



Remember Me

Login

www.htmlicious.com

CSS: A Power Course for Web Artists

Designing and styling websites with CSS3

CSS: A Power Course for Web Artists

Team Htmleceous (Babeeta K)



CSS: A Power Course for Web Artists Copyright © 2015 **Htmleceous**

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews. No patent liability is assumed with respect to the use of information contained herein. Although every precaution has been taken in the preparation of this book, neither the author, nor Htmleceous, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Htmleceous cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Book Code: **H0001**

<http://www.htmleceous.com>

About Htmleceous Htmleceous is a group of web designers and developers. They specialize in website

design & development, mobile website & application development, graphic design, logo & corporate branding, search engine optimization, and search engine marketing.

Preface

Why This Book

CSS (Cascading Style Sheet), the styling language used for websites, since its inception in 1996 has come a long way. Its latest incarnation is CSS3 that is completely compatible with the earlier specifications of CSS. This book is aimed at those who just started working on web design, CSS programming in particular. After completing this book, you would be able to style any website.

Available as eCourse

This eBook is also available as *eCourse*. You will receive **free updates** when additional chapters, tutorials, exercises, projects, and bonus content are added to the *eCourse*. To know more visit: <http://bit.ly/cssecourse>.

What You Will Learn This book will teach you to create complex layouts, effects, animations, transitions, transformations, and much more with CSS code. The CSS properties are explained with examples and screen captures. You will gain skills by completing the exercises provided in the book.

What You Need

You need a computer, internet connection, and a text editor to complete the examples and exercises given in the book.

How This Book Is Structured

This book contains 9 chapters, 150+ code listings, and 5 tutorials. Exercises [30+] are given at the end of each chapter for practice. The following table summarizes the content of the book.

Content	Description
---------	-------------

Chapter 1: Getting Started with CSS	This Chapter stats with the introduction of CSS and CSS syntax. You will learn to create and apply CSS styles.
Chapter 2: Identifying what to Select – Selectors	This Chapter deals with selectors. In the Chapter, core CSS selectors are explained.
Chapter 3: Working with the Box Model	This Chapter explains the box model and the properties that are used to style the box.
Chapter 4: Working with the Borders and Backgrounds	This Chapter explain the techniques to create borders around boxes to properly visualize the padding and margin spaces.

Chapter 5: **Working with** **Color**

This chapter deals with colors. You will learn to set foreground color, background color, and opacity for the elements.

Chapter 6: Formatting Text

The formatting of text is very important to create a webpage that appeals. In this Chapter, you will learn to use of the properties that you can use to format text.

Chapter 7: Styling Lists, Table, Forms, and Cursors

In this Chapter, you will learn how to change appearance of the form elements, format tables, and specify bullet point styles for the lists.

Chapter 8: Creating Layout	In this Chapter teaches you how to control the position of the element on a page and how to create different types of layouts using CSS including the multicolumn layout.
Chapter 9: Animation, Transformations, and Transitions	This chapter deals with animation, transformations, and transitions. You will learn how to apply animation to the elements of a page using CSS.
Tutorial 1	Styling a table.
Tutorial 2	Styling a table with rounded corners.
Tutorial 3	Styling a login form.
Tutorial 4	Styling a fixed width layout.
Tutorial 5	Creating a liquid layout.

What are the main features of the book?

- Beautifully designed and formatted chapters with screenshots and illustrations to help you quickly learn CSS programming.
- Less theory more practical. Saves many hours of struggle and pain.
- The important terms are italicized and color coded so that you never miss them.
- Covers CSS3 using HTML5 template.
- 9 chapters written in an easy to understand language.
- 150+ code listings, and 5 tutorials.
- 30+ Exercises for practice.
- Full source code included.

Downloading the Resources

This book is sold via multiple sales channels. If you don't have access to the source code used in this book, you can place a request for the resources by visiting the following link: <http://bit.ly/htmlcontact>. Please mention "**Resources-H001**" in the subject line.

Customer Support

At Htmleceous we believe support is personal. Our technical team is

always ready to take care of your technical query. If you have any problem with the technical aspect of the book, navigate to <http://bit.ly/htmlcontact> and let us know about your query. Please mention "*Technical Query-H0001*" in the subject line. We will do our best to resolve your queries.

Reader Feedback

Your feedback is always welcome. Your feedback is critical to our efforts at Htmleceous and it will help us in developing quality titles in the future. To send the feedback, visit <http://bit.ly/htmlcontact>. Please mention "*Feedback-H0001*" in the subject line.

Errata

We take every precaution while preparing the content of the book but mistakes do happen. If you find a mistake in this book general or technical, we would be happy that you report it to us so that we can mention it in the errata section of the book's online page. If you find any errata, please report them by visiting the following link: <http://bit.ly/htmlcontact>. This will help the other readers from frustration. Once your errata is verified, it will appear in the errata section of the book's online page. Please mention "*Errata-H0001*" in the subject line.

Stay Connected

Stay connected with us through Twitter ([@htmleceous](#)) and Facebook (<https://www.facebook.com/Htmleceous>) to know the latest updates about our products, information about books, and other related information.

1

Getting Started With CSS

Chapter 1: Getting Started With CSS

CSS stands for *Cascading Style Sheets*. HTML was never intended to contain tags for formatting a web page. CSS styles feature was added in HTML 4.0 to address this issue. To understand how CSS works, imagine that there is an *invisible box* around every HTML element. CSS allows you to change the appearance of the box using the CSS rules. In this chapter, I will show you how to create and apply CSS styles.

Key Points & Concepts

- What you need
- Getting familiar with CSS Syntax
- Understanding the Box model
- Creating internal, external, and inline style sheets
- Understanding how style sheet cascade
- Understanding the inheritance model

What you need?

You can use any *text editor* (even *Windows Notepad*) to write the *HTML* and *CSS* code written in this book. There are many free and paid editors available on the internet that you can download and use. *Notepad++* is one such editor that you can use to write the code. It is *open source* and supports many languages including *HTML* and *CSS*. You can download it from the following link: <http://notepad-plus-plus.org>.

I personally like *Brackets* from *Adobe*. It is a community guided *open source* text editor and allows you to easily design a web page. It includes lots of visual tools that help in the web design process. You can download *Brackets* from the following link: <http://brackets.io>.

Defining a Style

CSS declaration is a rule that tells a web browser how to *format something* on a web page. For example, create a *blue border* around an image, add a drop-shadow to the box, create gradients, and so on. A

style is made up of two elements: the web page element [*referred to as selector*] that the browser formats and the formatting instructions [*referred to as declaration block*].

Now, for example, if you want to set the *font color* and the *font size* for all paragraphs on a web document, you would write the following CSS rule:

```
p { color: blue; font-size: 14px;}
```

The above rule tells browser that change the *color* of the text in all paragraphs to *blue* and set the *size* of the text to *14 pixels*. Table 1.1 shows the description of each element of the CSS rule.

Table 1.1: Description of CSS rules	
Element	Description
Selector	The selector [p in this case] tells the browser which element(s) on a page to style.
Declaration Block	The code between the curly brackets [{ and }] is a declaration block.
Declaration	Between the opening and closing curly brackets, you can add one or more declarations. Every declaration has two parts, a property and a value separated by a colon. The declaration ends with a semi-colon.
Property	The properties in CSS are used to create wide range of formatting options. A property can be a single word or it can be few hyphenated words such as overflow, font-size, list-style-type, and so on.
Value	Values are assigned to the properties.

In the above example, *p* is a selector whereas *color:blue;* and *font-size:14px;* are declarations. In the first declaration, *color* is property and *red* is value. In second declaration, *font-size* is *property* and *14px* is *value*. The area inside { and } is *declaration block*.

To make CSS more readable, you can put declaration on *each line*. In the following code, the content in all *<p>* elements will be *center-aligned*, with the text color set to *blue*:

```
p {  
color: blue;  
text-align: center;  
  
}
```

CSS Comments

Comments are used to explain the code. They help you in understanding the code when you edit it at a later stage. If you are part of a team, comments help your team mates to understand the code you have written. You must make a habit of putting comments in your code.

A CSS comment starts with a `/*` and ends with `*/`.

```
p {  
color: blue;  
/* Sets the color of the text to blue */  
text-align: center;  
  
}
```

The comments can also span into multiple lines.

```
/* This is a multiline  
comment */
```



Comments

The comments are ignored by the browser.

Understanding the Box Model

The *CSS box model* is a *box* that wraps around the *HTML* elements. Each

element in a webpage is represented by a rectangular box. Each of these boxes are described via the *box* model. The box consists of *margins* [marked as *A* in Figure 1], *borders* [marked as *B*], *padding* [marked as *C*], and actual content [marked as *D*]. Table 1.2 shows the different element of a box.

Table 1.2: Elements of a box	
Element	Description
Content	The text and images appear here.
Padding	Clears a transparent area around the content.
Border	The border goes around the padding and content.
Margin	Like paddings, margin clears a transparent area outside the margin.

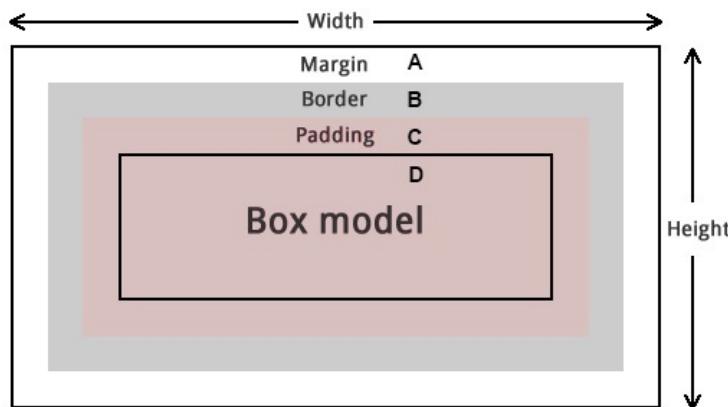


Figure 1

The *content* area is the area where the *real content* of the element lives. The *padding* area extends the *content* area and *add space* between the *content* and the *eventual borders* surrounding the content. The *border* area extends the *padding* area. It controls the area *inside the border edge*. The *margin* area extends the *border* area and adds space to separate an *element* and its neighboring elements.

The *box* model is discussed in detail in a later chapter.

Using External CSS

An *external style sheet* is a *text file* containing all the *CSS rules*. The text file always ends with an extension *.css*. Once you create an external style sheet, you must *connect* it with the web page that you intend to format. You can attach a style sheet to a web page by using the *<link>* html element. The syntax of the *<link>* tags varies a bit depending on whether you are using *HTML5*, *HTML 4.01*, or *XHTML*. Table 1.3 shows various link declarations.

Table 1.3: Link declarations	
Type	Description
HTML 5	<code><link rel="stylesheet" href="css/global.css"></code>
HTML 4.01	<code><link rel="stylesheet" type="text/css" href="css/global.css"></code> HTML 4.01 is similar to the HTML5 but requires an additional type attribute.
XHTML	<code><link rel="stylesheet" type="text/css" href="css/styles.css" /></code> XHTML is like HTML 4.01 but it requires a closing slash at the end of the <code><link></code> tag.



href attribute

The *href* attribute points to the *location* where *external CSS file* is *stored*. It works similar to the *src* attribute of the ** tag.

You can also link a style sheet using CSS. CSS allows you to link to the external style sheets using the *@import* directive.

```
@import url(css/global.css);
```

You can also call the *global.css* from the web page by adding this directive to an HTML style tag:

```
<style>
@import url(css/global.css);
</style>
```



HTML 4.01

If you are using *HTML 4.01* or *XHTML* doctype, you need to add *type="text/css"* to the opening *<style>* tag.



Multiple External Style Sheet

You can also include *multiple* external style sheets by using more than one `@import` directive.

You can use the regular CSS after the `@import` directive. You must place all the import directives *before CSS rules*. For example:

```
<style>
@import url(css/global.css);
@import url(css/formelements.css);
p { color: blue;}
</style>
```

You can also specify *character encoding* for a style sheet. The only declaration that can come before an `@import` statement is a `@charset` directive. This directive specifies the encoding used by the stylesheet.

```
@charset "UTF-8";
@import "global.css"
p { color: blue;}
```

Using Internal CSS

An *internal* [or *embedded*] style sheet can be used when a single document that has a unique style. You can define the styles in the *head* section of the page using the `<style>` element. Listing 1.1 shows the use of the `<style>` element.

Listing 1.1: c01embeddedcss.html

```
<!DOCTYPE html>
<html>
<head>
<title> Embedded CSS </title>

<style type="text/css">
p {
color: blue;
```

```
    }
```

```
h1 {  
color:red;
```

```
}
```

```
</style>  
</head>  
<body>  
<h1> Embedded styles in Action </h1> <p>  
You can use the style element to define an embedded style sheet !  
</p>  
</body>  
</html>
```

Inline Styles

The *inline* styles allow you to add style to *a specific element*. However, when you use the *inline* styles, you lose lots of advantages of a style sheet. Following is an example of the inline style:

```
<h1 style="color:red; margin-top: 2%;"> This is a heading ! </h1>
```

How CSS rules Cascade

If there are two or more rules that apply to a single element, it is important to know that which rule will take *precedence*. Generally, all styles will *cascade* into a *new virtual style sheet*. Following is the *order of precedence* from lower to higher priority [the *inline* style has the highest priority]:

1. Browser default (the default styles applied by the browsers)
2. User styles (styles that have been defined by the user)
3. External style sheet (styles imported using the link element)
4. Internal style sheet in the head element (styles created using the style element)
5. Inline style (styles that are defined using the style global attribute)

If there are two or more rules in a style sheet, you should remember

the following:

- Last Rule: If the two selectors are *identical*, the *last selector* will take *precedence*.
- Specificity: If one selector is more *specific* than the other, the *more specific* rule takes *precedence*.
- Important: You can also override the normal cascade order. You can add *!important* after any property value to indicate high priority.

Listing 1.2 shows demonstrate the specificity. Figure 2 shows the result of the page in the browser.

Listing 1.2: c01cascade.html

```
<!DOCTYPE html>
<html>
<head>
<title>Cascade in Action</title>

<style>

  *{

font-family: verdana, sans-serif;
font-size: 120%;

}

p{
font-family: arial, sans-serif;

}

}
```

```
strong{  
font-family: cursive;  
color: green;  
  
}  
}
```

```
p > strong{  
color:red !important;  
  
}  
}
```

```
p > strong{  
color: blue;  
  
}  
}
```

```
.cacd{  
background-color: #e7e7e7;  
width: 480px;  
text-align: center;  
border: 1px dashed #000;  
  
}  
}
```

```
i{  
text-decoration: underline;  
  
}  
}
```

```
i{  
text-decoration: overline;  
</style>
```

```

</head>
<body>
<p class="cacd">
This is the <strong> cascade </strong> in <i>action </i> !
</p>
</body>
</html>

```

This is the **cascade** in **action** !

Figure 2

In Listing 1.2:

- * is more specific than *p*.
- *strong* is more specific than *p > strong*.
- The second *i* selector takes precedence over the first *i*.
- In *p > strong* code block, *color:red !important;*, has more precedence than the *color: blue; declaration*.

If you enter in a situation where there are two styles that can be applied to a single element, browser determines the specificity of a style by using the following:

1. Number of *id values* in the *style selector*.
2. Number of *other attributes* and *pseudo-classes* in the *selector*
3. Number of *elements names* and *pseudo-classes* in the selector

To measure the *specificity*, count the number of *ID attributes* in the selector [say *a*], count the number of *other attributes* and *pseudo-classes* in the selector [say *b*], and then count the *number of elements names* and *pseudo-elements* in the selector [say *c*]. Now, concatenate the three numbers [*a-b-c*] to get the specificity. Listing 1.3 shows an example.

Listing 1.3: c01specificity.html

```
<!DOCTYPE html>
<html>
<head>
<title> Specificity </title>

<style>
h1 {
color:red;

}

p {
color:blue;

}

p.theClass{
color:chocolate;

}

</style>

</head>
<body>
<h1> Specificity in Action </h1>
<p class="theClass">
Browser uses specificity when it counters a tie-break situations!
</p>
</body>
</html>
```

In the Listing 1.3, the selector `p.theClass` has `0` id values, `1` other attributes, and `0` elements name, therefore, it has specificity of `0 1 0`. The other style has `0` id values, `0` other attributes, and `0` element names, therefore, the specificity of `0 0 0`.



Calculating specificity

You can also use the following website to calculate the specificity: <http://specificity.keegan.st>.

One other approach that is popular among the CSS programmers is to give a value of `100` to the `ID selector`, a value of `10` to `every class selector`, and a value of `1` to `every HTML selector`. Next, add them all to get the specificity value.

For example:

- `a`: has a specificity of `1` because it has `1 HTML selector`.
- `#theID`: has a specificity of `100`.
- `.theClass`: has a specificity of `10`.
- `div p.theClass`: has a `2` HTML selectors, and a class selector therefore a specificity `[1+1+10]` of `12`.

Considering the above rule and Listing 1.3, the `p` style has a specificity of `1` whereas the `p.theClass` style has a specificity of $1+10=11$.

Inheritance

`Inheritance` is a process by which some CSS properties are inherited from the parent tags. In other words, the properties applies to one tag is passed on to the nested tags. For example, if you specify `font-family` or `color` properties for the `<body>` element, they will be inherited by the most of the child elements such as `<h1>`, ``, ``, and so

on. Not all properties are inherited by the *child elements*. For example, *background-color* and *border* properties.

In the listing 1.4, the *inheritance* class takes inherited *padding* value of *10px* from the *body* style block.

Listing 1.4: c01inheritance.html

```
<!DOCTYPE html>
<html>
<head>
<title>Inheritance in Action</title>
<style>
body{
font-family: verdana, arial, sans-serif;
color: darkblue;
padding: 10px;

}

.inheritance{
width:500px;
border: 1px dotted #000;

}

</style>

</head>
<body>
<div class="inheritance">
<h1> Inheritance in Action </h1>
<p>
In object-oriented programming, inheritance is the concept that when a class of objects is defined, any subclass that is defined can inherit the definitions of
```

one or more general classes.

```
</p>
</div>
</body>
</html>
```

If you render the above Listing in browser, you will get the output as shown in Figure 3.

Inheritance in Action

In object-oriented programming, inheritance is the concept that when a class of objects is defined, any subclass that is defined can inherit the definitions of one or more general classes.

Figure 3

Now, add the following code in the *.inheritance* code block and re-render *[inheritance1.html]*: *padding: inherit;*.

```
.inheritance{
width:500px;
border: 1px dotted #000;
padding: inherit;
}
```

Notice in Figure 4, the *inheritance* class has *inherited padding value* of *10px* from the *body* code block.

Inheritance in Action

In object-oriented programming, inheritance is the concept that when a class of objects is defined, any subclass that is defined can inherit the definitions of one or more general classes.

Figure 4

Exercises

Exercise 1

Change the color of all `<p>` elements in Listing Ex1.1 to *red* using an *internal style sheet*.

Listing Ex1.1: c01ex1.html

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>
</head>
<body>

<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American
company that sells consumer electronics, computer software, and online
services. </p>
</body></html>
```

Exercise 2

Add an *external style sheet* to the Listing Ex1.1 with link to *global.css*.

Exercise 3

Using the Listing Ex1.1, set the color of the *second paragraph* to *blue*.

Exercise 4

Remove all styles from the Listing Ex1.2 except the *inline style*.

Listing Ex1.2: c01ex4.html

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>
<style>
body{
font-family: helvetica, verdana, sans-serif;
padding: 10px;

}

h1{
color: red;

}

#appleMania{
width: 300px;
border: 1px dotted #000;
color: blue;
padding: inherit;

}

</style>
<link rel="stylesheet" href="global.css"> </head>
<body>

<h1> About Apple </h1>
<div id="appleMania">
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p style="text-
```

```
decoration:underline">Apple Inc. is an american company that sells consumer  
electronics, computer software, and online services. </p> </div>  
</body>  
</html>
```

Summary

In this chapter, I have described how to *define* styles, how to connect to the *external* CSS style files, how CSS rules *cascade*, how *inheritance* work. You have also seen two different approaches you can take to calculate the specificity. In next chapter, we will talk about the selectors.

2

Identifying What to Select Selectors

In the previous chapter, I have explained how to *define* CSS rules. If you recall from the previous chapter, I talked about, how to style HTML elements such as *a*, *h1*, *body*, and so on. In CSS, these are referred to as *type selectors*. In this chapter, I will explain *core CSS3 selectors*. By the time you finish this chapter, you will have a firm understanding of making selections using various selectors.

Key Points & Concepts

- Using basic selectors
- Using pseudo-element selectors
- Using pseudo-class selectors

Using Basic CSS Selectors

A *selector* is the *most basic* component in CSS. In this section, I will explain some simple selectors.

Universal Selector (Asterisk)

The *universal selector* (*) is most fundamental selector of the CSS selectors. The *shorthand* for this selector is *asterisk*. It allows you to target every element in the document. However, it is used rarely because its scope is wide. For example, say you want all elements [*tags*] in the page to appear in *blue* color, your group selector would look like the following:

```
a, p, h1, h2, h3, h4, p, span, div ... ... and so on... {  
color: blue;  
  
}
```

The *universal selector* is a much shorter way to tell browser that you want all tags in the page to appear in *blue* color:

```
*  
  
{  
  
color: blue;  
  
}
```

The syntax for the *universal selector* is given next:

```
* {
```

css declarations;

}

Listing 2.1 shows an example that uses the *universal* selector.

Listing 2.1: c03universal-selector.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Universal Selector</title>
```

```
<style>
```

```
*{
```

```
border: 1px solid darkblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>
```

```
<h2> Universal Selector in Action </h2>
```

```
<p>
```

This is a paragraph. a distinct section of a piece of writing,
usually dealing with a single theme and indicated by a new line,
indentation, or numbering.

```
</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

The style I have defined in Listing 2.1 draws a *solid border* around the selected elements. You can see the effect of style in Figure 1.

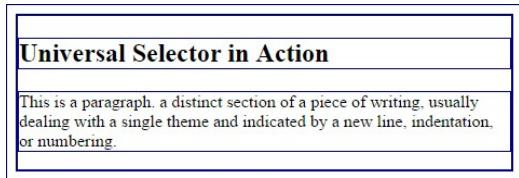


Figure 1

If you omit *asterisk (*)* with simple selectors, it does not affect the result. For example, **.warning* and *.warning* are same, **[lang=it]* and *[lang=it]* are same, and **#myid* and *#myid* are same. The *universal* selector can also select *everything inside an element*. In the following example, it selects all elements inside the *<div>* elements and set their *background color* to *blue*:

```
div * {  
    background-color: blue;  
  
}
```

Since the *universal* selector doesn't specify a particular element, it is rarely used by the CSS programmers.

Type Selector

If you want to select a *specific element* in a web page, you can use the *element* type as a selector. It matches elements by node name. For example, if you want to target all of *p* elements, use *p* as the *selector*. Listing 2-2 shows the use of the *type* selector, *p*:

Listing 2.2: c02type-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Type Selector </title>
<style>
p{
padding: 10px;
margin: 10px;
color: blue;

}

</style>
</head>
<body>
<h2> About Apple </h2>
<p>Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p>Apple Inc. is an American company that sells consumer electronics, computer software, and online services. </p>
</body>
</html>
```

Figure 2 shows the effect of the *p* selector used in Listing 2.2.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc. is an American company that sells consumer electronics, computer software, and online services.

Figure 2

Class Selector

The `class` selector matches an element based on the contents of the `class attribute` of the element. When you want to style elements different from the related elements on a page, for example, say if you want to add `padding` to a paragraph and while leaving other untouched, you can use the `class` selector with that paragraph. The `class` attribute can contain `space separated list` of items, for example, `<p class="class1 class2 class3"> </p>`.

You can create a `class` selector by first giving it a `name` and then applying it to the elements that you want to style. The syntax for the `class` selector is as follows:

```
.class {  
    css declarations;  
}
```

Listing 2.3 shows the demonstration of this selector.

Listing 2.3: c02class-selector.html

```
<!DOCTYPE html>  
<html>  
<head>  
<title> Class Selector </title>  
<style>  
.fruit{  
    color: blue;  
  
}  
  
.company{  
    color: red;
```

```
    }
```

```
.company1{  
font-family: Georgia, sans-serif;
```

```
}
```

```
</style>  
</head>  
<body>  
<h2> About Apple </h2>  
<p class="fruit">Apple is the round fruit of a tree of the rose family, which  
typically has thin green or red skin and crisp flesh.</p> <p class="company  
company1">Apple Inc. is an American company that sells consumer  
electronics, computer software, and online services. </p> </body>  
</html>
```

Figure 3 shows the effect of the *class* selector used in Listing 2.3. In this listing, I have used three classes: *fruit*, *company*, and *company1*. The *fruit* class selector targets the *first paragraph* whereas the other two selectors target the *second paragraph*.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.

Apple Inc. is an American company that sells consumer
electronics, computer software, and online services.

Figure 3

You can also pin point a tag using a *class* selector. You can target a *single type of element* that has been assigned to a class. In Listing 2.4, the *span* element is targeted which has been assigned the *company1* class selector. Figure 4 shows the result of the style.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics, computer software, and online services.

Figure 4

Listing 2.4: c02class-selector-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Class Selector </title>
<style>
.fruit{
color: blue;

}

.company{
color: red;

}

span.company1{
font-family: Georgia, sans-serif;
font-weight: bolder;

}

</style>
</head>
<body>
<h2> About Apple </h2>
<p class="fruit">Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p> <p class="company">
<span class="company1"> Apple Inc </span> is an American company that
```

sells consumer electronics, computer software, and online services. </p></body></html>

ID Selector

This selector allows you to select elements by using the *id* attribute. The value of the *id* attribute should be *unique* in a web page. It uses the following syntax:

```
#id {  
    css declarations;  
}
```

Listing 2.5 shows the *id* selector in action.

Listing 2.5: c02id-selector.html

```
<!DOCTYPE html>  
<html>  
<head>  
<title> ID Selector </title>  
<style>  
#company{  
    font-weight: 600;  
    text-decoration: underline;  
    color:red;  
}  
  
</style>  
</head>  
<body>  
  
<h2> About Apple </h2>  
<p id="fruit">Apple is the round fruit of a tree of the rose family, which  
typically has thin green or red skin and crisp flesh.</p> <p> <span  
id="company">Apple Inc</span> is an American company that sells  
consumer electronics, computer software, and online services. </p>
```

```
</body>  
</html>
```

In this example, I have targeted an element with *id* set to *company*. Figure 5 shows the result.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services.

Figure 5

Attribute Selectors

The **attribute selectors** allows you to target the elements based on a given attribute or attribute value. The following is the syntax for the attribute selector:

```
[attribute] { css declarations; }
```

Listing 2.6 shows the use of the **attribute** selector.

Listing 2.6:c02attribute-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[href] {
font-weight: 600;
color:blue;
text-decoration: none;

}

</style>
</head>
<body>
<h2> About Apple </h2>
<p><a href="www.apple.com"> Apple Inc. </a> is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p > Apple Inc is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs"> Steve Jobs </a> was one of the co-founder of the company. </p></body> </html>
```

In Listing 2.6, I have used the simplest form of the **attribute** selector that selects all tags with **href** attribute irrespective of the value the attribute holds. The style code in Listing 2.6 will select all tags with **href** attributes [the Apple and Steve Jobs links], as shown in Figure 6.

About Apple

[Apple Inc.](#) is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Figure 6

You can also use *conditions* to *match* attributes. The Table 2.1 shows various *conditions* you can use to *match attributes*.

Table 2.1: Conditions for the <i>attribute</i> selector
Selector (Condition)

Example

Example Description		
[attr]	[target]	Select all elements with a target attribute irrespective of the value assigned to the target attribute.
[attr="val"]	[target=_blank]	Select all elements with target set to _blank [see Listing 2.7 and Figure 7]
[attr^="val"]	a[href^="https"]	Select every <a> tag whose href attribute begins with the string https [see Listing 2.8 and Figure 8]
[attr\$="val"]	p[class\$="-2"]	Select every <p> tag whose class attribute ends with the string -2 [see Listing 2.9 and Figure 9]
[attr*="val"]	p[class*="pple"]	Select every <a> tag whose href attribute contains the string pple [see Listing 2.10 and Figure 10]
[attr~="val"]	a[title~="Apple"]	Select all tags with the title attribute containing space separated multiple values and one of them is Apple [see Listing 2.11 and Figure 11]
[attr = "val"]	p[class = "apple"]	Select all elements with a class attribute whose value is hyphen-separated list of values and the first of which is apple [see Listing 2.12 and Figure 12]

Listing 2.7: c02attribute-selector-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[href] {
font-weight: 600;
color:blue;
text-decoration: none;
```

}

```
[target=_blank]{
color:red;
```

}

```

</style>
</head>
<body>

<h2> About Apple </h2>
<p> Apple Inc is the round fruit of a tree of the rose family, which typically
has thin green or red skin and crisp flesh.</p> <p> <a
href="www.apple.com">Apple Inc</a> is an American company that sells
consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" target="_blank"> Steve Jobs
</a> was one of the co-founder of the company. </p>
</body>
</html>

```

About Apple

Apple Inc is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics, computer software, and online services. **Steve Jobs** was one of the co-founder of the company.

Figure 7

Listing 2.8: c02attribute-selector-2.html

```

<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[href]

```

{

text-decoration: none;

}

```

a[href^="https"]{
color: red;

```

```

        }

```

```

</style>
</head>
<body>

<h2> About Apple </h2>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p> <a
href="www.apple.com">Apple Inc</a> is an American company that sells
consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs"> Steve Jobs </a> was one of
the co-founder of the company. Use <a href="https://www.google.com">
Google </a> to know more about Steve Jobs. </p>
</body>
</html>

```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics,
computer software, and online services. Steve Jobs was one of the
co-founder of the company. Use [Google](#) to know more about Steve Jobs.

Figure 8

Listing 2.9: c02attribute-selector-3.html

```

<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[href]

```

```

{
```

text-decoration: none;

```

}
```

```

p[class$="-2"]{
border: 1px dotted darkgreen;
padding: 2px;

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p class="apple-1">Apple is the round fruit of a tree of the rose family, which
typically has thin green or red skin and crisp flesh.</p> <p class="apple-2" >
<a href="www.apple.com">Apple Inc</a> is an American company that sells
consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs"> Steve Jobs </a> was one of
the co-founder of the company. </p>
<p class="apple-2"> Use <a href="https://www.google.com"> Google </a> to
know more about Steve Jobs. </p>
</body>
</html>

```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics,
computer software, and online services. Steve Jobs was one of the co-
founder of the company.

Use [Google](#) to know more about Steve Jobs.

Figure 9

Listing 2.10: c02attribute-selector-4.html

```

<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[(href]

```

```
{
```

```
text-decoration: none;
```

```
}
```

```
p[class*="pple"]{  
border: 1px dotted green;  
padding: 2px;
```

```
}
```

```
</style>  
</head>  
<body>
```

```
<h2> About Apple </h2>
```

```
<p class="aple">Apple is the round fruit of a tree of the rose family, which  
typically has thin green or red skin and crisp flesh.</p>
```

```
<p class="apple-2" > <a href="www.apple.com">Apple Inc</a> is an  
American company that sells consumer electronics, computer software, and  
online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs"> Steve  
Jobs </a> was one of the co-founder of the company. </p>
```

```
<p class="apple-2"> Use <a href="https://www.google.com"> Google </a> to  
know more about Steve Jobs. </p>
```

```
</body>  
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. Steve Jobs was one of the co-founder of the company.

Use [Google](#) to know more about Steve Jobs.

Figure 10

Listing 2.11: c02attribute-selector-5.html

```
<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[href]
{
text-decoration: none;
}

p[class~="apple2"]{
border: 1px dotted red;
padding: 2px;
}

</style>
</head>
<body>
```

```

<h2> About Apple </h2>
<p class="apple1 apple2"> Apple is the round fruit of a tree of the rose family,
which typically has thin green or red skin and crisp flesh.</p>
<p class="apple1" ><a href="www.apple.com">Apple Inc</a> is an American
company that sells consumer electronics, computer software, and online
services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs"> Steve Jobs </a>
was one of the co-founder of the company. </p>
<p class="apple1 apple2"> Use <a href="https://www.google.com"> Google
</a> to know more about Steve Jobs. </p>
</body>
</html>

```

About Apple

Apple is the round fruit of a tree of the rose family, which typically
has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics,
computer software, and online services. Steve Jobs was one of the co-
founder of the company.

Use [Google](https://www.google.com) to know more about Steve Jobs.

Figure 11

Listing 2.12: c02attribute-selector-6.html

```

<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>
[href]

```

{

text-decoration: none;

}

```

p[class |="apple"]{
border: 1px dotted #000;

```

```
padding: 2px;
```

```
}
```

```
</style>
</head>
<body>
```

```
<h2> About Apple </h2>
<p class="apple-a-fruit">Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p class="apple-a-company"> <a href="www.apple.com" title="Apple the fruit">Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" title="Apple the company"> Steve Jobs </a> was one of the co-founder of the company. </p>
<p> Use <a href="https://www.google.com"> Google </a> to know more about Steve Jobs. </p>
</body>
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics, computer software, and online services. Steve Jobs was one of the co-founder of the company.

Use [Google](https://www.google.com) to know more about Steve Jobs.

Figure 12

Combining Selectors

You can *combine* different selectors to create CSS rules. In this section, I will describe various combination of the selectors.

Styling group of tags

To style a group of tags, define a *comma separated list* of selectors. In Listing 2.13, I have specified a type *selector(h1)*, and two attribute selectors (*p[class |= "apple"]* and *[href]*). You can see the effect of this union of selectors in Figure 13.

Listing 2.13: c02combining-selectors-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Combining Selectors </title>
<style>

h1, p[class |="apple"], [href] {
font-family: helvetica, verdana, sans-serif;
text-decoration: none;

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p class="apple-a-fruit">Apple is the round fruit of a tree of the rose family,
which typically has thin green or red skin and crisp flesh.</p>
<p class="apple-a-company"> <a href="www.apple.com" title="Apple the
fruit">Apple Inc</a> is an American company that sells consumer electronics,
computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" title="Apple the company">
Steve Jobs </a> was one of the co-founder of the company. </p>
<p> Use <a href="https://www.google.com"> Google </a> to know more
about Steve Jobs. </p>
</body>
</html>
```


Descendant Selector

You can use the *decedent* selector to match all elements that are *decedent* (*child*, *grandchild*, and *beyond*) of the specified element. To define a selector, add a *whitespace* between two simple selectors. In Listing 2.14, I have used a CSS rule that defines some CSS properties for all instances of *span* elements within the *p* element. Figure 14 shows the result of Listing 2.14.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics, computer software, and online services. Steve Jobs was one of the co-founder of the company.

Use [Google](#) to know more about Steve Jobs.

Figure 13

Listing 2.14: c02combining-selectors-2.html

```
<!DOCTYPE html>
<html>
<head>
<title> Combining Selectors </title>
<style>
body{
line-height: 1.5;
```

}

```
p span{
border: 1px dotted black;
padding: 3px;
margin: 2px;
```

}

```
</style>
```

```

</head>
<body>

<h2> About Apple </h2>
<p >Apple is the round <span class="apple-1"> fruit</span> of a tree of the
rose family, which typically has thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American <span
class="apple-2"> company</span> that sells consumer electronics, computer
software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of
the co-founder of the company. </p>
<p> Use <a href="https://www.google.com"> Google</a> to know more
about Steve Jobs. </p>
</body>
</html>

```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Figure 14

Listing 2.15 is another example of *descendant* selector. In this listing, I have defined two tables with id attributes. Then, I created two rules targeting all decedents of *th* elements within the tables. You can see the effect in Figure 15.

[January Sales Data]

Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	58	35	4500

[February Sales Data]

Salesman	Appointments	Sales	Expenses
Phillip	57	60	3200
Brent	98	52	2700
Bill	77	80	4200
Fred	105	62	4800
Tim	88	55	6200
Graham	46	41	4800

Figure 15

Listing 2.15: c02combining-selectors-3.html

```
<!DOCTYPE html>
<html>
<head>
<title> Combining Selectors </title>
<style>
body, h1{
font-family: verdana,san-sarif;
font-size: 90%;
```

}

```
h1{
width: 168px;
padding: 10px;
background-color: lightgreen;
```

}

```
#sales-table-1 th {
border-bottom: 3px solid red;
```

```

padding: 5px;

}

#sales-table-2 th{
border-bottom: 3px solid blue;
padding: 5px;

}

td{
color: darkbrown;
border-bottom: 1px solid black;
padding: 5px;

}

</style>
</head>
<body>

<h1> [ January Sales Data ] </h1>
<table id="sales-table-1">
<tr><th>Salesman</th> <th>Appointments</th> <th>Sales</th>
<th>Expanses</th></tr>
<tr><td>Phillip</td><td>110</td><td>57</td>
<td>2500</td></tr>
<tr><td>Brent</td><td>78</td><td>48</td>
<td>2000</td></tr>
<tr><td>Bill</td><td>100</td><td>75</td>
<td>3700</td></tr>
<tr><td>Fred</td><td>95</td><td>56</td>
<td>4000</td></tr>
<tr><td>Tim</td><td>99</td><td>45</td>
<td>6000</td></tr>
<tr><td>Graham</td><td>58</td><td>35</td>
<td>4500</td></tr> </table>

<h1> [ February Sales Data ] </h1>
<table id="sales-table-2">

```

Salesman	Appointments	Sales
3200	Phillip	57
2700	Brent	98
4200	Bill	77
4800	Fred	105
6200	Tim	88
4800	Graham	46

</body>
</html>

Child Selector

The **child** selector allows you to select the *direct children* of a *parent ancestor*. To define a child selector, place a **>** sign (greater than sign) between two simple selectors. In Listing 2.16, I am looking for the **li** element that is a direct child of **ul** element. You can see the effect in Figure 16.

Listing 2.16: c02combining-selectors-4.html

```
<!DOCTYPE html>
<html>
<head>
<title> Combining Selectors </title>
<style>
body, h1{
font-family: verdana,san-sarif;
font-size: 90%;
margin: 20px;

}

ul > li{
color: blue;

}

</style>
</head>
<body>
<ul>
<li> Item 1</li>
<li> Item 2 </li>
<ol>
```

```
<li> Nested Item 1</li>
<li> Nested Item 2</li>
</ol>
<li> Item 3 </li>
</ul>

</body>
</html>
```

If you use the following descendant selector with the Listing 2.16, all `` tags will be targeted:

```
ul li{
  color: blue;
```

```
}
```

- Item 1
- Item 2
 - 1. Nested Item 1
 - 2. Nested Item 2
- Item 3

Figure 16

Selecting Adjacent Sibling Elements

This selector only selects the specified element that is immediately preceded by the former element. This selector is almost similar to the general sibling selector with the only difference that the targeted element must be an *immediate sibling*, not just a general sibling. To define adjacent sibling selector, put a *plus* sign between two simple selectors. For example, `li + li { border-left: 1px dotted #000;}` rule select a *list item* that is directly following another *li* element.

In Listing 2.17, I am looking for all *p* elements that are immediately after the *div* element. Figure 17 shows the effect of the rule.

Listing 2.17: c02combining-selectors-5.html

```
<!DOCTYPE html>
<html>
<head>
<title> Combining Selectors </title>
<style>
body{
line-height: 1.5;

}

div + p{
color: blue;

}

</style>
</head>
<body>
<div>
<h1> About Apple </h1>
```

```
<p>Apple is the round <span class="apple-1"> fruit</span> of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p><a href="www.apple.com" >Apple Inc</a> is an American <span class="apple-2"> company</span> that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" >Steve Jobs </a> was one of the co-founder of the company. </p>
<div class="search"> Search the Web </div>
```

```
<p> Use <a href="https://www.google.com">Google</a> to know more about Steve Jobs. </p>
</div>
```

```
</body>
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about Steve Jobs.

Figure 17

Figure 18 shows the result if you use the following rule:

```
p + div{
color: blue;

}
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

[Search the Web](#)

Use [Google](#) to know more about Steve Jobs.

Figure 18

Figure 19 shows the result if you use the following rule:

```
p + p{  
color: blue;  
}  
}
```

Selecting General Sibling Selector

This selector matches elements that are *sibling* of the *specified element*. To create a general sibling selector, place a *tilde character (~)* between the simple selectors. In Listing 2.18, I am looking to match all *p* elements if it is a sibling of a *p* element. Figure 20 shows the render of Listing 2.18.

Listing 2.18:c02combining-selectors-6.html

```
<!DOCTYPE html>
<html>
<head>
<title> Combining Selectors </title>
<style>
body{
line-height: 1.5;

}

p~p{
color: blue;

}

</style>
</head>
<body>
<div>
<h1> About Apple </h1>
<p >Apple is the round <span class="apple-1"> fruit</span> of a tree of the
rose family, which typically has thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American <span
class="apple-2"> company</span> that sells consumer electronics, computer
software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" >Steve Jobs</a> was one of
```

the co-founder of the company. </p>
<div class="search"> Search the Web </div>

<p> Use Google to know more about Steve Jobs. </p>
</div>

</body>
</html>

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about Steve Jobs.

Figure 19

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about Steve Jobs.

Figure 20

Figure 21 shows the result, if you use the following rule:

```
div~p{  
color: blue;  
}  
}
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about Steve Jobs.

Figure 21

Using Pseudo-Element Selectors

The *pseudo elements* allow you to style certain parts of a document. For example, the `::first-line` pseudo-element is used to target the *first line* of an element.

Following is the syntax for pseudo elements:

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



: Vs ::

Sometimes `:` is used in place of `::`. The *double-colon* `::` convention is an attempt to distinguish between *pseudo-classes* and *pseudo-elements*. However, most of the browser supports both conventions.



Selection pseudo-element

The *Selection* pseudo-element always starts with a double colons (`::`).

In this section, I will discuss various pseudo-elements.

Using the `::first-line` Selector

The `::first-line` selector applies styles only to the first line of an element, it does not target any real HTML element. The volume of

text depends on many factors such as width of the element is one of them. In Listing 2.19, I am selecting first line of each paragraph. Figure 22 shows the result.

Listing 2.19:c02first-line-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> First Line Selector </title>
<style>
p::first-line{
background-color: grey;
color: black;

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p>Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American company that sells consumer electronics, computer software, and online services. </p>
</body>
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.

Apple Inc. is an American company that sells consumer electronics, computer software, and online services.

Figure 22



::first-line Selector

You can use the `::first-line` selector only with the *block-level* elements.

Using the ::first-letter Selector

The `::first-letter` selector allows you to select *first letter* of the first line of block. In Listing 2.20, the selector applies *red* color to the *first letter* of the *paragraph* assigned with `id1` selector. Figure 23 shows the result of Listing 2.20.

Listing 2.20:c02first-letter-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> First Letter Selector </title>
<style>
p#id1::first-letter{
color: red;
font-size: 200%;

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p id="id1">Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American company that sells consumer electronics, computer software, and online services. </p>
</body>
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.

Apple Inc. is an American company that sells consumer electronics, computer software, and online services.

Figure 23

Using the :before and :after Selectors

These two selectors are used to add *cosmetic content* to the page using the *content* property. These elements are *inline* by default. In Listing 2.21, I have placed two images before and after the *h2* element. Figure 24 shows the result.

Listing 2.21: c02before-after-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Class Selector </title>
<style>
h2::before{
content: url("images/aapple1.png");

}

h2::after{
content: url("images/apple2.png");

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p id="id1">Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American company that sells consumer electronics, computer software, and online services. </p>
</body>
</html>
```



About Apple

Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.

Apple Inc. is an American company that sells consumer electronics, computer software, and online services.

Figure 24

Using the ::Selection Selector

You can use this selector to a portion of the document that is selected by the user in the browser using mouse or any other pointing device. In Listing 2.22, this selector applies **chocolate color background** and **white colored text** for the selected text in the browser. Figure 25 shows the result.

Listing 2.22:c02selection-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> ::selection Selector </title>
<style>
::selection {
background: chocolate;
color: white;

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p>Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American company that sells consumer electronics, computer software, and online services. </p>
```

```
</body>  
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family. It typically has thin green or red skin and crisp flesh.

Apple Inc. is an American company that sells consumer electronics, computer software, and online services.

Figure 25

Using the Pseudo-Classes Selectors

A *pseudo-class* is a keyword that is used with selectors to target a special state of the element. For example, the *:hover* pseudo-class will apply style to an element when the user hovers over the element. Psuedo-classes together with pseudo-elements give you ability to not only select elements from the page hierarchy but also from the history of content. For example, the *:visited* pseudo class allows you to select links that have been visited by the user. In this section, I will explain different pseudo-classes.

Following is the syntax for declaring a pseudo-class:

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

Using the *:root* Selector

The *:root* pseudo-class selects the first element in the document tree. You are not going to use this selector very often. The root in an HTML document will be always the *html* element. Listing 2.23 shows the root selector in use. You can see the effect of this selector in Figure 26.

Listing 2.23:c02root-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title>Root Selector</title>
<style>
:root{
padding: 20px;
border: 1px solid blue;

}

</style>
</head>
<body>
<div class="inheritance">
<h2> Root Selector in Action </h2>
<p>
This is a paragraph. a distinct section of a piece of writing,
usually dealing with a single theme and indicated by a new line,
indentation, or numbering.
</p>
</div>
</body>
</html>
```

Root Selector in Action

This is a paragraph. a distinct section of a piece of writing,
usually dealing with a single theme and indicated by a new line,
indentation, or numbering.

Figure 26

Using the :firstchild Selector

This class selects any element that is the *first child* element of its *parent*. Listing 2.24 shows the *:firstchild* selector in use.

Listing 2.24: c02firstchild-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> First Child Selector </title>
<style>
body{
line-height: 1.5;

}

p > a:firstchild{
color:red;

}

</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
<p > Apple is the round <span class="apple-1"> fruit</span> of a tree of the
rose family, which typically has thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American <span
class="apple-2"> company</span> that sells consumer electronics, computer
software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of
the co-founder of the company. </p>
<div class="search"> Search the Web </div>

<p> Use <a href="https://www.google.com"> Google </a> to know more
about Steve Jobs. </p>
```

```
</div>

</body>
</html>
```

The `:firstchild` selector in the Listing 2.23 will match any `a` element that is *first child* of the `p` element. You can see the match in Figure 27.

Using the `:lastchild` Selector

This pseudo-class selects any element that is the *last child* of its *parent*. Listing 2.25 shows this selector in use.

Listing 2.25:c02lastchild-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Last Child Selector </title>
<style>
body{
line-height: 1.5;

}

p > a:lastchild{
color:red;

}

</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
<p > Apple is the round fruit of a tree of the rose family, which typically has
```

```

thin green or red skin and crisp flesh.</p>
<p> <a href="www.apple.com" >Apple</a> Inc is an American company that
sells consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" >Steve Jobs</a> was one of
the co-founder of the company. </p>
<div class="search"> Search the Web </div>

<p> Use Google to know more about Steve Jobs. </p>

</div>

</body>
</html>

```

You can see in Figure 28 the elements this pseudo class matches. Notice there are two *a* elements in the middle paragraph but the *:lastchild* selector affected the second *a* element in the paragraph because it is last child of its parent (*p*).

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about Steve Jobs.

Figure 27

Using the *:onlychild* Selector

This class represents any element which is the *only child* of its parent. Listing 2.26 shows the selector in use.

Listing 2.26: c02onlychild-selector.html

```
<!DOCTYPE html>
```

```
<html>
<head>
<title> Only Child Selector </title>
<style>
body{
line-height: 1.5;

}

p > a:onlychild{
color:red;

}

</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
<p > Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of the co-founder of the company. </p>
<div class="search"> Search the Web </div>

<p> Use <a href="https://www.google.com">Google</a> to know more about <span class="class1">Steve Jobs.</span></p>
</div>

</body>
</html>
```

Figure 29 shows the result of the Listing 2.26. Notice the selector matches *a* elements in the first two paragraphs. It did not affect the *a* tag in the third paragraph because it has two children.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use Google to know more about Steve Jobs.

Figure 28

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about Steve Jobs.

Figure 29

Using the :only-of-type Selector

This pseudo-class allows you to match any element that has *no siblings* of the *given type* (define by their parent). Listing 2.27 shows the use of this class.

Listing 2.27: c02onlyoftype-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Only of Type Selector </title>
<style>
body{
```

```
line-height: 1.5;
```

```
}
```

```
p > span:only-of-type{  
color:red;
```

```
}
```

```
</style>  
</head>  
<body>  
<div id="desc">  
<h1> About Apple </h1>  
<p >Apple is the round fruit of a tree of the rose family, which typically has  
thin green or red skin and crisp flesh.</p>  
<p ><a href="www.apple.com" >Apple Inc</a> is an American company that  
sells consumer electronics, computer software, and online services. <a  
href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of  
the co-founder of the company. </p>  
<div class="search"> Search the Web </div>
```

```
<p> Use <a href="https://www.google.com"> Google</a> to know more  
about <span class="class1">Steve Jobs.</span></p>  
</div>  
</body>  
</html>
```

Notice the result in Figure 30, this selector matches the *span* element in the third paragraph.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use [Google](#) to know more about [Steve Jobs](#).

Figure 30

Using :firstof-type Selector

This pseudo-class matches the *first sibling* of its *type* from the children of its parent. Listing 2.28 shows the use of this selector.

Listing 2.28: c02firstoftype-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> First of Type Selector </title>
<style>
body{
line-height: 1.5;
```

```
}
```

```
p > span:firstof-type{
color:red;
```

```
}
```

```
</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
```

```

<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p>
<p><a href="www.apple.com" >Apple Inc</a> is an American company that
sells consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of
the co-founder of the company. </p>
<div class="search"> Search the Web </div>

<p>Use <span class="class1">Google</span> search to know more about
<span class="class1">Steve Jobs.</span></p>
</div>

</body>
</html>

```

Figure 31 shows the effect of this selector. Notice the class matches the *first span* element in the last paragraph *ignoring* the second span element.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics,
computer software, and online services. [Steve Jobs](#) was one of the co-
founder of the company.

Search the Web

Use [Google](#) search to know more about Steve Jobs.

Figure 31

Using :last-of-type Selector

This pseudo-class matches the *last sibling* of its *type* from the *children* of its *parent*. Listing 2.29 shows the use of this selector.

Listing 2.29:c02lastoftype-selector.html

```

<!DOCTYPE html>
<html>
<head>

```

```

<title> Last of Type Selector </title>
<style>
body{
line-height: 1.5;

}

p > span:last-of-type{
color:red;

}

</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
<p >Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American company that
sells consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of
the co-founder of the company. </p>
<div class="search"> Search the Web </div>

```

```

<p>Use <span class="class1">Google</span> search to know more about
<span class="class1">Steve Jobs.</span></p>
</div>
</body>
</html>

```

Notice in Figure 32, the selector matches the last span element in the last paragraph.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use Google search to know more about [Steve Jobs](#).

Figure 32

Using :nth-of-type(n) Selector

The ***nth-of-type(n)*** selector targets every element that is the ***nth*** child of a specific element. ***n*** can be a ***number***, a ***keyword***, or ***formula***. Listing 2.30 shows the selector in use.

Listing 2.30: c02nthoftype-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> nth of Type Selector </title>
<style>
body{
line-height: 1.5;
```

```
}
```

```
div p:nth-of-type(2){
color:red;
```

```
}
```

```
</style>
</head>
<body>
<div id="desc">
```

```

<h1> About Apple </h1>
<p >Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American company that
sells consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" >Steve Jobs </a> was one of
the co-founder of the company. </p>
<div class="search"> Search the Web </div>

<p>Use <span class="class1">Google</span> search to know more about
<span class="class1">Steve Jobs.</span></p>
</div>

</body>
</html>

```

Notice in Figure 33, this selector targets the *middle* paragraph which is the *second child* of the *div* element.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Search the Web

Use Google search to know more about Steve Jobs.

Figure 33

Listing 2.31 shows another use of the selector. You can use *even* and *odd* keywords to select the even and odd children, respectively, of an element. Figure 34 shows the result. Notice that I have used the selector to change the color of the alternate rows.

Listing 2.31: c02nthoftype-selector1.html

```

<!DOCTYPE html>
<html>
<head>
<title> nth of Type Selector </title>
<style>
h1{
font-size: 20px;
padding: 10px;
background-color: lightgreen;
width: 200px;

}

#sales-table-1 tr:nth-of-type(odd) {
background-color: slategray;

}

#sales-table-1 tr:nth-of-type(even) {
background-color: lightblue;

}

</style>
</head>
<body>
<h1> [ January Sales Data ] </h1>
<table id="sales-table-1">
<tr><th>Salesman</th>      <th>Appointments</th>      <th>Sales</th>
<th>Expenses</th></tr>      <tr><td>Phillip</td><td>110</td><td>57</td>
<td>2500</td></tr>      <tr><td>Brent</td><td>78</td><td>48</td>
<td>2000</td></tr>      <tr><td>Bill</td><td>100</td><td>75</td>
<td>3700</td></tr>      <tr><td>Fred</td><td>95</td><td>56</td>
<td>4000</td></tr>      <tr><td>Tim</td><td>99</td><td>45</td>
<td>6000</td></tr>      <tr><td>Graham</td><td>58</td><td>35</td>

```

```

<td>4500</td></tr> </table>
</body>
</html>

```

[January Sales Data]			
Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	58	35	4500

Figure 34

You can also use a *formula*, $(an+b)$ with this selector. In this formula, *n* is the *counter* which starts at *0*, *a* is the *cycle size* and *b* is an *offset value*. In the Listing 2.32, I have use the formula, $3n+0$. It will select all *tr* elements whose index is multiple of *3*. This selector will target every *third row* of the table, refer to Figure 35.

Listing 2.32: c02nth-of-type-selector.html

```

<!DOCTYPE html>
<html>
<head>
<title> nth of Type Selector </title>
<style>
h1{
font-size: 20px;
padding: 10px;
background-color: lightgreen;
width: 200px;

}

#sales-table-1 tr:nth-of-type(3n+0){
background-color: lightgray;

```

```
}
```

```
</style>
</head>
<body>
<h1> [ January Sales Data ] </h1>
<table id="sales-table-1">
<tr><th>Salesman</th>          <th>Appointments</th>          <th>Sales</th>
<th>Expanses</th></tr>        <tr><td>Phillip</td><td>110</td><td>57</td>
<td>2500</td></tr>          <tr><td>Brent</td><td>78</td><td>48</td>
<td>2000</td></tr>          <tr><td>Bill</td><td>100</td><td>75</td>
<td>3700</td></tr>          <tr><td>Fred</td><td>95</td><td>56</td>
<td>4000</td></tr>          <tr><td>Tim</td><td>99</td><td>45</td>
<td>6000</td></tr>          <tr><td>Graham</td><td>58</td><td>35</td>
<td>4500</td></tr> </table>
</body>
</html>
```

[January Sales Data]

Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	58	35	4500

Figure 35

Using :nthchild(n) Selector

This class matches the elements that are *nth* child of their parent. In the Listing 2.32, the *span:nthchild(2)* style targets every *span* element that is *second child* of its parent. Figure 36 shows the result.

Listing 2.33: c02nthchild-selector.html

```
<!DOCTYPE html>
<html>
<head>
```

```
<title> nth of Type Selector </title>
<style>
body{
line-height: 1.5;

}

span:nthchild(2){
color: red;

}

</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p><a href="www.apple.com" >Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of the co-founder of the company. </p> <p>Use <span class="class1">Google</span> search to <span class="class2">know more </span> about <span class="class3">Steve Jobs.</span></p>
</div>
</body>
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Use Google search to [know more](#) about Steve Jobs.

Figure 36

Using :nthlast-child(n) Selector

In the Listing 2.33, the *span:nthlast-child(3)* style selects every *span* element that is *second child* of its *parent* counting from the *last child*. Figure 37 shows the result.

Listing 2.34: c02nthlastchild-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> nth of Type Selector </title>
<style>
body{
line-height: 1.5;
```

}

```
span:nthlast-child(3){
color: red;
```

}

```
</style>
</head>
<body>
<div id="desc">
```

```

<h1> About Apple </h1>
<p >Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American company that
sells consumer electronics, computer software, and online services. <a
href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of
the co-founder of the company. </p>

<p>Use <span class="class1">Google</span> search to <span
class="class2">know more </span> about <span class="class3">Steve
Jobs</span> and <span class="class3">Apple Inc.</span></p>
</div>

</body>
</html>

```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics,
computer software, and online services. [Steve Jobs](#) was one of the co-
founder of the company.

Use Google search to [know more](#) about Steve Jobs and Apple Inc.

Figure 37

Using :nthlast-of-type(n) Selector

This class selector matches every element that is *nth* child of *a particular type* counting from the last child. In Listing 2.35, this selector matches third *span* element [counting from last child of parent, *p*] of the last paragraph. Figure 38 shows the result.

Listing 2.35: c02nthoflasttype-selector.html

```

<!DOCTYPE html>
<html>
<head>
<title> nth of Type Selector </title>
<style>
body{

```

```

line-height: 1.5;

}

span:nthlast-of-type(3){
color: green;

}

</style>
</head>
<body>
<div id="desc">
<h1> About Apple </h1>
<p >Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p>
<p ><a href="www.apple.com" >Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" > Steve Jobs</a> was one of the co-founder of the company. </p>
<p>Use <span class="class1">Google</span> search to <span class="class2">know more </span> about <span class="class3">Steve Jobs</span> and <span class="class3">Apple Inc.</span></p>
</div>
</body>
</html>

```

Using the `:enabled`, `:disabled`, `:checked`, and `:default` Selectors The *enabled* and *disabled* classes represent an enabled and disabled elements, respectively. The checked CSS selector represents any *radio*, *check box*, or *option* element that is *checked* or is in the *on* state. The *default* selector targets an element that is default [such as the *Submit* button]. Listing 2.36 shows the use of these classes.

Listing 2.36: c02enableddisabled-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> :enabled, :disabled Selector </title>
<style>
body{
line-height: 1.5;
padding: 10px;

}

input:enabled{
background-color:#e5dded;

}

input:disabled{
background-color: lightblue;

}

input:checked+span{
color:#931505;

}

:default{
color: red;

}

</style>
```

```

</head>
<body>
<form action="example.com">
<p>
<label for="userid">User ID:</label>
<input type="text" id="userid"><br />
<label for="emailID">Email ID:</label>
<input type="text" id="emailid" disabled="disabled"><br/>
<label for="pw">Password:</label>
<input type="password" id="pw"><br/>

<label for="rememberme">Remember me:</label>
<input type="checkbox" id="rememberme" checked>
<span> Ensure cookies are enabled !</span> <br/>
</p>
<input type="submit" value="Submit" />
<a href="example.com"> Reset Password </a>
</form>
</body>
</html>

```

Figure 39 shows the result of the Listing 2.36. Notice the color of the ***Submit*** button is ***red*** because it is the ***default*** element in the ***form*** element. There isn't much styling you can use with check boxes therefore I have added a rule ***input:checked+span*** that targets the text attached to check box using the ***span*** element.

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Use Google search to [know more](#) about Steve Jobs and Apple Inc.

Figure 38

User ID:

Email ID:

Password:

Remember me: Ensure cookies are enabled !

[Submit](#) | [Reset Password](#)

Figure 39

Using the :valid, :invalid, and :required Selectors

The **:valid** class targets those input elements whose content validates correctly whereas the **:invalid** class targets the elements whose content fails to validate. The **:required** targets those **input** elements whose required attribute is set on it. Listing 2.37 shows the use of these classes. Figure 40 shows the result.

Listing 2.37: c02validinvalid-selectors.html

```
<!DOCTYPE html>
<html>
<head>
<title> :valid, :invalid, and :required Selectors </title>
<style>
body{
line-height: 1.5;
padding: 10px;
}

input:invalid{
background-color: #807b7b;

}

input:valid{
background-color: lightgreen;

}
```

```

input:required{
background-color: chocolate;

}

</style>
</head>
<body>
<form action="example.com">
<p>
<label for="userid">User ID:</label>
<input type="text" id="userid" required><br />
<label for="emailID">Email ID:</label>
<input type="email" id="emailid" ><br/>
<label for="pw">Password:</label>
<input type="password" id="pw"><br/>

<label for="rememberme">Remember me:</label>
<input type="checkbox" id="rememberme" checked>
<span> Ensure cookies are enabled !</span> <br/>
</p>
<input type="submit" value="Submit" />
<a href="example.com"> Reset Password </a>
</form>
</body>
</html>

```

User ID:

Email ID:

Password:

Remember me: Ensure cookies are enabled !

[Reset Password](#)

Figure 40

Using the :optional Selector

This class selects an *input element* that does not have the *required* option set on it. Listing 2.38 demonstrate the use of this class. Figure 41 shows the result.

Listing 2.38: c02optional-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> :valid, :invalid, and :required Selectors </title>
<style>
body{
line-height: 1.5;
padding: 10px;

}

input:optional{
background-color: yellow;

}

</style>
</head>
<body>
<form action="example.com">
<p>
<label for="userid">User ID:</label>
<input type="text" id="userid" required><br />
<label for="emailID">Email ID:</label>
<input type="email" id="emailid" ><br/>
<label for="pw">Password:</label>
<input type="password" id="pw"><br/>

<label for="rememberme">Remember me:</label>
<input type="checkbox" id="rememberme" checked>
<span> Ensure cookies are enabled !</span> <br/>
</p>
<input type="submit" value="Submit" />
```

```
<a href="example.com"> Reset Password </a>
</form>
</body>
</html>
```

User ID:
Email ID:
Password:
Remember me: Ensure cookies are enabled!
 [Reset Password](#)

Figure 41

Using :link and :visited Selectors

The **:link** class selects the links inside elements. It selects any link that has not visited yet. The **:visited** class selects those links that have already been visited. Listing 2.39 shows the demonstration of these classes. Figure 42 shows the result.

Listing 2.39: c02linkvisited-selectors.html

```
<!DOCTYPE html>
<html>
<head>
<title> Attribute Selector </title>
<style>

a:link {
border: 1px solid black;
background-color: lightgrey;
padding:4px;
margin:4px;

}

a:visited{
background-color: lightgreen;
```

```
}
```

```
</style>
</head>
<body>

<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p><a href="www.apple.com">Apple Inc</a> is an American company that sells
consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs">Steve Jobs</a> was one of
the co-founder of the company. </p>
</body>
</html>
```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Figure 42

Using the :hover Selector

This class selector matches elements when you *hover* your mouse over them. Listing 2.40 shows the demonstration of this class. Figure 43 shows the result.

Listing 2.40: c02hover-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Hover Selector </title>
<style>

a:link {
border: 1px solid black;
```

```

background-color: lightgrey;
padding:4px;
margin:4px;

}

a:hover{
background-color: lightgreen;

}

</style>
</head>
<body>

<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p><a href="www.apple.com">Apple Inc</a> is an American company that sells
consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs">Steve Jobs</a> was one of
the co-founder of the company. </p>
</body>
</html>

```

About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Figure 43



:hover Selector

You can use the `:hover` selector on all elements, not just links.

Using :active Selector

This class targets an element when it is being *activated* by the user. If you are using *mouse*, the active state is when you *press* the *mouse button* and then *release* it. Listing 2.41 and Figure 44 demonstrate the use of this selector.

Listing 2.41: c02active-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> :active Selector </title>
<style>
body{
line-height: 1.5;
padding: 10px;

}

a:active{
border: 3px dotted red;

}

</style>
</head>
<body>
<form action="example.com">
<p>
<label for="userid">User ID:</label>
<input type="text" id="userid" required><br />
<label for="emailID">Email ID:</label>
<input type="email" id="emailid" ><br/>
<label for="pw">Password:</label>
<input type="password" id="pw"><br/>

<label for="rememberme">Remember me:</label>
```

```

<input type="checkbox" id="rememberme" checked>
<span> Ensure cookies are enabled !</span> <br/>
</p>
<input type="submit" value="Submit" />
<a href="example.com"> Reset Password </a>
</form>
</body>
</html>

```

User ID:

Email ID:

Password:

Remember me: Ensure cookies are enabled !

Figure 44



Styling Links

In order to style links properly, you need to define the `:active` rule after all link related rules. You should style the links using the **LVHA** order: `:link`, `:visited`, `:hover`, and `:active`.

Using `:focus` Selector

This class is used to style elements when they *receive focus*. Listing 2.42 demonstrates the use of this selector. Figure 45 shows the result.

Listing 2.42: c02focus-selector.html

```

<!DOCTYPE html>
<html>
<head>
<title> :focus Selector </title>
<style>
body{
line-height: 1.5;
padding: 10px;
}

:focus{
border: 3px dotted red;

```

```
}
```

```
</style>
</head>
<body>
<form action="example.com">
<p>
<label for="userid">User ID:</label>
<input type="text" id="userid" required><br />
<label for="emailID">Email ID:</label>
<input type="email" id="emailid" ><br/>
<label for="pw">Password:</label>
<input type="password" id="pw"><br/>

<label for="rememberme">Remember me:</label>
<input type="checkbox" id="rememberme" checked>
<span> Ensure cookies are enabled !</span> <br/>
</p>
<input type="submit" value="Submit" />
<a href="example.com"> Reset Password </a>
</form>
</body>
</html>
```

User ID:

Email ID:

Password:

Remember me: Ensure cookies are enabled !

Figure 45

Using :empty Selector

This selector matches the elements that define *no children*. Listing 2.43 shows the use of this selector. Notice in Figure 46 the *empty td* element has the *red background color*.

Listing 2.43: c02empty-selector.html

```

<!DOCTYPE html>
<html>
<head>
<title> :empty Selector </title>
<style>
body{
line-height: 1.5;
padding: 10px;

}

td:empty{
background-color: red;

}

</style>
</head>
<body>
<table>
<tr><td>1</td><td>One</td></tr>
<tr><td>2</td><td>Two</td></tr>
<tr><td>3</td><td></td></tr>
</table>
</body>
</html>

1 One
2 Two
3

```

Figure 46

Using :lang() Selector

This class styles elements based on the *lang* attribute. Listing 2.44 shows the use of this selector. Figure 47 shows the result.

Listing 2.44: c02lang-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Lang Selector </title>
<style>
:lang(en){color:red}
:lang(de){color:green}
:lang(fr){color:blue}
</style>
</head>
<body>

<p lang="en"> Write text in English here !</p>
<p lang="de"> Write text in German here !</p>
<p lang="fr"> Write text in French here !</p>
```

```
</body>
</html>
```

Write text in English here !

Write text in German here !

Write text in French here !

Figure 47

Using :target Selector

This class styles a unique element with an *id* matching the fragment identifier of the *url* of the document. To see the use of this selector, run the Listing 2.45 using the following url: <http://localhost/icssc02target-selector.html#id1> or <http://localhost/icssc02target-selector.html#id2>. Figure 48 shows the result.

Listing 2.45: c02target-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Target Selector </title>
<style>
:target{
border: 1px solid #222222;
padding: 10px;
}

</style>
</head>
<body>

<h2> About Apple </h2>
<p id="id1">Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p> <p id="id2" ><a href="www.apple.com">Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" target="_blank">Steve Jobs</a> was one of the co-founder of the company. </p>
</body>
</html>
```



About Apple

Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

[Apple Inc](#) is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Figure 48

Using :not() Selector

The **:not** selector also known as ***negation*** selector. It used the

following syntax: `not(x)`. It matches an element that is not represented by the `x`. Listing 2.46 shows the use of this selector and Figure 49 shows the result.

Listing 2.46: c02negation-selector.html

```
<!DOCTYPE html>
<html>
<head>
<title> Negation Selector </title>
<style>
p:not(.class1){
border: 1px dotted #CCC;
color:red;

}

body :not(p){
color:blue;

}

</style>
</head>
<body>

<h2> About Apple </h2>
<p class="class1"><a href="www.apple.com"> Apple Inc</a> is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p> <p class="class2" > Apple Inc is an American company that sells consumer electronics, computer software, and online services. <a href="http://en.wikipedia.org/wiki/Steve_Jobs" target="_blank">Steve Jobs</a> was one of the co-founder of the company. </p>
</body>
</html>
```

About Apple

[Apple Inc](#) is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.

Apple Inc is an American company that sells consumer electronics, computer software, and online services. [Steve Jobs](#) was one of the co-founder of the company.

Figure 49



Negation Selector

X must not contain another *negation selector*.

Exercises

Exercise 1

Using the template shown in Listing Ex2.1, change the color of the *first letter* of the paragraph to *blue* using the paragraph defined by *id1*. Also, scale font size to *130%*.

Listing Ex2.1: c02ex1.html

```
<!DOCTYPE html>
<html>

<head>
<title> First Letter Selector </title>
<style>
</style>
</head>

<body>

<h2> About Apple </h2>
<p id="id1">Apple is the round fruit of a tree of the rose family. It typically
has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American
company that sells consumer electronics, computer software, and online
services. </p>
</body>

</html>
```

Exercise 2

Using the template shown in Listing Ex2.2, change the *color* of the *h1* element to *blue* that is *first child* of its parent.

Listing Ex2.2: c02ex2.html

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

```

<head>
<title> Child Selector </title>
</head>

<body>
<div>
<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American company that sells consumer electronics, computer software, and online services. </p> </div>
</body>

</html>

```

Exercise 3

Use template shown in Listing Ex2.3. Create a rule for the ***h1*** and ***p*** elements when they are ***active*** using the following declarations:

1. Set ***background color*** to ***light yellow***.
2. Set ***1px dotted*** border with ***black*** color.
3. Assign ***padding*** of ***10px***.

Listing Ex2.3: c02ex3.html

```

<!DOCTYPE html>
<html>

<head>
<title> Active Elements </title>
<style>

</style>
</head>

<body>

```

```
<div>
<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American
company that sells consumer electronics, computer software, and online
services. </p> </div>
</body>

</html>
```

Exercise 4

Use the template shown in Listing Ex2.4. Change the *color* of the *Apple Inc* link to *red* when use hovers mouse over it. Change it to *blue* when the link is *active*.

Listing Ex2.4: c02ex4.html

```
<!DOCTYPE html>
<html>

<head>
<title> Psuedo Classes</title>
<style>

</style>
</head>

<body>
<div>
<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p><a
href="http://www.apple.com" class="apple">Apple Inc</a> is an American
company that sells consumer electronics, computer software, and online
services. </p> </div>
```

```
</body>
```

```
</html>
```

Exercise 5

Use the template shown in Listing Ex2.5. Place the *underline* under the *first line* of the paragraph defined by *id1*.

Listing Ex2.5: c02ex5.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> First Line Selector </title>
```

```
<style>
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2> About Apple </h2>
```

```
<p id="id1">Apple is the round fruit of a tree of the rose family. It typically  
has thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American  
company that sells consumer electronics, computer software, and online  
services. </p>
```

```
</body>
```

```
</html>
```

Summary

In this chapter, I explained various CSS selectors such as basic selectors, pseudo-element selectors, and pseudo-class selectors. The knowledge you have gained in this chapter will help you immensely in the later chapters.

3

Working With the Box Model

class element

Key Points and Concepts

- Understanding the Box model
- Assigning padding to the elements
- Assigning margin to the elements
- Centering content on a page
- Working with the box dimensions
- Setting minimum and maximum sizes of the box
- Dealing with the overflowing content
- Hiding elements
- Switching between the inline and block-level display
- Creating floating boxes
- Creating box shadows

Padding

The *padding* properties allow you to *add space* between the *content* and *its border*. You can set padding for *individual edges* or you can use the *padding shorthand* to declare padding in one line. Generally, padding is specified using *pixels*. However, if padding is specified in *percentages*, the padding is a percentage of the browser window or of the containing box. Listing 3.1 shows how you can apply padding to an element.



Width of the Box

If a *width* is specified for the box, the padding is added to the *width* of the box.

Listing 3.1: c03padding-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Padding Properties </title>
<style>
p{
border: 1px dotted blue
```

}

```
p.example{
padding-top: 10px;
padding-right: 5px;
padding-bottom: 10px;
padding-left: 5px;
border: 1px dotted blue;
```

}

```
</style>
</head>
```

```

<body>
<div>
<h1> About Apple </h1>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family</p> <p class="example"><a href="http://www.apple.com" class="apple">Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. </p> </div>
</body>
</html>

```

In Listing 3.1, I have used different padding for *top/bottom* and *left/right* edges of the box. It is evident from the Figure 1 that the second paragraph on which I have applied the padding is more readable than the first paragraph.

About Apple

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family
Apple Inc is an American company that sells consumer electronics, computer software, and online services.

Figure 1

You can also apply equal padding to all four edges of the box using the *padding* property. For example, if you want to assign *10px* padding to all sides, use the following CSS rule: `padding: 10px;`.

You can also use the *shorthand* where the values will be in clock-wise order: *top*, *right*, *bottom*, and *left*. For example, `padding: 10px, 5px, 3px, 6px;`. The padding property can use any of the syntax shown in Table 3.1.

Table 3.1: The *padding* property syntax

Property

Description	
padding: 10px 5px 8px 12px;	Top: 10px, Right: 5px, Bottom: 8px, Left: 12px
padding: 10px 5px 8px;	Top: 10px, Bottom: 8px, Left & Right: 5px
padding: 10px 15px;	Top & Bottom: 10px, Left & Right: 15px
padding: 10px	All four edges of the box will have 10px padding



padding Property

The value of the **padding** property is not *inherited automatically*. You need to specify padding for *each element* that you need to use.

Margin

The *margin* properties allow you to adjust *space around an element* [outside the border]. It controls the gap between the boxes. Generally, the margin is specified using *pixels* however you can also use *percentages* and *ems*. You can specify the margin using the *margin-top*, *margin-right*, *margin-bottom*, and *margin-left* properties. You can also use a *shorthand* margin value to change all margin at once.

The *margin* property can use any of the syntax shown in Table 3.2.

Table 3.2: The *margin* property syntax

Property

Description	
margin: 10px 5px 8px 12px;	Top: 10px, Right: 5px, Bottom: 8px, Left: 12px
margin: 10px 5px 8px;	Top: 10px, Bottom: 8px, Left & Right: 5px
margin: 10px 15px;	Top & Bottom: 10px, Left & Right: 15px
margin: 10px	All four edges of the box will have 25px margin



margin property

The value of the `margin` property is *not inherited* automatically. You need to specify `margin` for each element that you need to use.

Listing 3.2 shows the use of the `margin` property. You can see the effect in Figure 2. Notice that the `20px` margin is applied to the `top` and `bottom` edges whereas `30px` margin is applied to the `left` and `right` edges.

Listing 3.2: c03margin-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>

<style>
p{
border: 1px dotted blue;

}

p.example{
margin: 20px 30px;
border: 1px dotted blue;

}
```

```
</style>
</head>
<body>
<div>
<h1> About Apple </h1>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family</p> <p class="example"><a href="http://www.apple.com" class="apple">Apple Inc</a> is an American company that sells consumer electronics, computer software, and online services. </p> </div>
</body>
</html>
```

About Apple

Apple is usually round, red or yellow, edible fruit of a small tree.
Malus sylvestris, of the rose family

[Apple Inc](http://www.apple.com) is an American company that sells consumer electronics, computer software, and online services.

Figure 2

Centering Content

You can use the *margin* property to center a box on the page or center it in its containing box. To do so, set the *left-margin* and *right-margin* properties to *auto*. In order to center the box, you need to specify width of the box or it will take entire width available to it. Listing 3.3 shows the described method in action. Figure 3 shows the box defined by the *example* class at the center of the page.

Listing 3.3: c03centering-content.html

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>

<style>
p{
border: 1px dotted blue;
width: 300px;
}

p.example{
margin: 20px 30px;
border: 1px dotted blue;
margin: 10px auto;
}

</style>
</head>
<body>
<div>
<h1> About Apple </h1>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family</p> <p class="example"><a
```

```
href="http://www.apple.com" class="apple">Apple Inc</a> is an American  
company that sells consumer electronics, computer software, and online  
services. </p> </div>  
</body>  
</html>
```

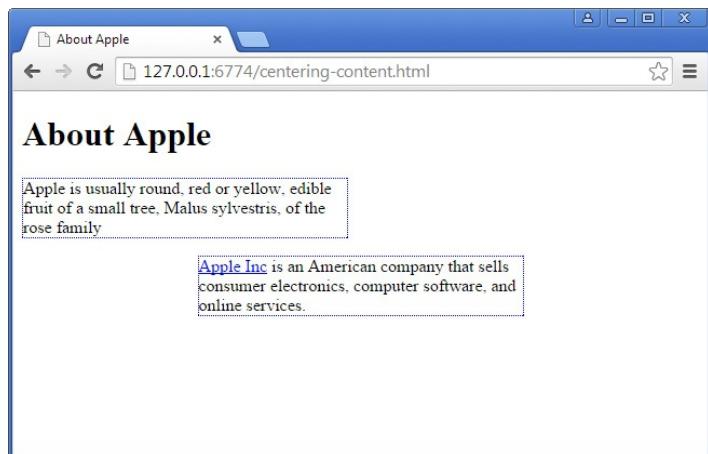


Figure 3

Box Dimensions

You can use the *height* and *width* properties to set the dimensions of a box. You can set these properties in *pixel*, *percentages*, or *ems*. When you use percentages, the size of the box is *relative* to the browser or to the containing box. When you use *ems*, the size is dependent on the *size of text* within it. While calculating the *width* and *height* of a box, you should also include the *padding*, *border*, and *margin* dimensions. The total width and height should be calculated as follows:

Total Width = Width + Left Padding + Right Padding + Left Border + Right Border + Left Margin + Right Margin

Total Height = Height + Top Padding + Bottom Padding + Top Border + Bottom Border + Top Margin + Bottom margin.

Consider the following rule:

```
div{  
    width=300px;  
    padding:10px;  
    margin-right:10p;  
  
}
```

The total width of the box will be *340px* [*Width+Left Padding+Right Padding+Margin Right*].

Listing 3.4 shows the use of the *width* and *height* properties. Figure 4 shows the result. Notice the size of the *div* element defined by the *class2* class is *75%* of the size of the *div* element defined by the *class1* class. In other words, the size of the *div* element defined by the *class2* class is *300 pixel* wide and *225 pixels* high.

Listing 3.4: c03box-dimensions.html

```
<!DOCTYPE html>
<html>
<head>
<title> Box Dimensions </title>

<style>
div.class1 {
width: 400px;
height: 300px;
background-color: lightgreen;

}

div.class2 {
width: 75%;
height: 75%;
background-color: lightyellow;
margin: 10px auto;

}

</style>
</head>
<body>
<div class="class1">
<div class="class2">
The width and height properties in action !
</div>
</div>
</body>
</html>
```



Figure 4

Setting Minimum and Maximum Sizes You can use the *min-* and *max-* properties to set the limits for *width* and *height*. These properties allow browser to freely adjust the content within the limits. The web pages with the responsive design *shrink to fit* the screen size. You can control how small a box can be displayed using the *min-width* property. The *max-width* property allows you to set the maximum width a box can stretch. In Listing 3.5, I have used the *min-width* and *max-width* properties to control the width of the table cells. Figure 5 shows the result. Resize your browser to see the effect.

Listing 3.5: c03limiting-dimensions.html

```
<!DOCTYPE html>
<html>
<head>
<title> Limiting Dimensions </title> <style>
.desc{
min-width: 250px;
max-width: 350px;
```

}

```
td, tr{
border: 1px dotted blue;
```

}

```

</style>
</head>
<body>
<table class="tble">
<tr>
<td> Icon </td>
<td> Description </td>
</tr>
<tr>
<td><td>
<td class="desc"> Apple is usually round, red or yellow, edible fruit of a small
tree, Malus sylvestris, of the rose family</td>
</tr>
<tr>
<td><td>
<td class="desc"> Apple is an American company that sells consumer
electronics, computer software, and online services. </td> </tr>
</table>
</body>
</html>

```

Icon	Description
	Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family
	Apple is an American company that sells consumer electronics, computer software, and online services.

Figure 5

Dealing with Overflowing Content The *overflow* property allows you to control the behavior of the browser when the content of a box is larger than the box. Table 3.3 shows the *overflow* properties. Table 3.4 shows the allowed values for this property.

Table 3.3: *Overflow* properties

Property

Description
overflow-x

Sets the horizontal overflow style
overflow-y

Sets the vertical overflow style

`overflow`

Shorthand property

Table 3.4: *Overflow* property values

Value

Description

visible	This is the default value. The overflow is not clipped.
---------	---

hidden|The overflow is clipped and rest of content is hidden.

scroll|The overflow is clipped, scroll bar appears to view the clipped content.

auto	If the overflow is clipped, scroll bar appears to see the rest of the content.
------	--

initial	Sets property to its default value.
---------	-------------------------------------

inherit	Inherits this property from the parent element.
---------	---

Listing 3.6 shows the use of the *overflow* property. In Figure 6, you can see two boxes whose contents expand beyond the box size. For the first box, the *overflow* property is set to *scroll*. For the second box, it is set to *hidden*.

Listing 3.6: c03overflow.html

```
<!DOCTYPE html>
<html>
<head>
<title> Overflow </title>
<style>
p.scroll{
width:100px;
height:100px;
overflow: scroll;

}

p.hidden{
width:100px;
height:100px;
overflow: hidden;

}

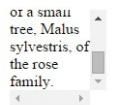
</style>
</head>
<body>
```

```
<h2>overflow:scroll</h2>
<p class="scroll">Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
<h2>ovreflow:hidden</h2>
<p class="hidden">Apple Inc is an American company that sells consumer electronics, computer software, and online services.</p> </body>
</html>
```

Hiding Elements

You can hide elements using the *visibility* property. This property takes three values: *collapse*, *hidden*, and *visible*. The *default* value is *visible* which makes the element visible on the page. The *collapse* value hides the element but element doesn't occupy space in the page layout. The *hidden* value hides the element but element occupies the space on the page.

overflow:scroll



or a small
tree, Malus
sylvestris, of
the rose
family.

overflow:hidden



Apple Inc is an
American
company that
sells consumer
electronics.
[commentar](#)

Figure 6



overflow property

This property is not particularly useful when used on its own however you can create some cool effects with *JavaScript*.

Element Box Type

The *display* property lets you turn an *inline* element into a *block-level* element or vice versa. It supports many values, the commonly used values this property takes are listed in Table 3.5.

Table 3.5: *display* properties

Value

Description

inline	Turns a block-level element to an inline element
--------	--

block	Turns a inline element to a block-level element
inline-block	Turns a block-level element to an inline element while retaining other features of a block-level element.

none	Hide the elements from the page and it does not take any space in the page.
------	---

Listing 3.7 shows the use of this property. In Figure 7, notice that the list items are displayed as *block-level* elements and are appearing in a *single* row.

Listing 3.7: c03display.html

```
<!DOCTYPE html>
<html>
<head>
<title> Display </title>
<style>
li {
display: inline;
padding-right: 10px;
}

</style>
</head>
<body>
<ul>
<li>Item-1</li>
<li>Item-2</li>
<li>Item-3</li>
<li>Item-4</li>
<li>Item-5</li>
</ul>

</body>
</html>
```

Item-1 Item-2 Item-3 Item-4 Item-5

Figure 7

Creating Floating Boxes

The `float` property allows you to create the floating boxes. The boxes are shifted to one side of the page until the `left` or `right` edge touches of the containing box or another floating box in the page. This property takes three values: `left`, `right`, and `none`, which are self-explanatory.

In Listing 3.8, I have floated content of the six paragraphs to the `left`. These six paragraphs are contained within the body element whose width is set to `750px`. Figure 8 shows the result.

Listing 3.8: c03float.html

```
<!DOCTYPE html>
<html>
<head>
<title> Float </title>
<style>
body{
width: 750px;

}

p{
float:left;
width: 200px;
border: 1px dotted blue;
padding: 10px;
margin: 10px;

}

</style>
```

```
</head>
<body>

<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family. </p>
<p>Apple Inc is an American company that sells consumer electronics, computer software, and online services.</p>
<p>To know more about Apple, use Google search.</p>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
<p>Apple Inc is an American company that sells consumer electronics, computer software, and online services.</p>
<p>To know more about Apple, use Google search.</p> </body>
</html>
```

In Figure 8, notice that the fourth paragraph sits below the third paragraph. It did not move to below the first paragraph because there is space below the third paragraph. To overcome this issue, use the **clear** property. I have used this property in Listing 3.9. Figure 9 shows the result.

Listing 3.9: c03float-clear.html

```
<!DOCTYPE html>
<html>
<head>
<title> Float </title>
<style>
body{
width: 750px;
}
}
```

```
p{  
float: left;  
width: 200px;  
border: 1px dotted blue;  
padding: 10px;  
margin: 10px;  
  
}  
}
```

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

```
</style>  
</head>  
<body>
```

```
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus  
sylvestris, of the rose family. </p>  
<p>Apple Inc is an American company that sells consumer electronics,  
computer software, and online services.</p>  
<p>To know more about Apple, use Google search.</p>  
<p class="clear">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p>Apple Inc is an American company that sells consumer electronics,  
computer software, and online services.</p>  
<p>To know more about Apple, use Google search.</p> </body>  
</html>
```

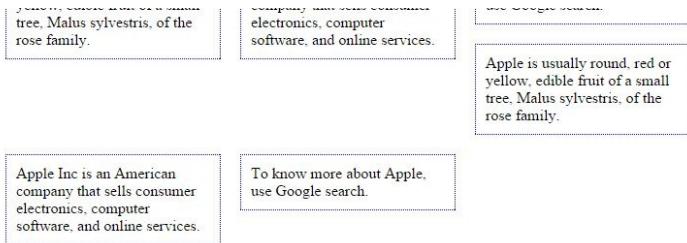


Figure 8



Figure 9

Creating Box Shadow

The *box-shadow* property allows you to add *drop-shadow* to a box. It takes the following format:

box-shadow: hoffset voffset blur spread color inset; Table 3.6 shows the values of the *box-shadow* property.

Table 3.6: Values of the *box-shadow* property

Value

Description

hoffset	Horizontal offset of the shadow. Negative values position the shadow to the left of the box.
---------	--

voffset	Vertical offset of the shadow. Negative values position the shadow to the top of the box.
---------	---

blur	[Optional] Blurs the shadow. If omitted, the shadow is a solid line like a border
------	---

spread	[Optional] The spread of the shadow. Positive values make the shadow expand in all directions.
--------	--

color	[Optional] The color of the shadow.
-------	-------------------------------------

inset	[Optional] Makes the shadows to be inset inside the box.
-------	--

Listing 3.10 shows the use of the *box-shadow* property. Figure 10 shows the effect of the style.

Listing 3.10: c03box-shadow-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Box Shadow </title>
<style>
p{
padding: 10px;
margin: 10px;
box-shadow: 6px 10.392px 4.95px 0.05px #7e8325;

}

</style>
</head>
<body>

<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family. Apple Inc is an American company that sells consumer electronics, computer software, and online services. </p>
</body>
</html>
```

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family. Apple Inc is an American company that sells consumer electronics, computer software, and online services.

Figure 10

You can also define multiple shadows in a *single box-shadow declaration*. Listing 3.11 shows the use. Here, *two shadows* are defined one of which is *inset*. Figure 11 shows the result.

Listing 3.11: c03box-shadow-2.html

```
<!DOCTYPE html>
<html>
<head>
<title> Box Shadow </title>
<style>
p{
padding: 10px;
margin: 10px;
background-color: #7b7f24;
color: white;
box-shadow: 6px 10.392px 4.95px 0.05px #7e8325, inset 10px 5.321px
66.88px 9.12px #000; }
</style>
</head>
<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family. Apple Inc is an American company that sells consumer electronics, computer software, and online services. </p>
</body>
</html>
```

Apple is usually round, red or yellow, edible fruit of a small tree, *Malus sylvestris*, of the rose family. Apple Inc is an American company that sells consumer electronics, computer software, and online services.

Figure 11

CSS Browser Prefixes

The browser vendors regularly add extensions to the existing CSS specifications. These extensions may not be part of the formal specification and generally proprietary to their browsers. In most cases, these features are developed according to the *W3C* drafts that have not reached the *Candidate Recommendation* status. The browser prefixes are used to test the extension before they become part of the regular specification.

Following are some browser prefixes:

Android: -webkit—

Chrome: -webkit—

Firefox: -moz—

Internet Explorer: -ms—

iOS: -webkit—

Opera: -o—

Safari: -webkit-

To use the prefix, take the standard CSS property and then add prefix for each browser. For example, if you want to use the prefixes with the *transition* property, first write the standard syntax and then use the prefixes. For example:

```
transition: all 4s ease;  
-webkit-transition: all 4s ease;  
-moz-transition: all 4s ease;  
-ms-transition: all 4s ease;  
-o-transition: all 4s ease;
```

Note that I have added prefixes in the following format: *-prefix-*.

I personally use the following service to check the browser compatibility: <http://caniuse.com>. For example, if you type *transition*

in the *Can I use* field, the results with the browser compatibility are displayed below the field (see Figure 12).

Notice in Figure 13, the *transition* property is not supported by *version 8* of the *Opera Mini* browser. Also, if you want to use this property with versions *4.1* and *4.3*, you have to use the prefix *-webkit-*.



Google Chrome I have used version *41* of the *Google Chrome* browser to test the code in this book.

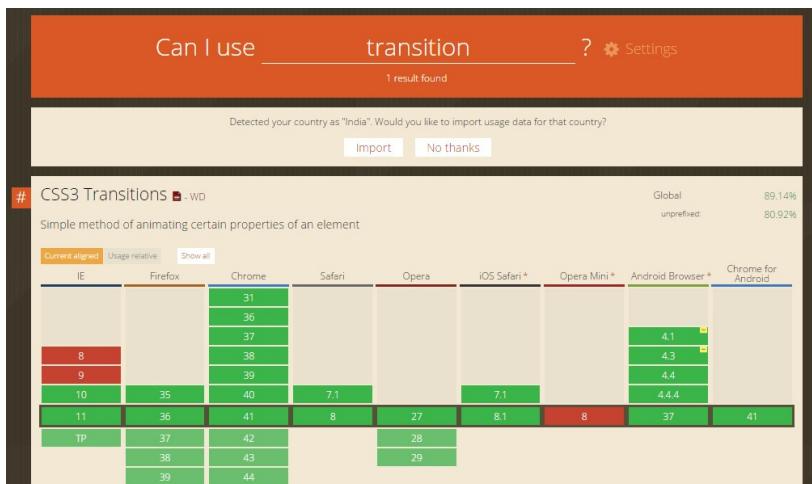


Figure 12



Figure 13

Exercises

Exercise 1

Write the CSS rule that represents every `<p>` element that is the *first child element* of its *parent* and then set background color of the `<p>` element to *green*.

Exercise 2

Using Listing Ex3.1, set the color of the `<h1>` element to *blue* using the `:first-child` selector.

Listing Ex3.1: c03ex2.html

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>
<style></style>
</head>
<body>
<div>
<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p>Apple Inc. is an American
company that sells consumer electronics, computer software, and online
services. </p> </div>
</body>
</html>
```

Exercise 3

Using the Listing Ex3.1, set the following properties when you click on the `<h1>` or `<p>` element:

1. Set the *background color* of these elements to *light yellow*.
2. Set the *1px dotted border* around these elements. Set the *color* of the *border* to *black*.
3. Set *padding* of *10px*.

Exercise 4

Using the Listing Ex3.2, change the *color* of the *link* in the *second paragraph* to *red* when you *click* or *hover* on the link:

[Listing Ex3.2: c03ex4.html](#)

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>
</head>
<body>
<div>
<h1> About Apple </h1>
<p>Apple is the round fruit of a tree of the rose family, which typically has
thin green or red skin and crisp flesh.</p> <p><a
href="http://www.apple.com" class="apple">Apple Inc</a> is an American
company that sells consumer electronics, computer software, and online
services. </p> </div>
</body>
</html>
```

Exercise 5

Write the CSS rule to set the *background color* of all elements to *red* that are *not* of the *h1* type.

Summary

In this chapter, I have explained the *box* model to you. You learned about setting the *dimensions* of the box as well as how to adjust *space around the box* and *its contents* using the *padding* and *margin* properties. I also discussed about controlling the *overflowing* content, *hiding* elements, *creating floating* boxes, and switching between the *block* and *inline* display. You also learned about creating *box shadows* for the boxes. The knowledge you gained about the box model in this chapter will help you immensely when you will work on some examples and exercises in the coming chapters.

4

Working With Borders and Backgrounds

padding *margin*

borders

margin

padding

Key Points and Concepts

- Applying a border to an element
- Defining the border width and type
- Border shorthand
- Creating rounded corners
- Using images as borders
- Setting background color and images
- Using the background shorthand
- Creating gradients

Applying a Border to an Element There are three basic *border* properties that controls the border around elements. Table 4.1 describes these properties. Listing 4.1 shows these properties in action.

Table 4.1: Basic border properties

Property

Description	
border-width	Controls the width of the border.
border-style	Controls the style used to draw the border.
border-color	Controls the color of the border.
border-top-width border-top-style border-top-color	Controls the top border.
border-bottom-width border-bottom-style border-bottom-color	Controls the bottom border.
border-left-width border-left-style border-left-color	Controls the left border.
border-right-width border-right-style border-right-color	Controls the right border.

Listing 4.1: c04applying-border.html

```

<!DOCTYPE html>
<html>

<head>
<title> Applying Border </title>
<style>
p {
padding: 10px;
border-width: 2px;
border-style: double;
border-color: darkred;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus

```

```
sylvestris, of the rose family.</p> <p>Apple Inc is an American company that  
sells consumer electronics, computer software, and online services.</p>  
</body>  
  
</html>
```

In Listing 4.1, I have applied border to the paragraph elements using *border-width*, *border-style*, and *border-color* properties. Figure 1 shows the *border* around the *paragraphs*.

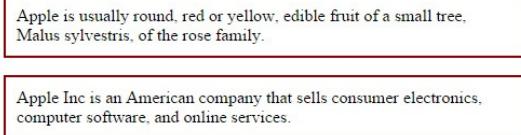


Figure 1

You can set the border width using CSS measurement units such as *em*, *px*, or *cm*. You can also specify the *percentages*. When *percentage* is used, browser draws border with *percentage of the width* of the area around which the border is drawn. You can also use *preset widths*. There presets are: *thin*, *medium*, and *thick*.

Values of the border-style Property In Listing 4.1, I have used *double* as *border style* which causes border to appear as *two parallel lines* with some *gap* between them. The *border-style* property supports other styles as well. Table 4.2 lists them.

Table 4.2: Values of the *border-style* property

Values

Description

none	Displays no border.
------	---------------------

hidden	Like none, displays no border. In case, there is conflicting border is set in a table, the hidden value has the highest priority. In other words, the conflicting border will not be displayed.
--------	---

dotted	Displays border as a series of rounded dots.
--------	--

dashed	Displays border as a series of short square ended dots.
--------	---

solid	Displays border as a single, straight, solid line.
-------	--

double	Displays border as two straight line with a gap between them.
--------	---

groove	Displays border as leading to a carved effect. It is opposite of the ridge style.
--------	---

ridge	Displays border with 3D effect.
-------	---------------------------------

inset|Displays border in a way that makes the box appear embedded.

outset	Displays border in a way that makes the box appear embossed.
--------	--

Listing 4.2 shows the demonstration of the *values* shown in Table 4.2. Figure 2 shows the result.

Listing 4.2: c04border-style.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border Style </title>
<style>
body {
width: 750px;

}

p {
float: left;
width: 150px;
height: 100px;
background-color: bisque;
padding: 5px;
margin: 5px;

}

.b1 {
border-style: none;

}
```

```
.b2 {  
border-style: hidden;  
}  
}
```

```
.b3 {  
border-style: dotted;  
}  
}
```

```
.b4 {  
border-style: dashed;  
}  
}
```

```
.b5 {  
border-style: solid;  
}  
}
```

```
.b6 {  
border-style: double;  
}  
}
```

```
.b7 {  
border-style: groove;  
}  
}
```

```
.b8 {  
border-style: ridge;
```

}

```
.b9 {  
border-style: inset;
```

}

```
.b10 {  
border-style: outset;
```

}

```
</style>  
</head>
```

```
<body>  
<p class="b1">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b2">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b3">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b4">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b5">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b6">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b7">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b8">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b9">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
<p class="b10">Apple is usually round, red or yellow, edible fruit of a small
```

tree, *Malus sylvestris*, of the rose family.</p></body>

</html>

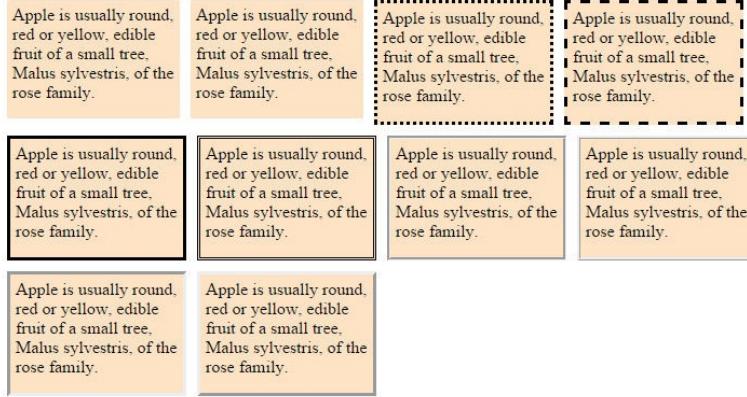


Figure 2

Using the border Shorthand Property You can use the *border shorthand property* to set all border properties in one declaration. You need to write the values that can be used in the following order: *border-width*, *border-style*, and *border-color*.

In Listing 4.3, I have used the *border shorthand property* to draw a 3 pixel, *dotted* border around a *paragraph* element. Figure 3 shows the result.

Listing 4.3: \c04\border-shorthand.html

```
<!DOCTYPE html>
<html>

<head>
<title> Applying Border </title>
<style>
p {
padding: 5px;
margin: 10px;
border: 3px dotted #4557bc;
```

```
        }
```

```
</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>

</html>
```

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.

Figure 3



border Shorthand Property

If you are using the **border shorthand** property, it does not matter if one of the values is missing. For example, the following example is valid: **border: solid #000;**

Creating Rounded Corners

The *border-radius* property gives you ability to create *rounder corners* on any box. The value of this property indicates the *size* of the *radius* in pixels. Table 4.3 shows the list properties for creating *rounder corners*. Notice in Figures 4 and 5, the *radii* values project an oval to create corner of the box.

Table 4.3: *Border radius* properties

Property

Description	
border-top-left-radius	Controls the radius of the top left corner of the box.
border-top-right-radius	Controls the radius of the top right corner of the box.
border-bottom-left-radius	Controls the radius of the bottom left corner of the box.
border-bottom-right-radius	Controls the radius of the bottom right corner of the box.
border-radius	This is a shorthand property which sets all corners at once.

To define a *curved corner*, you need to specify *two radii values* for the corner. The first value specifies the *horizontal radius* whereas the second value specifies the *vertical radius*. If you are using *percentages*, the values are of the vertical and the horizontal size of the box of the element. Listing 4.4 shows the use of the *border-bottom-right-radius* property. Figure 4 shows the result.

Listing 4.4: c04border-radius-1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border Radius </title>
<style>
div {
width: 300px;
height: 100px;
padding: 20px;
margin: 10px;
border: 1px solid blue;
border-bottom-right-radius: 80px 50px;
}

</style>
</head>
```

```

<body>
<div>Apple is usually round, red or yellow, edible fruit of a small tree, Malus
sylvestris, of the rose family.</div>
</body>

</html>

```

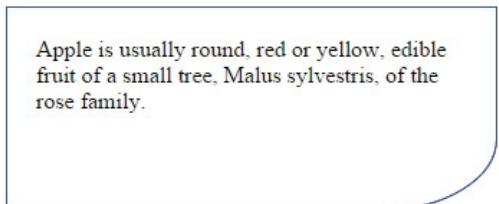


Figure 4

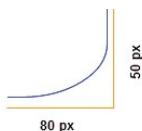


Figure 5



One Radius Value

If you supply only **one radius**, both the **horizontal** and **vertical** radii will use the **same value**.



Padding Property

If border is touching the text in the element due to any corner of the element, you can create the space between the content and corner by using the **padding** property as I have done in listing 4.4.

Using the Shorthand Property The **border-radius** shorthand property is used to set the four **border-* -radius** properties at **once**. For example:

`border-radius: 10px;`

is equivalent to:

```

border-top-left-radius:10px;
border-top-right-radius: 10px;
border-bottom-right-radius: 10px;
border-bottom-left-radius: 10px;

```

The syntax for the *first radius* allows one of the following *four* values:

border-radius: radius

border-radius: top-left-and-bottom-right top-right-and-bottom-left border-

radius: top-left top-right-and-bottom-left bottom-right border-radius: top-left

top-right bottom-right bottom-left

The syntax of the *second radius* also allows you to use one of the following *four* values:

border-radius: (first radius values) / radius

border-radius: (first radius values) / top-left-and-bottom-right top-right-and-

bottom-left border-radius: (first radius values) / top-left top-right-and-bottom-

left bottom-right border-radius: (first radius values) / top-left top-right bottom-

right bottom-left

The declaration, border-radius: *2em 1em 4em / 0.5em 3em;*, is equivalent to:

border-top-left-radius: 2em 0.5em;

border-top-right-radius: 1em 3em;

border-bottom-right-radius: 4em 0.5em;

border-bottom-left-radius: 1em 3em;

Listing 4.5 demonstrate an example.

Listing 4.5: c04border-radius-2.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> Border Radius </title>
```

```
<style>
```

```
p {
```

```
border: 1px solid firebrick;
```

```
padding: 10px;
```

```
}
```

```

p.cls1 {
border-radius: 20px / 15px;

}

p.cls2 {
border-radius: 2em 1em 4em / 0.5em 3em;

}

</style>
</head>

<body>
<p class="cls1">Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p> <p class="cls2">Apple Inc is an American company that sells consumer electronics, computer software, and online services.</p>
</body>

</html>

```

In Listing 4.5, there are two paragraphs represented by the *cls1* and *cls2* classes, respectively. The declarations for the *cls1* class define two values which are applied to all four corners, refer to Figure 6. Notice the horizontal values are separated from the vertical values using the / character.

Notice the declarations in the *cls2* class. The first four values represents the *horizontal radius values* followed by a / character and then the next four values are *vertical radius values*, refer Figure 6.

Apple is usually round, red or yellow, edible fruit of a small tree.
Malus sylvestris, of the rose family.

Apple Inc is an American company that sells consumer
electronics, computer software, and online services.

Figure 6

Using Images as Borders

The *border-image* property allows you to draw an image on the borders of an element. It allows you to create beautiful looking borders. Table 4.4 shows the *border-image* properties.

Table 4.4: The *border-image* properties

Property

Description	
border-image-source	Defines the source of the image.
border-image-slice	Controls the offsets for slicing the image
border-image-width	Controls the width of the border
border-image-outset	Defines the area outside the standard border that is used to show the border of the image.
border-image-repeat	Controls the repetition of the image along the border. It can take four values: stretch, repeat, round, or space.
border-image	This is the shorthand property to set all values at once.

Before you use any image as border, you need to *slice* the image. You use numeric values to specify offset values that browser use to slice the image into nine parts. To demonstrate how browser slice an image, I have created an image as shown in Figure 7. Each diamond in Figure 7 is *30px* wide and *30px* tall that makes image *90px* wide and *90px* tall. The middle diamond in Figure 7 is transparent. Listing 4.6 shows the use of the *border-image* properties. Figure 8 shows the result.

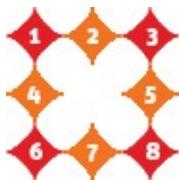


Figure 7

Listing 4.6: c04border-image-1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border Image </title>
<style>
p {
```

```

border: 1px solid firebrick;
padding: 10px;

}

p.cls1 {
border: 30px solid transparent;
border-image: url("images/diamond.png");
border-image-slice: 30;
border-image-outset: 5px;

}

</style>
</head>

<body>
<p class="cls1">Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>

</html>

```

Notice in Figure 8, the diamond marked as **1**, **3**, **6**, and **8** are used to draw the corners of the border whereas the diamond marked as **2**, **4**, **5**, and **7** are used to draw the border edges. Also, I have set **30px** inward offsets of the image border as well as **5px** image outset that controls the amount by which the border image area extends beyond the border box.

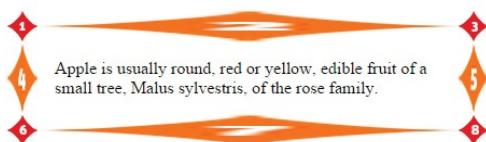


Figure 8



Specifying Slice Size

Notice the declaration in `p.cls1` rule, while specifying the `slice` size, you don't have to provide the units. The units are assumed `pixels`, by default.

You can also use the `border-image` shorthand property. The syntax for this property is as follows:

`border-image: source slice width outset repeat;` Listing 4.7 shows the use of this property. Figure 9 shows the result. Listing 4.8 and Figure 10 demonstrate another example.

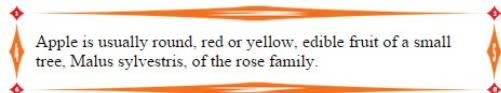


Figure 9

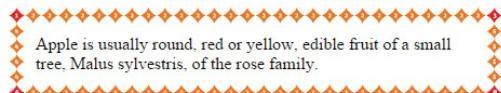


Figure 10

Listing 4.7: c04border-image-2.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border Image </title>
<style>
p {
border: 1px solid firebrick;
padding: 10px;

}

p.cls1 {
border: 15px solid transparent;
```

```
border-image: url("images/diamond.png") 30 30 stretch; }  
</style>  
</head>  
  
<body>  
<p class="cls1">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
</body>  
  
</html>
```

Listing 4.8: c04border-image-3.html

```
<!DOCTYPE html>  
<html>  
  
<head>  
<title> Border Image </title>  
<style>  
p {  
border: 1px solid firebrick;  
padding: 10px;  
  
}  
  
p.cls1 {  
border: 15px solid transparent;  
border-image: url("images/diamond.png") 30 30 round; }  
</style>  
</head>  
  
<body>  
<p class="cls1">Apple is usually round, red or yellow, edible fruit of a small  
tree, Malus sylvestris, of the rose family.</p>  
</body>  
  
</html>
```

Listing 4.8a shows more examples of border-radius property. Figure 10a shows the result.

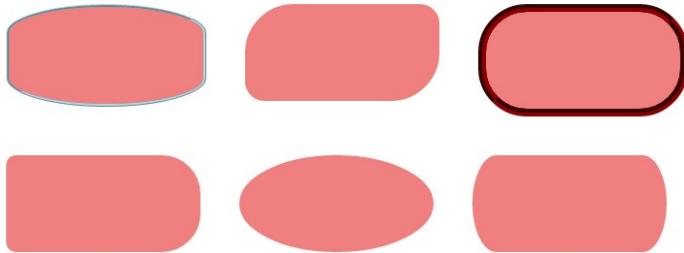


Figure 10a

Listing 4.8a: c04border-radius-3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border and Backgrounds </title> <style>
#box {
width: 200px;
height: 100px;
border: ridge lightblue;
background-color: lightcoral;
border-radius: 120px/36px;
float: left;
margin: 20px;

}

#box1 {
width: 200px;
height: 100px;
border: none;
background-color: lightcoral;
border-radius: 50px 20px;
```

```
float: left;  
margin: 20px;
```

```
}
```

```
#box2 {  
width: 200px;  
height: 100px;  
border: groove 8px darkred;  
background-color: lightcoral;  
border-radius: 50px;  
float: left;  
margin: 20px;
```

```
}
```

```
#box3 {  
width: 200px;  
height: 100px;  
border: none;  
background-color: lightcoral;  
border-radius: 10px 40px 40px 10px;  
float: left;  
margin: 20px;
```

```
}
```

```
#box4 {  
width: 200px;  
height: 100px;  
border: none;  
background-color: lightcoral;  
border-radius: 70%;  
float: left;  
margin: 20px;
```

```
}
```

```
#box5 {  
width: 200px;  
height: 100px;  
border: none;  
background-color: lightcoral;  
border-radius: 20%/80%;  
float: left;  
margin: 20px;
```

```
}
```

```
</style>  
</head>
```

```
<body>  
<div id="box"></div>  
<div id="box1"></div>  
<div id="box2"></div>  
<div id="box3"></div>  
<div id="box4"></div>  
<div id="box5"></div>  
</body>
```

```
</html>
```

Setting Background Color and Images You can use the *background* properties to style *background behind content*. The properties are summarized in Table 4.5.

Table 4.5: The *background* properties

Property

Description

background	The shorthand property for styling background.
background-color	Defines the background color of the content.
background-image	Sets the background image for element.
background-repeat	Sets the repeat style for the background image.
background-size	Sets the size of the background.
background-position	Sets the position of the background image.
background-attachment	Sets the attachment style. Defines whether the background images are fixed or scrolls with the rest of the page content.
background-clip	Defines the painting area of the background image.
background-origin	Sets the positioning area of the background image.

The *background-image* property allows you to place an image *behind* any HTML element. By default, the background image fills the *entire* page or box. In listing 4.9, I have used a pattern image (*pattern.jpg*) to apply it to the whole page because it is being used with the *body* element. Figure 11 shows the result.

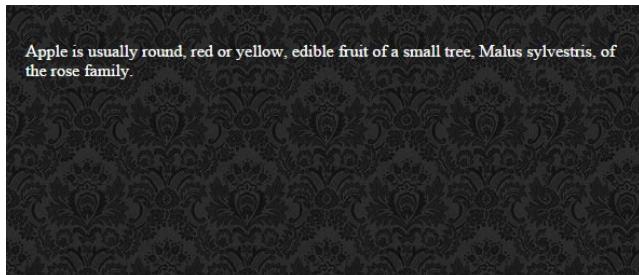


Figure 11

Listing 4.9: c04background-image.html

```
<!DOCTYPE html>
<html>

<head>
<title> Background Image </title>
```

```

<style>
Body {
Background-image: url(images/pattern.jpg);
padding: 10px;
color: white;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>

</html>

```

In Listing 4.10, I have applied the image just to a paragraph. Figure 12 shows the result.

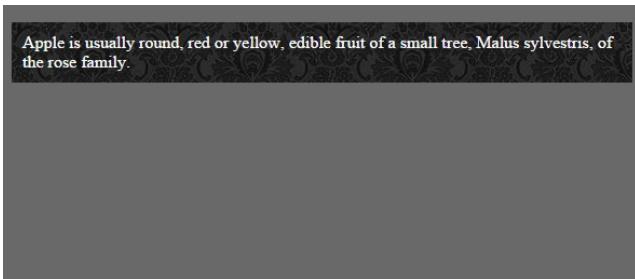


Figure 12

Listing 4.10: c04background-image-1.html

```

<!DOCTYPE html>
<html>

<head>
<title> Background Image </title>
<style>

```

```

body {
background-color: dimgrey;

}

p {
Background-image: url(images/pattern.jpg);
padding: 10px;
color: white;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus
sylvestris, of the rose family.</p>
</body>

</html>

```

The ***background-repeat*** property sets the background ***repeat style***. Table 4.6 shows the values of this property.

Table 4.6: The ***background-repeat*** values

Property

Description

`repeat` The background image will be repeated vertically as well as horizontally.

repeat-y	The background image will be repeated vertically.
repeat-x	The background image will be repeated horizontally.
no-repeat	The background image will not be repeated.

In Listing 4.10, I have set the *image repetition* to *horizontal* by using the *repeat-x* value. Figure 13 shows the result.

Listing 4.10: c04background-image-2.html

```
<!DOCTYPE html>
<html>

<head>
<title> Background Image </title>
<style>
body {
background-image: url(images/flower.png);
background-repeat: repeat-x;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus
sylvestris, of the rose family.</p>
</body>

</html>
```

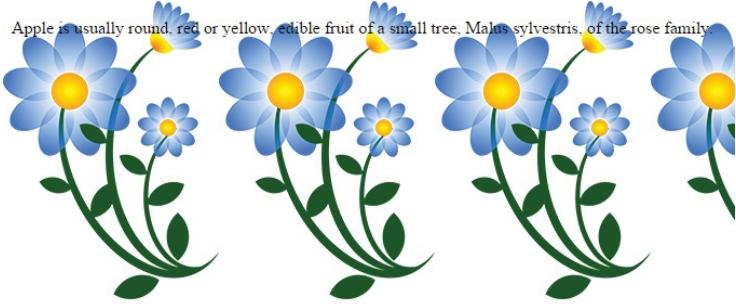


Figure 13

The ***background-attachment*** property lets you specify whether the background image should stay in one position or moves as the user scroll the page. This property can have two values: ***fixed*** and ***scroll***. When you use the fixed value, the image stays in the same position of the page. Listing 4.11 shows the use of this property. Figure 14 shows the result. Now, when you scroll the text the image should stay in the same position.

Listing 4.11: c04background-image-3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Background Image </title>
<style>
body {
background-image: url(images/flower.png);
background-repeat: no-repeat;
background-attachment: fixed;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus
```


left top

left center

left bottom

center top

center center

center bottom

right top

right center

right bottom



background-position Property

If you take one value, second will set to *center*, by default.



background-position Property

You can also use a pair of *pixels* or *percentages*. The pair represents the distance from the *top-left* corner of the browser window. The top left corner is set to *0% 0%*. If you set it to *50% 50%*, the image will be centered horizontally and vertically.

Listing 4.12 place the image at the *center top* position. Figure 15 shows the result.

Listing 4.12: c04background-image-4.html

```
<!DOCTYPE html>
<html>

<head>
<title> Background Image </title>
<style>
body {
background-image: url(images/flower.png);
background-repeat: no-repeat;
background-position: center top;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>

</html>
```

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.



Figure 15

Using the Background Shorthand The *background* property acts as a *shorthand* for all other background properties. The properties must be specified in the following order however you can omit a property if you don't want to use it: *background-color*, *background-image*, *background-repeat*, *background-attachment*, and *background-position*.

Listing 4.13 demonstrates the use of this property. Figure 16 shows the result.

Listing 4.13: c04background-image-5.html

```
<!DOCTYPE html>
<html>

<head>
<title> Background Image </title>
<style>
body {
background: #e7e7e7 url("images/flower.png") no-repeat fixed center; }
</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>
```

</html>

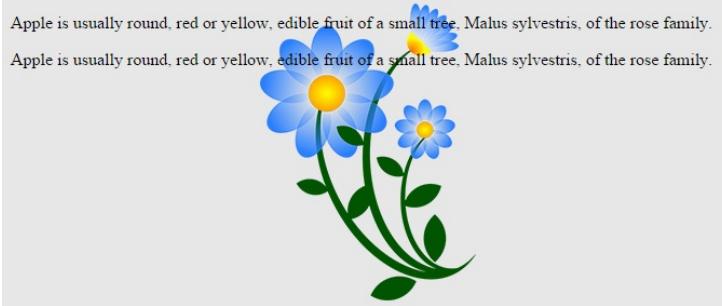


Figure 16

Creating Gradients

You can create a gradient for a background using the *background-image* property. The gradients are created by creating a smooth transition between *two* or *more* specified colors. You can create two types of gradients: *Linear* gradients and *radial* gradients.

The *linear* gradient is most commonly used gradient. The gradient axis can go from left to right, top to bottom, or to any *angle* that you choose. If you don't declare the *angle*, the direction will be assumed *top to bottom*. For specifying colors, you can use *named colors*, *rgba*, or *hsla* values.

In Listing 4.14, I have created a *linear gradient* from top to bottom using the *90* degrees angle. Figure 17 shows the result. If you change the angle to *180* degrees the output will look like something shown in Figure 18.

Listing 4.14: c04background-image-6.html

```
<!DOCTYPE html>
<html>

<head>
<title> Gradients </title>
<style>
p {
background-image: -moz-linear-gradient( 90deg, rgb(255, 109, 0) 0%, 
rgb(255, 182, 0) 26%, rgb(255, 255, 0) 50%, rgb(255, 110, 2) 100%);
background-image: -webkit-linear-gradient( 90deg, rgb(255, 109, 0) 0%, 
rgb(255, 182, 0) 26%, rgb(255, 255, 0) 50%, rgb(255, 110, 2) 100%);
background-image: -ms-linear-gradient( 90deg, rgb(255, 109, 0) 0%, 
rgb(255, 182, 0) 26%, rgb(255, 255, 0) 50%, rgb(255, 110, 2) 100%);
/*default syntax*/
background-image: linear-gradient( 90deg, rgb(255, 109, 0) 0%, 
rgb(255, 182,
```

```
0) 26%, rgb(255, 255, 0) 50%, rgb(255, 110, 2) 100%);  
left: 81px;  
top: 150px;  
width: 365px;  
height: 187px;  
padding: 10px;  
  
}  
  
</style>  
</head>  
  
<body>  
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>  
</body>  
  
</html>
```

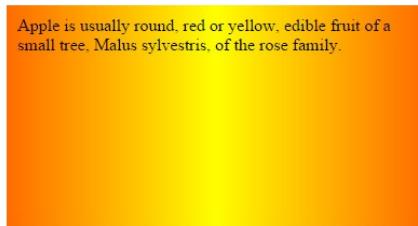


Figure 17

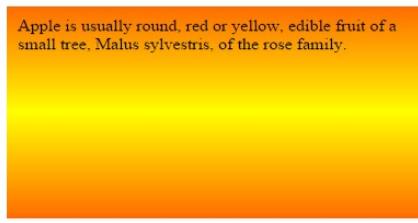


Figure 18

Exercises

Exercise 1

Using the Listing Ex4.1, assign **20%** padding to the *bottom edge* of the paragraph defined by the *example* class.

Listing Ex4.1:

```
<!DOCTYPE html>
<html>
<head>
<title> About Apple</title>

<style>
</style>
</head>
<body>
<div>
<h1> About Apple </h1>
<p class="example">Apple is usually round, red or yellow, edible fruit of a
small tree, Malus sylvestris, of the rose family</p> <p ><a
href="http://www.apple.com" class="apple">Apple Inc</a> is an American
company that sells consumer electronics, computer software, and online
services. </p> </div>
</body>
</html>
```

Exercise 2

Consider the following rule and calculate the *total width* of the element.

```
div {
width: 300px;
padding: 20px;
border: 5px;
margin: 0;
```

}

Exercise 3

Using the template shown in Listing Ex4.2, specify the following settings:

Apply *padding* of *5px* and *margin* of *10px* to the paragraph. Create a *3px* border around the element with *dotted* style and *#4557bc* color. Change the *border style* to *solid* and *color* to *red* when user hovers mouse over it.

Listing Ex4.2: c04ex3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border and Backgrounds </title> <style>
</style>
</head>

<body>
<p class="brdr">Apple is usually round, red or yellow, edible fruit of a small
tree, Malus sylvestris, of the rose family.</p>
</body>

</html>
```

Exercise 4

Using the template shown in Listing Ex4.3 create a *circle* from a *200x200px* box (see Figure 19).

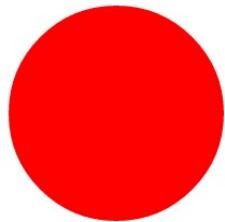


Figure 19

Listing Ex4.3: c04ex4.html

```
<!DOCTYPE html>
<html>

<head>
<title> Border and Backgrounds </title> <style>

</style>
</head>

<body>
<div id="box">

</div>

</body>

</html>
```

Summary

In this chapter, I have shown you the use of the properties that allows you to add *borders*, and *backgrounds* to the elements. You have seen how we can use the images to define borders for an element. I have also showed you how to create rounded corners using the *border* properties. In the end, I have showed you to create background using the gradients. In the next chapter, I will show you the use of the *color* properties.

5

Working with Colors

Chapter 5: Working with Colors

In the previous chapters, I have used the *color*, *border-color*, and *background-color* properties in quite a few listings. In this chapter, you will see how to set *foreground color*, *background color*, and *opacity* for the elements.

Key Points and Concepts • Setting foreground and background colors

- Specifying colors using color names and hex codes
- Specifying colors using rgb and hsl values
- Understanding various color terminologies
- Setting element's opacity

Setting the Foreground and Background Colors The *color* property sets the *foreground* color for the element whereas the *background-color* property is used for the *background* color. If you don't specify a *background* color, the background is *transparent*. Listing 5.1 shows the *color* and *background-color* properties in use.

Listing 5.1: c05color.html

```
<!DOCTYPE html>
<html>

<head>
<title> Applying Color </title> <style>
p {
padding: 5px;
margin: 10px;
border: 3px dotted #4557bc; background-color: rgba(183, 122, 7, 0.2); color:
darkblue;

}

</style>
</head>
```

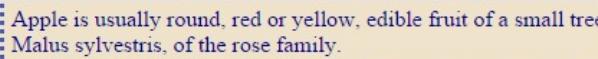
```

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus
sylvestris, of the rose family.</p>
</body>

</html>

```

In listing 5.1, I have used two different styles for defining the color. Figure 1 shows the result. The background color of the paragraph is specified using the *rgba* property. This property allows you to specify a color just like the *rgb* property. However, with *rgba* you can add a fourth value which is known as the *alpha* value. The *alpha* value is a *number* between *0* to *1*. The value *1* represents *100%* opacity, a value of *0.5* represents *50%* opacity, and so on. I have also set the color of the text to *darkblue* using the *color* properties.



Apple is usually round, red or yellow, edible fruit of a small tree,
Malus sylvestris, of the rose family.

Figure 1

In CSS, you can specify a color using *RGB values*, *HSL values*, *HEX Codes*, and *color names*. Table 5.1 summarizes these types and other terminology used.

Table 5.1: The terminologies used for colors

Property	Description
RGB Values	These are values for red, green, and blue color components. These are expressed as numbers from 0 to 255. For example, <code>rgb(255, 0, 0)</code> represents red color.
HEX Codes	These values represent red, green, and blue values as hexadecimal code. These values are preceded by a pound sign (#). For example, <code>#ff0000</code> represents red color.
Color Names	The colors are represented by the predefined names. For example, <code>red</code> can be used to represent red color.
Hue	It's another word for color. A color can also have saturation and brightness.
Saturation	Saturation is the intensity or purity of a hue. It represents amounts of gray in the color. At maximum saturation, there would be no gray.
Brightness	It is sometimes also referred to as value. It represents the amount of black present in the color.

Setting Element's Opacity The *opacity* property allows you to specify the *opacity* of an element. It takes a number from *0* to *1* range. A value of *0.5* represents *50%* opacity, and a value of *0.2* represents *20%* value. Listing 5.2 shows the use of this property.

Listing 5.2: c05color-1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Applying Color </title> <style>
.clr {
background-color: rgb(255, 0, 0); position: absolute;
left: 28px;
top: 32px;
width: 134px;
height: 113px;
z-index: 1;

}

.clr1 {
background-color: rgb(255, 240, 0); position: absolute;
left: 40px;
top: 60px;
width: 128px;
height: 87px;
z-index: 2;
opacity: 0.5;

}
```

```

</style>
</head>

<body>
<div class="clr">
<div class="clr1">

</div>
</div>

</body>

</html>

```

In Listing 5.2, I have created two rectangles defined by the *clr* and *clr1* classes. The *clr1* class represents the *yellow* rectangle. I have set opacity of this rectangle to *50%* using the *opacity* property. Figure 2 shows the result.

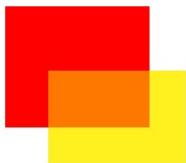


Figure 2

Using the hsl and hsla Properties These two properties are an alternate way to define colors. *HSL* stands for *hue*, *saturation*, and *lightness*. The *hue* value is represented as an angle from *0* to *360* degrees. The *saturation* and *lightness* are represented using *percentages*. For lightness, *0%* is white whereas *100%* is black. Listing 5.3 and Figure 3 demonstrate the use of this property.

Listing 5.3: c05color-2.html

```
<!DOCTYPE html>
```

```
<html>

<head>
<title> Applying Color </title> <style>
.clr {
background-color: hsl(0, 100%, 50%); position: absolute;
left: 28px;
top: 32px;
width: 134px;
height: 113px;
z-index: 1;
```

```
}
```

```
.clr1 {
background-color: hsla(99, 93%, 31%, 0.77); position: absolute;
left: 40px;
top: 60px;
width: 128px;
height: 87px;
z-index: 2;
```

```
}
```

```
</style>
</head>
```

```
<body>
<div class="clr">
<div class="clr1">
```

```
</div>
</div>
```

```
</body>
```

```
</html>
```

Exercises Exercise 1

Write all possible rules to set the *color* of the *paragraph* element to *red*.

Exercise 2

Write the rules to set the *color* of the *paragraph* element to *red* with *40%* translucency.

Exercise 3

Using the template shown in Listing Ex5.1, set the *opacity* of the *img* element to *50%*. Change it to *100%* when user hover mouse over the image.

Listing Ex5.1: c05ex3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Applying Color </title> <style>
</style>
</head>

<body>
 </body>

</html>
```

Summary

In this chapter, I demonstrated the different ways for assigning colors to the elements. You learned setting colors by using *color names*, *hex codes*, *RGB*, and *HSL*. You also gained knowledge about various *terminologies* used for the color.

6

Fomattting Text

formatting

Key Points and Concepts

- Aligning and justifying text
- Specifying text direction
- Specifying letter and word spacing
- Working with word breaks
- Indenting text
- Decorating and transforming text
- Creating drop shadow
- Specifying typefaces
- Using the Google fonts

Aligning and Justifying the Text

The *text-align* and *text-justify* properties are used to align and justify text. Table 6.1 summarizes these properties. Listing 6.1 shows the use of the *text-align* property. The text in the paragraph is aligned to the *right* using the *right* value of the property. Figure 1 shows the result.

Table 6.1: The properties to *align* and *justify* text

Property

Description	
text-align	Controls the alignment for a block of text. It takes the following values: start , end , left , right , center , and justify .
text-justify	This property is used to justify the text. The text-justify property takes one of the following values: auto , none , inter-word , inter-ideograph , inter-cluster , distribute , and kashida .

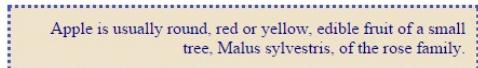


Figure 1

Listing 6.1: c06text-1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
text-align: right;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus
sylvestris, of the rose family.</p>
</body>
```

```
</html>
```

To justify the text, you need to use *text-align* and *text-justify* properties, refer to Listing 6.2. In this listing, the *text-justify* property is set to *inter-word* value which causes spacing to be *distributed between the words*. Figure 2 shows the result.

Listing 6.2: c06text-2.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
text-align: justify;
text-justify: inter-word;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>
```

</html>

Apple is usually round, red or yellow, edible fruit of a small tree. *Malus sylvestris*. of the rose family.

Figure 2

Working with Whitespace

Whitespace such as multiple *spaces*, *line breaks*, and *tabs* is ignored in HTML. The whitespace is *collapsed* to a *single space character* and lines are wrapped. However, you can control (*preserve*) the formatting of whitespace using the *whitespace* property. Table 6.2 shows the values corresponding to this property.

Table 6.2: The *whitespace* property values

Property

Description

normal|The **default** value. It **collapses** the whitespace and **wraps** the lines.

nowrap|The whitespace is collapsed but lines are not wrapped.

pre	The whitespace is preserved and wraps only on line breaks .
pre-line	The whitespace is collapsed to a single whitespace . When necessary, text will be wrapped on line breaks .
pre-wrap	The whitespace is preserved by the browser. The text will wrap on line breaks when necessary.

Listing 6.3 shows the HTML code that I am using with the *whitespace* property. Listing 6.4 shows the *whitespace* property in action. Figure 3 shows the effect of the *pre-line* value. Notice that text is *wrapped* and *line breaks* are *preserved*.

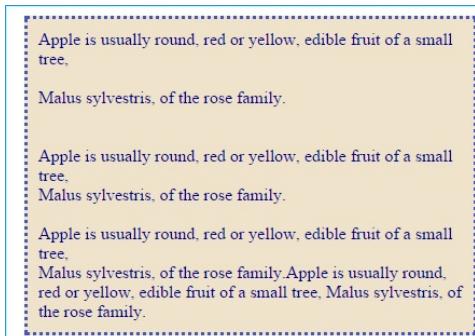


Figure 3

Listing 6.3:

<p>Apple is usually round, red or yellow, edible fruit of a small tree,
Malus sylvestris, of the rose family.

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.Apple is usually round, red or yellow, edible fruit

of a small tree, Malus sylvestris, of the rose family.
</p>

Listing 6.4: c06text-3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
white-space: pre-line;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree,  
Malus sylvestris, of the rose family.

```

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.

Apple is usually round, red or yellow, edible fruit of a small tree, Malus

sylvestris, of the rose family. Apple is usually round, red or yellow, edible fruit of a small tree, *Malus sylvestris*, of the rose family.

</p>

</body>

</html>

Specifying Text Direction

The *direction* property allows you to specify the *text direction/writing* direction. This property takes two values: *ltr* and *rtl*. Listing 6.5 and Figure 4 shows use of this property.

Listing 6.5: c06text-4.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;

}

.cls1 {
direction: ltr;

}

.cls2 {
direction: rtl;

}

</style>
</head>
```

```

<body>
<p class="cls1">Left to right text: Apple is usually round, red or yellow,
edible fruit of a small tree, Malus sylvestris, of the rose family.
</p>

<p class="cls2">Right to left text: Apple is usually round, red or yellow,
edible fruit of a small tree, Malus sylvestris, of the rose family.
</p>

</body>

</html>

```

Figure 4

Specifying Letter and Word Spacing You can use the *letter-spacing*, *word-spacing*, and *line-height* properties to adjust space between the *letters*, *words*, and *lines*. Listing 6.6 demonstrate the use of these properties.

Listing 6.6: c06text-5.html

```

<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);

```

```
color: darkblue;  
word-spacing: 5px;  
letter-spacing: 2px;  
line-height: 1.5em;  
  
}  
  
</style>  
</head>  
  
<body>  
<p>Left to right text: Apple is usually round, red or yellow, edible fruit of a  
small tree, Malus sylvestris, of the rose family.  
</p>  
</body>  
  
</html>
```

The effect of these properties is shown in Figure 5.

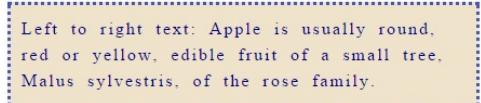


Figure 5

Working with Word Breaks

The **word-break** property allows you to set the *line breaking rules*. This property supports three values: **normal**, **break-all**, and **keep-all**. The **normal** value breaks the words according to their usual roles. The **break-all** breaks the line between *any two* letters. The **keep-all** value prohibits the breaks between *pair of letters*. Figure 6.7 shows the use of this property.

Listing 6.7: c06text-6.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
float: left;

}

.cls1 {
width: 100px;
word-break: keep-all;

}

.cls2 {
width: 100px;
```

```

word-break: break-all;

}

</style>
</head>

<body>
<p class="cls1">Left to right text: Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
<p class="cls2">Left to right text: Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>
</body>

</html>

```

There are two *p* elements in the above listing to which I have applied the *word-break* property. You can see the effect of the properties in Figure 6, the *break-all* value is breaking the line between any two characters.

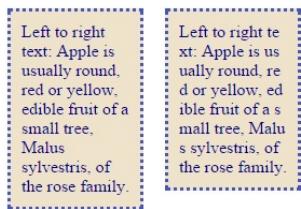


Figure 6

The *word-wrap* property allows to break a long word and wrap onto the next line. It takes two values: *normal* and *break-word*. The *break-word* value lets you break an *unbreakable* word. Listing 6.8 shows the use of this value.

Listing 6.8: c06text-7.html

```

<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
word-wrap: break-word;
width: 200px;

}

</style>
</head>

<body>
<p>Left to right text: Apple is usually round, red or yellow, edible fruit  

of a small tree Malus sylvestris of the rose family.</p>

</body>

</html>

```

The effect of *word-wrap* property is shown in Figure 7.

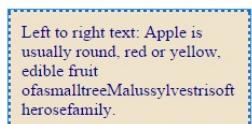


Figure 7

Indenting Text

The ***text-indent*** property lets you specify an *indentation* for the *first line* of block of text. You can specify indent as *length* or as a *percentage*. Listing 6.9 shows the use of this property.

Listing 6.9: c06text-8.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
word-wrap: break-word;
width: 200px;
text-indent: 10%;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p>

</body>

</html>
```

The *indent effect* is shown in Figure 8.

Apple is usually round,
red or yellow, edible fruit of a
small tree, *Malus sylvestris*, of
the rose family.

Figure 8

Decorating and Transforming Text

The *text-decoration* and *text-transform* properties let you *decorate* and *transform* the text. Table 6.3 summarizes the *text-decoration* and *text-transform* property values.

Table 6.3: The *text-decoration* and *text-transform* properties values

Property

Description	
text-decoration	Allows you to decorate the text. It takes the following values: none underline overline line-through blink
text-transform	Allows you to apply transformations to a block of text. It accepts the following values: none capitalize uppercase lowercase

Listing 6.10 shows the use of these values.

Listing 6.10: c06text-9.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;

}

.test1 {
text-decoration: underline;

}

.test2 {
text-decoration: overline;
```

```
}
```

```
.test3 {  
text-decoration: line-through;
```

```
}
```

```
.test4 {  
text-transform: capitalize;
```

```
}
```

```
.test5 {  
text-transform: uppercase;
```

```
}
```

```
.test6 {  
text-transform: lowercase;
```

```
}
```

```
</style>  
</head>
```

```
<body>  
<p>  
<span class="test1"> This text is underlined.</span> <br>  
<span class="test2"> This text is overlined.</span> <br>  
<span class="test3"> This text has a line through it.</span> <br>  
<span class="test4"> The first letter of each word is in uppercase.</span>  
<br>  
<span class="test5"> This is uppercase text.</span> <br>  
<span class="test6"> This is lowercase text.</span> <br>
```

```
</p>  
  
</body>  
  
</html>
```

Figure 9 shows the effect of the *text-decoration* and *text-transform* properties.

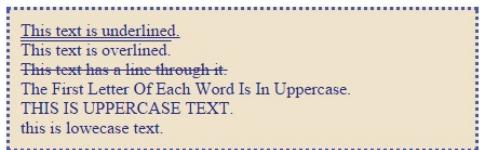


Figure 9

In chapter 3, I showed how to create *drop-shadow* effect for a box. Similarly, you can create drop-shadow effect for a block of text. The *text-shadow* property is used to create drop-shadow. It takes the following values: *h-shadow*, *v-shadow*, *blur-radius*, and *color*. The *h-shadow* and *v-shadow* values are required and controls the position of the *horizontal* and *vertical* shadow, respectively. Listing 6.11 shows the use of this property.



Color

The color can be specified before or after the offsets.

[Listing 6.11: c06text-10.html](#)

```
<!DOCTYPE html>  
<html>  
  
<head>  
<title> Formatting Text </title>  
<style>  
p {  
padding: 10px;
```

```
margin: 10px;  
border: 3px dotted #4557bc;  
background-color: rgba(183, 122, 7, 0.2);  
color: darkblue;  
word-wrap: break-word;  
width: 380px;  
text-shadow: 2px 2px 4px #000000;  
  
}  
  
</style>  
</head>  
  
<body>  
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p> </body>  
  
</html>
```

Figure 10 shows the drop-shadow effect on a block of text.

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.

Figure 10

Working with Fonts

In this section, I will show you how to use *font* properties.

Selecting a Font

The **font-family** property allows you to select the font that you want to use in your design. This property can hold several font names. It works as a **fallback** system, if one font is not available, browser tries the next font. Use a **comma separated** font names with this property.



whitespace

If a font name contains whitespace such as *Times New Roman*, it must be **quoted**: "Times New Roman".

You can use a **family-name** [*times*, *courier*, *arial* etc] or **generic-family** name [*serif*, *sans-serif*, *cursive*, *fantasy*, and *monospace*]. Table 6.4 shows the list of generic font family members.

Table 6.4: The generic font family

Family

Font

serif

Times

sans-serif

Helvetica

cursive|Zapf-Chancery

fantasy

Western

monospace

courier

Listing 6.12 shows the *font-family* property applies to a paragraph.

Listing 6.12: c06text-11.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
word-wrap: break-word;
width: 380px;
font-family: Helvetica, sans-serif;
}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p> </body>

</html>
```

In Listing 6.13, I have used *Helvetica* font for the *font-family*

property. I have used generic *sans-serif* as *fallback* if the browser could not find *Helvetica* in the system. You can see the effect of this property in Figure 11.

.....
Apple is usually round, red or yellow, edible fruit of a
small tree, *Malus sylvestris*, of the rose family.
.....

Figure 11

Setting the Font Size

The *font-size* property enables you to specify the size of the font. There are many ways to specify the font size. The most commons are *pixels*, *percentages*, and *ems*. The pixels are widely used with font size property as it gives you control to precisely how much space text takes. You can specify size in *percentages*. The default size of the text in a browser is *16px*. Therefore, if you want to use a pixel size of *12px*, you need to set the percentage to *75%*. A percentage of *200* would be equal to *32px*. You can also specify size in *ems*, an *em* is equivalent to the width of the letter *m*.

Listing 6.13 shows the use of the *font-size* property.

Listing 6.13: c06text-12.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
body {
font-family: helvetica, sans-serif;
font-size: 12px;

}

h1 {
font-size: 200% }

p {
padding: 10px;
```

```

margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
word-wrap: break-word;
width: 380px;
font-size: 1.3em;

}

</style>
</head>

<body>
<h1>About Apple</h1>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p> </body>

</html>

```

In Listing 6.13, I have set font size for the body element to *12px*. The size for *h1* is *200%* which is equivalent to *24px* because *body* is the containing block for *h1*. The font size of the paragraph is set to *1.3em* which is equivalent to *18px*. Figure 12 shows the effect of this property.

About Apple

Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.

Figure 12

Setting Weight and Style for Font The *font-weight* property allows you to create *bold* text. It takes two values: *normal* and *bold*. The *bold* value causes the text to appear in *bold*. Listing 6.14 shows the use of this property. The effect is shown in Figure 13.

Listing 6.14: c06text-13.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
word-wrap: break-word;
width: 380px;

}

.test1 {
font-weight: bold;

}

</style>
</head>

<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, <span
class="test1">Malus sylvestris</span>, of the rose family.</p> </body>

</html>
```

Apple is usually round, red or yellow, edible fruit of a small tree, **Malus sylvestris**, of the rose family.

Figure 13

The *font-style* property enables you to create *italic* or *oblique* text. It takes three values: *normal*, *italic*, and *oblique*. The *italic* value allows you to make text appear *italic*. The *oblique* value causes text to appear *oblique*. Listing 6.14 shows the use of this property. The effect is shown in Figure 14.

Listing 6.15: c06text-14.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
p {
padding: 10px;
margin: 10px;
border: 3px dotted #4557bc;
background-color: rgba(183, 122, 7, 0.2);
color: darkblue;
word-wrap: break-word;
width: 380px;

}

.test1 {
font-style: italic;

}

</style>
</head>
```

```
<body>
<p>Apple is usually round, red or yellow, edible fruit of a small tree, <span
class="test1">Malus sylvestris</span>, of the rose family.</p> </body>
</html>
```

Apple is usually round, red or yellow, edible fruit of a
small tree, *Malus sylvestris*, of the rose family.

Figure 14

Using Webfonts

The `@font-face` rule enables you to specify a font, even if it is *not installed* in a system from where the user is accessing webpage. It eliminates the need to depend on the number of installed font that users might have installed on their systems.

The `font-family` property specified the name of the font. This name can be used as a value for the `font-family` property in the rest of the document. The `src` property allows you to define the path of the font. Depending on your needs, you can specify multiple paths using this property. The `format` property specifies the format of the font.



Converting Fonts

You can convert a font to multiple formats using the service provided by [Font Squirrel](#). You can access this service by navigating to the following link: <http://www.fontsquirrel.com/tools/webfont-generator>.

In Listing 6.16, I have used the `chase.ttf` font that is inside the `c06fonts` folder using the `src` property. I have first defined the name of the font family and then accessed it using `src`. Then, I have used the `font-family` name that I have defined in the `@font-face` rule code block for the `body` element to be able to use the font.

Font Courtesy: Anthony James
(<https://www.behance.net/gallery/13895473/CHASE-Font>)

[Listing 6.16: c06text-15.html](#)

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
@font-face {
font-family: chase;
src: url("fonts/chase.ttf")
```

```
        }
```

```
body {  
font-family: chase, sans-serif;  
font-size: 200%;
```

```
}
```

```
p {  
padding: 10px;  
margin: 10px;  
border: 3px dotted #4557bc;  
background-color: rgba(183, 122, 7, 0.2);  
color: darkblue;  
word-wrap: break-word;  
width: 380px;
```

```
}
```

```
.test1 {  
font-style: italic;
```

```
}
```

```
</style>  
</head>
```

```
<body>  
<p>Apple is usually round, red or yellow, edible fruit of a small tree, <span  
class="test1">Malus sylvestris</span>, of the rose family.</p> </body>  
</html>
```

Figure 15 shows the font applied to the block of text.

Apple is usually round, red or yellow, edible fruit of a small tree, *Malus sylvestris*, of the rose family.

Figure 15

Using Google Fonts

Google provides hundreds of *free, open-source* fonts optimized for the web that you can use in your projects. To use Google fonts, navigate to www.google.com/fonts and then browse through the fonts. Select font of your choice and then click *Quick-use* button, see Figure 16. From the page that is displayed, select the styles that you want to include and then add the font to page using *Standard*, *@import*, or *JavaScript* method, see Figure 17. Now, add font to the CSS using *font-family* property.



Figure 16



Figure 17

In Listing 6.17, I have incorporated the font using the *@import* directive and then assign this font to the *font-family* property.

Listing 6.17: c06text-16.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
@import url(http://fonts.googleapis.com/css?
family=Yanone+Kaffeesatz:400,700);
body {
font-family: 'Yanone Kaffeesatz', sans-serif;
```

```

        }
    }

    p {
        padding: 10px;
        margin: 10px;
        border: 3px dotted #4557bc;
        background-color: rgba(183, 122, 7, 0.2);
        color: darkblue;
        word-wrap: break-word;
        width: 380px;
        font-size: 120%;
    }

    
```

</style>

</head>

<body>

<p>Apple is usually round, red or yellow, edible fruit of a small tree, Malus sylvestris, of the rose family.</p> </body>

</html>

The effect of the *Yanone Kaffeesatz* font on the text in Figure 18.

Apple is usually round, red or yellow, edible fruit of a small tree,
Malus sylvestris, of the rose family.

Figure 18

Exercises

Exercise 1

Create four paragraphs with *Lorem Ipsum* text apply CSS styling to it (see Figure 19) using the following instructions:

1. Use the template shown in Listings Ex6.1.
2. Create four paragraphs with Lorem Ipsum text and assign the following *classes* in order: *left, right, center, and justify*.
3. Create a rule for the *p* element. Set its width to *450px*. Set border of the block with width *1px, dotted* style, and *blue* color. Finally set *padding* and *top* margin to *5px* and *10px*, respectively.
4. Add the following classes in the style block: *left, right, center, and justify*. Now, set the property to *align* the text to *left, right, center, and justify* in order using these classes.

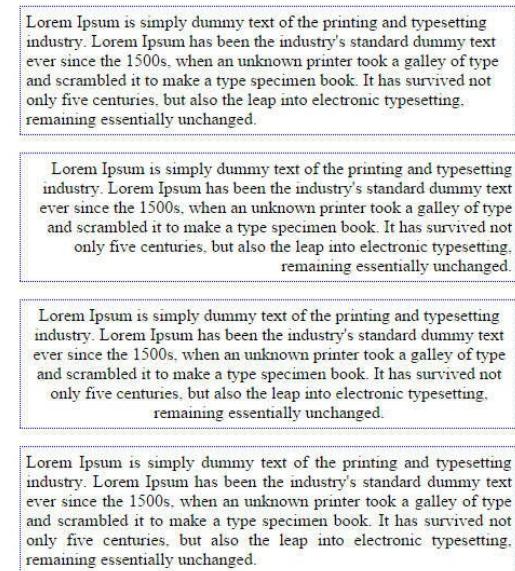


Figure 19

Listing Ex6.1: c06ex1.html

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

```
<head>
```

```
<title> Formatting Text </title>
<style>
</style>
</head>

<body>

</body>

</html>
```

The Lorem Ipsum text to be used:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

Exercise 2

Use the paragraph defined in Listing Ex6.2 and the write a CSS rule to preserve the ***whitespace*** (see Figure 20). The lines should only break at newline characters in the source and at ***
*** elements.

Listing Ex6.2: c06ex2.html

<p>

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.

 Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown
 printer took a galley of type and

scrambled it to make a type specimen book.

</p>

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.

 Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
 when an unknown
 printer took a galley of type and

 scrambled it to make a type specimen book.

Figure 20

Exercise 3

Using the code shown in Listing Ex6.3, create a *blue* colored *drop-shadow* with the *horizontal* and *vertical* offsets set to *0* and *-2px*, respectively (see Figure 21).

Listing Ex6.3: c06ex1.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title> Formatting Text </title>
```

```
  <style>
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p>
```

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum has been the industry's standard dummy text ever since the
 1500s, when an unknown printer took a galley of type and scrambled it to
 make a type specimen book.

```
  </p>
```

```
  </body>
```

```
</html>
```

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Figure 21

Exercise 4

Navigate to the *Google fonts site* and then find the font *Jolly Lodger*. Then, use this font to style the text (Figure 22) contained in the paragraph in Listing Ex6.4. Import the font using the *Standard* method and set the font size to *2em*.

Listing Ex6.4: c06ex4.html

```
<!DOCTYPE html>
<html>

<head>
<title> Formatting Text </title>
<style>
</style>
</head>

<body>
<p>
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book.
</p>
</body>

</html>
```

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Figure 22

Summary

In this chapter, I showed the use of various properties that affects text on your web document and allow you to style the text. You learned about *font* and *text* alignment related properties. You have also learned how to use *custom fonts* and use them effectively. In the end, you learned to use the *Google web fonts*.

7

Styling Lists, Table, Forms, and Cursors

tables

bullet point styles

form elements format

Key Points and Concepts • Styling lists • Styling tables (adding borders and backgrounds) • Styling form elements • Styling Cursor

[Styling Lists](#)

Table 7-1 summarizes the properties that are specific to styling lists.

Table 7.1: The properties specific to style lists

Property

Description	
list-style-type	This property allows you to control the shape or style of a bullet point [also referred to as marker].
list-style-image	It allows you to use an image as a marker.
list-style-position	It controls position of the marker. It takes two values: inside and outside.
list-style	Shorthand for specifying all list characteristics in one line.

You can use the values summarized in Table 7.2, with *list-style-type* property.

Table 7.2: The values of the *list-style-type* property

Property

Description

none	No bullet is drawn.
box check circle diamond disc dash square	The specified shape is used as a marker.

decimal	Use decimal numbers [1 2 3].
decimal-leading-zero	Use decimal numbers with padded zeros [01 02 03].
lower-roman	Use lowercase roman numerals [i ii iii].
upper-roman	Use uppercase roman numerals [I II III].
lower-alpha	Use lowercase alpha characters [a b c].
upper-alpha	Use uppercase alpha characters [A B C].



list-style-type Property

There are many more values that you can use with this property. Visit the following link to know more about these values: <https://developer.mozilla.org/en-US/docs/Web/CSS/list-style-type>.

Listing 7.1 shows the use of these values.

Listing 7.1: c07lists-1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Lists </title> <style>
ol {
list-style-type: lower-roman

}

ol#li1 {
list-style-type: upper-alpha;

}

ul {
list-style-type: square;

}
```

```
</style>
</head>

<body>
<h2> List-1 </h2>
<ol>
<li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ol>
<h2> List-2 </h2>
<ul>
<li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>
<h2> List-3 </h2>
<ol id="li1"> <li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>

</body>

</html>
```

Listing 7.2 shows the use of the *list-style-image* property.

Listing 7.2: c07lists-2.html

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

```
<head>
<title> Lists </title> <style>
ul {
list-style-image: url("images/bullet.png"); }

li {
margin: 10px 0px 0px 0px;

}

</style>
</head>

<body>
<h2> List-1 </h2>
<ul>
<li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>
</body>

</html>
```

The effect of this property is shown in Figure 1. Notice in Listing 7.2 that I have used *margin* property to add *vertical space* between the list items.

List-1

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

List-2

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

Figure 1

You can position the marker using the *list-style-position* property. It controls whether marker should appear on the *inside* or the *outside* of the box. This property takes two values: *inside* and *outside*. The *outside* value causes marker to sit to the left of the block of text whereas the *inside* value causes the marker to sit inside the block of text.

Listing 7.3 shows the use of this property. Figure 2 shows the result.

Listing 7.3: c07lists-3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Lists </title> <style>
ul {
width: 50px;

}
```

```
.one {
list-style-position: inside;
```

```
list-style-image: url(images/bullet.png); }

.two {
list-style-image: url(images/bullet.png); list-style-position: outside;

}

</style>
</head>

<body>
<h2> List-1 </h2>
<ul class="one">
<li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>
<h2> List-2 </h2>
<ul class="two">
<li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>
</body>

</html>
```

List-1

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

List-2

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

Figure 2

You can also use the *list* shorthand to style lists using one declaration.
For example, *list-style: inside circle;*.

Styling Tables

There are many properties that you can use to style the tables. In this section, I will talk about such properties.

Working with Empty Cells The *empty-cells* property allows you to control the behavior of the empty cells. It takes three values: *show*, *hide*, and *inherit*. The *show* property shows border around the cells whereas the *hide* property hides the border. In Listing 7.3, I have left two cells of the table empty. If you run this listing in the browser, you will get the result something like shown in Figure 3.

Listing 7.3: c07empty-cells.html

```
<!DOCTYPE html>
<html>

<head>
<title> Empty Cells </title> <style>
body,
h1 {
font-family: verdana, san-sarif; font-size: 90%;

}

h1 {
width: 168px;
padding: 10px;
background-color: lightgreen;

}

th,
td {
padding: 2px;
```

```
}
```

```
td {  
border: 1px solid #000;
```

```
}
```

```
</style>  
</head>
```

```
<body>
```

```
<h1> [ January Sales Data ] </h1> <table id="sales-table-1"> <tr>  
<th>Salesman</th>  
<th>Appointments</th> <th>Sales</th>  
<th>Expanses</th>  
</tr>  
<tr>  
<td>Phillip</td>  
<td>110</td>  
<td>57</td>  
<td>2500</td>  
</tr>  
<tr>  
<td>Brent</td>  
<td>78</td>  
<td>48</td>  
<td>2000</td>  
</tr>  
<tr>  
<td>Bill</td>  
<td>100</td>  
<td>75</td>  
<td>3700</td>  
</tr>  
<tr>
```

```

<td>Fred</td>
<td>95</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>99</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td></td>
<td></td>
<td>4500</td>
</tr>
</table>

</body>

</html>

```

[January Sales Data]

Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham			4500

Figure 3

Notice that browser has drawn border around the empty cells that is not looking good. Now, lets hide the empty cells. Listing 7.4 shows the code. Figure 4 shows the result.

Listing 7.4: c07empty-cells1.html

```

<!DOCTYPE html>
<html>

```

```
<head>
<title> Empty Cells </title> <style>
body,
h1 {
font-family: verdana, san-sarif; font-size: 90%;
```

```
}
```

```
h1 {
width: 168px;
padding: 10px;
background-color: lightgreen;
```

```
}
```

```
th,
td {
padding: 2px;
```

```
}
```

```
td {
border: 1px solid #000;
```

```
}
```

```
table {
empty-cells: hide;
```

```
}
```

```
</style>
```

```
</head>
<body>

<h1> [ January Sales Data ] </h1> <table id="sales-table-1"> <tr>
<th>Salesman</th>
<th>Appointments</th> <th>Sales</th>
<th>Expanses</th>
</tr>
<tr>
<td>Phillip</td>
<td>110</td>
<td>57</td>
<td>2500</td>
</tr>
<tr>
<td>Brent</td>
<td>78</td>
<td>48</td>
<td>2000</td>
</tr>
<tr>
<td>Bill</td>
<td>100</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>95</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>99</td>
<td>45</td>
<td>6000</td>
```

```

</tr>
<tr>
<td>Graham</td>
<td></td>
<td></td>
<td>4500</td>
</tr>
</table>

</body>

</html>

```

Controlling the Distance between the Adjacent Cells The *border-spacing* property lets you control the distance between the adjacent cells. Notice in Figure 4, browser adds little bit space between the adjacent cells. The *border-spacing* property allows you to control that distance. You can also *collapse* the *adjacent borders* using the *border-collapse* property. It takes two values: *collapse* and *separate*. The *collapse* value collapses the borders to a *single* border. The *separate* value *detaches* borders from each other. In this case, the *border-spacing* and *empty-cells* properties will be considered.

Listing 7.5 shows the use of these properties. Here, I have created two *class selectors* for the two tables. First table uses the *border-spacing* property. The borders of the second table are collapsed using the *border-collapse* property. Figure 5 shows the result of this property.

[January Sales Data]

Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham			4500

Figure 4

Listing 7.5: c07border-distance.html

```
<!DOCTYPE html>
<html>

<head>
<title> Table Properties </title> <style>
body,
h1 {
font-family: verdana, san-sarif; font-size: 90%;

}

h1 {
width: 168px;
padding: 10px;
background-color: lightgreen;

}

th,
td {
padding: 2px;

}

td,
th {
border: 1px solid #000;

}

table.one {
border-spacing: 10px 5px;
```

}

```
table.two {  
border-collapse: collapse;
```

}

```
</style>  
</head>
```

```
<body>
```

```
<h1> [ January Sales Data ] </h1> <table class="one">  
<tr>  
<th>Salesman</th>  
<th>Appointments</th> <th>Sales</th>  
<th>Expanses</th>  
</tr>  
<tr>  
<td>Phillip</td>  
<td>110</td>  
<td>57</td>  
<td>2500</td>  
</tr>  
<tr>  
<td>Brent</td>  
<td>78</td>  
<td>48</td>  
<td>2000</td>  
</tr>  
<tr>  
<td>Bill</td>  
<td>100</td>  
<td>75</td>  
<td>3700</td>  
</tr>
```

```
<tr>
<td>Fred</td>
<td>95</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>99</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td>50</td>
<td>30</td>
<td>4500</td>
</tr>
</table>
```

```
<table class="two">
<tr>
<th>Salesman</th>
<th>Appointments</th> <th>Sales</th>
<th>Expanses</th>
</tr>
<tr>
<td>Phillip</td>
<td>110</td>
<td>57</td>
<td>2500</td>
</tr>
<tr>
<td>Brent</td>
<td>78</td>
<td>48</td>
<td>2000</td>
```

```
</tr>
<tr>
<td>Bill</td>
<td>100</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>95</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>99</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td>50</td>
<td>30</td>
<td>4500</td>
</tr>
</table>
```

```
</body>
```

```
</html>
```

[January Sales Data]			
Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	50	30	4500

Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	50	30	4500

Figure 5

Controlling the Position of the Caption By default, the *caption* is shown at the *top* of the table. If you want to change this position, you can use the *caption-side* property. It takes two values: *top* and *bottom*. The *top* value is the default value. Listing 7.6 shows the use of this property. Figure 6 shows the result.

Listing 7.6: c07caption.html

```
<!DOCTYPE html>
<html>

<head>
<title> Table Properties </title> <style>
body {
font-family: verdana, san-sarif; font-size: 90%;

}

h1 {
width: 168px;
padding: 10px;
background-color: lightgreen;

}


```

```
th,  
td {  
padding: 2px;  
}  
}
```

```
td,  
th {  
border: 1px solid #000;  
}  
}
```

```
table {  
border-collapse: collapse;  
caption-side: bottom;  
}  
}
```

```
caption {  
margin: 10px;  
}  
}
```

```
</style>  
</head>  
  
<body>  
<table class="one">  
<caption> [ January Sales Data ]</caption> <tr>  
<th>Salesman</th>  
<th>Appointments</th> <th>Sales</th>  
<th>Expenses</th>  
</tr>  
<tr>
```

Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	50	30	4500

```
</body>
```

```
</html>
```

Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	50	30	4500

[January Sales Data]

Figure 6

Working with the Layout of the Table By default, browser sets the width of the table automatically. You can disable this behavior by using the *table-layout* property and using the value *fixed* for it. When you use this value, the size of the table is set by the width values for the table and individual columns. In Listing 7.7, I have set *width* to *100%* that causes table to occupy the available space. I have also set the layout to *fixed*.

[Listing 7.7: c07fixed-layout.html](#)

```
<!DOCTYPE html>
<html>

<head>
<title> Table Properties </title> <style>
body {
font-family: verdana, san-sarif; font-size: 90%;

}

h1 {
width: 168px;
padding: 10px;
```

```
background-color: lightgreen;
```

```
}
```

```
th,  
td {  
padding: 2px;
```

```
}
```

```
td,  
th {  
border: 1px solid #000;
```

```
}
```

```
table {  
border-collapse: collapse;  
caption-side: bottom;  
table-layout: fixed;  
width: 100%;
```

```
}
```

```
caption {  
margin: 10px;
```

```
}
```

```
</style>  
</head>
```

```
<body>
```

```
<table class="one">
<caption> [ January Sales Data ]</caption> <tr>
<th>Salesman</th>
<th>Appointments</th> <th>Sales</th>
<th>Expanses</th>
</tr>
<tr>
<td>Phillip</td>
<td>110</td>
<td>57</td>
<td>2500</td>
</tr>
<tr>
<td>Brent</td>
<td>78</td>
<td>48</td>
<td>2000</td>
</tr>
<tr>
<td>Bill</td>
<td>100</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>95</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>99</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
```

```
<td>Graham</td>
<td>50</td>
<td>30</td>
<td>4500</td>
</tr>
</table>
```

```
</body>
</html>
```

Styling the Cursor The *cursor* property allows you to specify the type of cursor to be displayed when pointing on an element. It takes many values such as *crosshair*, *help*, *progress*, *wait*, and so on. Listing 7.8 shows the use of this property. To show the cursor style, hover the cursor on the paragraphs.

Listing 7.8: c07cursor.html

```
<!DOCTYPE html>
<html>

<head>
<title> Cursor </title> <style>
p.crosshair {
cursor: crosshair;

}

p.progress {
cursor: progress;

}

p.help {
```

```
cursor: help;  
}  
  
</style>  
</head>  
  
<body>  
<p class="crosshair">Apple is the round fruit of a tree of the rose family,  
which typically has thin green or red skin and crisp flesh.</p>  
<p class="progress">Apple is the round fruit of a tree of the rose family,  
which typically has thin green or red skin and crisp flesh.</p>  
<p class="help">Apple is the round fruit of a tree of the rose family, which  
typically has thin green or red skin and crisp flesh.</p>  
</body>  
  
</html>
```

Exercises

Exercise 1

Using the template shown in Listing Ex7.1. Position the marker inside in *List1* and outside in *List2*. Generate the output shown in Figure 7.

This is list1

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

This is list2

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

Figure 7

Listing Ex7.1: c07ex1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Lists </title> <style>
</style>
</head>

<body>
<ul id="id1">
This is list1
<li> Item 1</li>
<li> Item 2</li>
<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>

<ul id="id2">
This is list2
<li> Item 1</li>
<li> Item 2</li>
```

```

<li> Item 3</li>
<li> Item 4</li>
<li> Item 5</li>
</ul>
</body>

</html>

```

Exercise 2

Create a table using the values shown in Figure 8 [You can also use the Listing Ex7.2]. Then, format the table, see Figure 8. Below are some hints:

1. Width of the table is *600 px*.
2. *Letter spacing* is set to *0.5em* and text *transform* is set to *uppercase* for the *caption* element. It also uses *10px* margin.
3. A *padding* of *2px* is used for the *th* and *td* elements.
4. Use colors as per your choice.
5. Change *background color* of the *cell* to *#895050* when you hover mouse over a cell.
6. Align text in the *Expenses* column to the right.

[J A N U A R Y S A L E S D A T A]			
Salesman	Appointments	Sales	Expenses
Phillip	110	57	2500
Brent	78	48	2000
Bill	100	75	3700
Fred	95	56	4000
Tim	99	45	6000
Graham	32	68	4500

Figure 8

Listing Ex7.2: c07ex2.html

```

<!DOCTYPE html>
<html>

<head>
<title>January Sales Data</title> <style>
body {
font-family: verdana, san-sarif; font-size: 90%;

```

}

```
</style>
</head>

<body>
<table>
<caption> [ January Sales Data ] </caption> <tr>
<th>Salesman</th>
<th>Appointments</th> <th>Sales</th>
<th>Expanses</th>
</tr>
<tr>
<td>Phillip</td>
<td>110</td>
<td>57</td>
<td>2500</td>
</tr>
<tr>
<td>Brent</td>
<td>78</td>
<td>48</td>
<td>2000</td>
</tr>
<tr>
<td>Bill</td>
<td>100</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>95</td>
<td>56</td>
<td>4000</td>
```

```
</tr>
<tr>
<td>Tim</td>
<td>99</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td>32</td>
<td>68</td>
<td>4500</td>
</tr>
</table>

</body>

</html>
```

Summary

In this chapter, I have described the properties that you can use to style lists. You have also seen the use of the properties that affects the appearance of the table. I also showed the use of cursor property to style the appearance of the cursor.

8

Creating Layout

Chapter 8: Creating Layout

In this lesson, I will show you how to control the position of the element on a page and how to create different types of *layouts* using CSS including the *multicolumn* layout. I will also show you how to adjust the *position* of the elements using the *offset* properties.

Key Points and Concepts • Normal positioning • Relative positioning • Absolute positioning • Fixed positioning • Drawing order of the elements • Multicolumn layout

Positioning Content You can *position content* of on a page using the *positioning* properties. Table 8.1 summarizes the properties that you can use. Table 8.2 shows the values that *position* property takes.

Table 8.1: The <i>position</i> properties	
Property	Description
position	Allows to set the positioning method.
left right top bottom	Sets the offset values for positioning elements.
z-index	Set the ordering of the overlapping elements.

Table 8.2: The <i>position</i> property values	
Property	Description
static	The default value. The element is laid out as normal.
relative	You can use this property to position element relative to its normal position.
absolute	The element is positioned relative to its ancestor.
fixed	The element is positioned relative to the browser window.

Normal Positioning This is the default way in which browser places one block top of another block. Since it is the *default behavior* therefore you don't need a *CSS* property. However, the *syntax* for

declaring *normal* flow is as follows:

```
position: static;
```

Listing 8.1 shows the use of the static value.

Listing 8.1: c08normal-flow.html

```
<!DOCTYPE html>
<html>

<head>
<title> Positioning Content </title> <style>
body {
width: 750px;
font-family: helveticaA, verdana, sans-serif; }

h1 {
background-color: #a3a3a3; padding: 5px;

}

p {
width: 500px;

}

</style>
</head>

<body>
<h1>What is Lorem Ipsum</h1> <p class="class1">
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
```

the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

</p>

<p class="class2">

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

</p>

<p class="class3">

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

</p>

</body>

</html>

Figure 1 shows the *normal* flow. Notice that paragraphs are appearing below heading in order. The *width* of the paragraph elements is *500px*. I have not provided *width* for the *h1* element that causes it to stretch to the full width [*750px*] specified using the *body* rule.

What is Lorem Ipsum

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

Figure 1

Relative Positioning The *relative* value of the *position* property allows you to *offset* the element relative to its *normal* position. You need to use the *offset* properties [*top*, *bottom*, *left*, or *right*] to place element relative to its *normal* position. You can use *pixels*, *percentages*, or *ems* to assign values.

Listing 8.2 shows the use of this property.

Listing 8.2: c08relative.html

```
<!DOCTYPE html>
<html>

<head>
<title> Positioning Content </title> <style>
body {
width: 750px;
font-family: helvetica, verdana, sans-serif; }

h1 {
background-color: #a3a3a3; padding: 5px;
```

}

```
p {  
width: 500px;
```

}

```
.class2 {  
position: relative;  
top: 15px;  
left: 150px;  
margin-bottom: 50px;
```

}

```
</style>  
</head>
```

```
<body>  
<h1>What is Lorem Ipsum</h1> <p class="class1">  
Lorem Ipsum is simply dummy text of the printing and typesetting industry.  
Lorem Ipsum has been the industry's standard dummy text ever since the  
1500s, when an unknown printer took a galley of type and scrambled it to  
make a type specimen book. It has survived not only five centuries, but also  
the leap into electronic typesetting, remaining essentially unchanged. It was  
popularised in the 1960s with the release of Letraset sheets containing Lorem  
Ipsum passages, and more recently with desktop publishing software like  
Aldus PageMaker including versions of Lorem Ipsum.
```

```
</p>  
<p class="class2">
```

```
Contrary to popular belief, Lorem Ipsum is not simply random text. It has  
roots in a piece of classical Latin literature from 45 BC, making it over 2000  
years old. Richard McClintock, a Latin professor at Hampden-Sydney College  
in Virginia, looked up one of the more obscure Latin words, consectetur, from  
a Lorem Ipsum passage, and going through the cites of the word in classical
```

literature, discovered the undoubtable source.

</p>

<p class="class3">

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

</p>

</body>

</html>

Figure 2 shows the result of the Listing 8.2. Notice the change in the *position* of the second paragraph. It moved *150px* from the *left* edge and *15px* from the *top* edge from the position where it would have been in the normal flow.

What is Lorem Ipsum

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

Figure 2

Absolute Positioning When you use the *absolute* value, the element

will be taken out of the flow and it no longer affects the position of the other elements on the page. The *offset* properties let you place the *box* relative to the browser. In the Listing 8.3, the second paragraph is positioned *15px* away from the *top* edge and *150px* from the *left* edge of the browser. Figure 3 shows the result. I have set a *background color* for the second paragraph so that you can clearly see its position on the page.

Listing 8.3: c08absolute.html

```
<!DOCTYPE html>
<html>

<head>
<title> Positioning Content </title> <style>
body {
width: 750px;
font-family: helvetica, verdana, sans-serif; }

h1 {
background-color: #a3a3a3; padding: 5px;

}

p {
width: 500px;

}

.class2 {
position: absolute;
background-color: antiquewhite; top: 15px;
left: 150px;
```

}

</style>

</head>

<body>

<h1>What is Lorem Ipsum</h1> <p class="class1">

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

</p>

<p class="class2">

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

</p>

<p class="class3">

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

</p>

</body>

</html>

What is

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at

Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and ever since the 1500s going through the cites of the word in classical literature, discovered and scrambled it to the undoubtable source.

only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

Figure 3

Fixed Positioning The *fixed* value of the *position* property can be used to *position* element *relative* to the browser. The position of the elements is fixed therefore when a user scrolls the page, the elements stays in its place. The placement of the element is controlled by the *offset* properties. Listing 8.4 shows use of this value.

Listing 8.4: c08fixed.html

```
<!DOCTYPE html>
<html>

<head>
<title> Positioning Content </title> <style>
body {
width: 750px;
font-family: helvetica, verdana, sans-serif; }

h1 {
background-color: #a3a3a3; padding: 5px;
position: fixed;
margin: 0;
top: 0px;
left: 400px;
```

```
        }
```

```
p {  
width: 500px;
```

```
}
```

```
</style>  
</head>
```

```
<body>  
<h1>What is Lorem Ipsum</h1> <p class="class1">  
Lorem Ipsum is simply dummy text of the printing and typesetting industry.  
Lorem Ipsum has been the industry's standard dummy text ever since the  
1500s, when an unknown printer took a galley of type and scrambled it to  
make a type specimen book. It has survived not only five centuries, but also  
the leap into electronic typesetting, remaining essentially unchanged. It was  
popularised in the 1960s with the release of Letraset sheets containing Lorem  
Ipsum passages, and more recently with desktop publishing software like  
Aldus PageMaker including versions of Lorem Ipsum.
```

```
</p>
```

```
<p class="class2">
```

```
Contrary to popular belief, Lorem Ipsum is not simply random text. It has  
roots in a piece of classical Latin literature from 45 BC, making it over 2000  
years old. Richard McClintock, a Latin professor at Hampden-Sydney College  
in Virginia, looked up one of the more obscure Latin words, consectetur, from  
a Lorem Ipsum passage, and going through the cites of the word in classical  
literature, discovered the undoubtable source.
```

```
</p>
```

```
<p class="class3">
```

```
Contrary to popular belief, Lorem Ipsum is not simply random text. It has  
roots in a piece of classical Latin literature from 45 BC, making it over 2000  
years old. Richard McClintock, a Latin professor at Hampden-Sydney College  
in Virginia, looked up one of the more obscure Latin words, consectetur, from  
a Lorem Ipsum passage, and going through the cites of the word in classical  
literature, discovered the undoubtable source.
```

```
</p>
</body>

</html>
```

Notice in Figure 4, the heading stays in its place when page is *scrolled* and the paragraph content is obscured by the heading box. The paragraph are in the normal flow and therefore ignore the heading element.

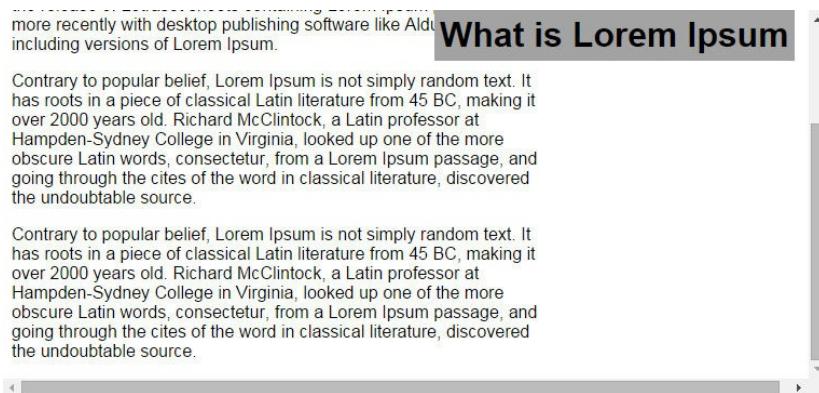


Figure 4

Setting the Z-Order The *z-index* property allows you to set the *front-to-back* order of the element. It takes a number as value. You can also use the *negative* numbers. The smaller the value, further to the back the element will be placed. Listing 8.5 shows use of this property.

Listing 8.5: c08z-order.html

```
<!DOCTYPE html>
<html>

<head>
<title> Overlapping Elements </title> <style>
.clr {
background-color: rgb(255, 0, 0); position: absolute;
left: 28px;
top: 32px;
width: 134px;
```

```
height: 113px;  
z-index: 1;
```

```
}
```

```
.clr1 {  
background-color: rgb(255, 240, 0); position: absolute;  
left: 40px;  
top: 60px;  
width: 128px;  
height: 87px;  
z-index: 2;
```

```
}
```

```
</style>  
</head>
```

```
<body>  
<div class="clr">  
<div class="clr1">
```

```
</div>  
</div>
```

```
</body>
```

```
</html>
```

Notice in Figure 5, the *red* square is placed behind the *yellow* square because the *red* square's *z-index* value is *smaller* than that of the *yellow* square.

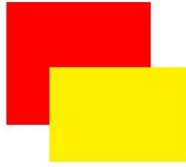


Figure 5

Creating Multicolumn Layout The CSS *multicolumn* layout allows you to define multiple column of text as you see in newspapers. Table 8.3 summarizes the multicolumn properties.

Table 8.3: The *multicolumn* properties

Property	Description
column-count	Specifies the number of columns.
column-gap	Controls the distance between columns.
column-rule-color	Sets color of the rule between columns.
column-rule-style	Sets the style of the rules.
column-rule-width	Sets the width of the rules.
column-rule	Shorthand property of setting column rules.
column-width	Specifies width of the columns.

Listing 8.6 creates a three column layout. Figure 6 shows the result.

Listing 8.6: c08three-columns.html

```
<!DOCTYPE html>
<html>

<head>
<title> Creating Layout </title> <style>
body {
width: 750px;
font-family: helvetica, verdana, sans-serif; }

.news {
```

```
-webkit-column-count: 3;  
/* Chrome, Safari, Opera */
```

```
-moz-column-count: 3;  
/* Firefox */
```

```
-webkit-column-rule: 1px solid blue; -webkit-column-gap: 18px;  
/* Chrome, Safari, Opera */
```

```
-moz-column-rule: 1px solid blue; -moz-column-gap: 18px;  
/* Firefox */
```

```
column-count: 3;  
column-gap: 18px;  
column-rule: 1px solid blue; }  
</style>  
</head>
```

```
<body>  
<h1>What is Lorem Ipsum</h1> <p class="news">  
Lorem Ipsum is simply dummy text of the printing and typesetting industry.  
Lorem Ipsum has been the industry's standard dummy text ever since the  
1500s, when an unknown printer took a galley of type and scrambled it to  
make a type specimen book. It has survived not only five centuries, but also  
the leap into electronic typesetting, remaining essentially unchanged. It was  
popularised in the 1960s with the release of Letraset sheets containing Lorem  
Ipsum passages, and more recently with desktop publishing software like  
Aldus PageMaker including versions of Lorem Ipsum. Contrary to popular  
belief, Lorem Ipsum is not simply random text. It has roots in a piece of  
classical Latin literature from 45 BC, making it over 2000 years old. Richard  
McClintock, a Latin professor at Hampden-Sydney College in Virginia,  
looked up one of the more obscure Latin words, consectetur, from a Lorem  
Ipsum passage, and going through the cites of the word in classical literature,
```

discovered the undoubtable source. Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

```
</p>
</body>
```

```
</html>
```

What is Lorem Ipsum

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more

recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical

literature, discovered the undoubtable source. Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source.

Figure 6

Exercises Exercise 1

Using the Listing Ex8.1, create a *three column* newspaper style layout. Set the *column gap* of *40* pixels and set *rule color* to *red* and make it *2px* wide.

Listing 8.1: c08ex1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Creating Layout </title> <style>
</style>
</head>
```

```
<body>
<h1>What is Lorem Ipsum</h1> <p class="news">
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It was
popularised in the 1960s with the release of Letraset sheets containing Lorem
Ipsum passages, and more recently with desktop publishing software like
Aldus PageMaker including versions of Lorem Ipsum. Contrary to popular
belief, Lorem Ipsum is not simply random text. It has roots in a piece of
classical Latin literature from 45 BC, making it over 2000 years old. Richard
McClintock, a Latin professor at Hampden-Sydney College in Virginia,
looked up one of the more obscure Latin words, consectetur, from a Lorem
Ipsum passage, and going through the cites of the word in classical literature,
discovered the undoubtable source. Contrary to popular belief, Lorem Ipsum is
not simply random text. It has roots in a piece of classical Latin literature from
45 BC, making it over 2000 years old. Richard McClintock, a Latin professor
at Hampden-Sydney College in Virginia, looked up one of the more obscure
Latin words, consectetur, from a Lorem Ipsum passage, and going through the
cites of the word in classical literature, discovered the undoubtable source.
</p>
</body>

</html>
```

Exercise 2

Create three squares, refer Figure 7, and then use the ***z-index*** property to decide their ***front-to-back*** order. The size of the boxes is ***250x250px***. Following are the values of the ***top*** and ***left*** offset properties.

1. Red Square: 39px, 43px
2. Green Square: 102px, 117px
3. Blue Square: 164px, 190px

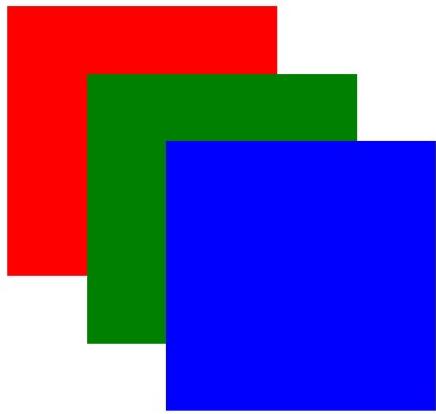


Figure 7

Summary

In this chapter, you explored different ways to position elements using *normal*, *relative*, and *absolute* positioning. You have also learned the use of the *offset* properties to place the content relative to the containing box and browser.

9

Animation, Transformations, and Tran- sitions

animation

interactivity

JavaScript

JavaScript

Key Points and Concepts • Creating Transitions • Creating Animations • Using Transforms

Creating Transitions When you apply a CSS property to an element, browser immediately applies it to the element. However, you can create a transition to gradually apply that effect to the element. Table 9.1 shows the properties that you can use to control the transitions.

Table 9.1: The *transition* properties

Property

Description	
transition-delay	Controls the delay after which the transition will start.
transition-duration	Controls the time of the transition.
transition-property	Specifies property that the transition applies to. You can specify multiple properties.
transition-timing-function	Controls the speed curve of the transition effect. It allows you to change speed over time.

transition	This is a shorthand property for declaring all four properties in one sentence.
------------	---



transition-duration Property

Always apply the *transition-duration* property because the *default* value of this property is *zero*. As a result, no *transition* is applied.

In Listing 9.1, I have set a *100x100px* box and set its *background color* to *red*. The properties that I want to apply transition to are *width* and *background-color*. When user hover over the box, I want to change width of the box to *300px* and also want to change its *background color* to *blue*. I have set *transition delay* to *2* seconds and I want transition to last *5* seconds that I am controlling through *transition-duration* property.

Listing 9.1: c09transition-delay.html

```
<!DOCTYPE html>
<html>

<head>
<title> Animation </title> <style>
div {
width: 100px;
height: 100px;
background-color: red;
transition-property: width, background-color; transition-duration: 5s;
transition-delay: 2s;
transition-timing-function: linear; }

div:hover {
width: 300px;
background-color: blue;
}
```

```
</style>
</head>

<body>
<div></div>
</body>

</html>
```

Render the listing in the browser and hover the cursor on the *red* square, the transition should start after *2* seconds. Notice in Listing 9.1, I have set *speed change curve* to *linear*. It sets the transition effect with the same speed from stat to end. Table 9.2 describes various values.

Table 9.2: The *transition-timing-function* values

Value

Description

ease	This is the default value. It causes transition to start slow, then fast, and then again end slowly.
------	--

linear	Same speed from start to end.
ease-in	Starts slow.
ease-out	Ends slow.
ease-in-out	Slow start and end.
cubic-bezier(n, n, n, n)	You can use it to define your own values.

Creating Animations Animations are like *transitions* however they offer more control and more flexibility. Table 9.3 summarizes the *animation* properties.

Table 9.3: The *animation* properties

Property

Description	
animation-name	Sets the name of the keyframe you want to bind to a selector.
animation-duration	Sets the total time of the animation.
animation-timing-function	Controls the speed curve of the animation.
animation-delay	Controls the time before the animation start.
animation-iteration-count	Controls how many times an animation is played.
animation-direction	Controls whether the animation should be played backward on alternate cycles.
animation-fill-mode	Controls the behavior when animation is not playing. For example, when it is finished or when it has a delay).
animation-play-state	Controls whether animation is playing or stopped.

Before we work on the *animation* properties, let's first understand the *@keyframes* rule. It allows you to gradually switch from one set of CSS styles to another set of CSS styles. The style change happens in percentage where *0%* is the start of the animation and *100%* is the end of the animation. You can also use the *from* and *to* keywords where from represents *0%* and to represents *100%*.

Listing 9.2 shows the use of the *animation* properties. When you run the Listing 9.2 in the browser, the *red* box animates *infinitely* because I have set the *animation-iteration-count* property to *infinite*. The positioning of the box is controlled by the relative value of the *position* property.

Listing 9.2: c09animation-1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Animation </title> <style>
div {
```

```
width: 100px;  
height: 100px;  
background-color: red;  
position: relative;  
-webkit-animation-duration: 5s;  
-webkit-animation-iteration-count: infinite; -webkit-animation-timing-  
function: ease-in-out; -webkit-animation-name: moveBox;
```

```
}
```

```
@-webkit-keyframes moveBox {  
0% {  
top: 0px;
```

```
}
```

```
25% {  
top: 200px;
```

```
}
```

```
75% {  
top: 50px;
```

```
}
```

```
100% {  
top: 100px;
```

```
}
```

```
}
```

```
</style>
</head>

<body>
<div></div>
</body>

</html>
```

You can also change as many styles as you want in an animation. Listing 9.3 shows an example. Run the listing and notice that the *background color* and the *width* are animated.

Listing 9.3: c09animation-2.html

```
<!DOCTYPE html>
<html>

<head>
<title> Animation </title> <style>
div {
width: 100px;
height: 100px;
background-color: red;
position: relative;
-webkit-animation-duration: 5s;
-webkit-animation-iteration-count: infinite; -webkit-animation-timing-
function: ease-in-out; -webkit-animation-name: moveBox;

}

@-webkit-keyframes moveBox {
0% {
top: 0px;
background-color: aqua;
width: 50px;
```

```

        }
    100% {
        top: 100px;
        background-color: yellow;
        width: 100px;
    }

}

}

</style>
</head>

<body>
<div></div>
</body>

</html>

```

Listing 9.4 shows the Listing 9.3 with the *form* and *to* keywords.

Listing 9.4: c09animation-3.html

```

<!DOCTYPE html>
<html>

<head>
<title> Animation </title> <style>
div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    -webkit-animation-duration: 5s;
    -webkit-animation-iteration-count: infinite;
    -webkit-animation-timing-

```

```
function: ease-in-out; -webkit-animation-name: moveBox;  
}  
  
@-webkit-keyframes moveBox {  
from {  
top: 0px;  
background-color: greenyellow;  
width: 50px;  
  
}  
  
to {  
top: 100px;  
background-color: lawngreen;  
width: 100px;  
  
}  
  
}  
  
</style>  
</head>  
  
<body>  
<div></div>  
</body>  
  
</html>
```

Using Transforms

You can apply *2D* and *3D* transformations to an element using the *Transform* property. This property lets you *rotate*, *scale*, *move*, *skew*, and so on. The *transform* property specifies the *transform* function to apply. The *transform-origin* property specifies the *origin* or *pivot* of the transform.

Table 9.4 describes the *transform* property values.

Table 9.3: The *transform* properties

Value

Description

none	No transformation.
matrix(n,n,n,n,n,n)	Defines the 2D transforms. A matrix of six values is used.
matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	Defines the 3D transforms, A matrix of sixteen values is used.
translate(x,y)	Defines the 2D transform.
translate3d(x,y,z)	Defines the 3D transform.
translateX(x) translateY(y) translateZ(z)	Defines a transform using the value for x, y, or z axis, respectively.
scale(x,y)	Defines a 2D scale transformation.
scale3d(x,y,z)	Defines a 3D scale transformation.
scaleX(x) scaleY(y) scaleZ(z)	Defines a scale transform using the value for x, y, or z axis, respectively.
rotate(angle)	Defines a 2D rotation. Accepts angle as parameter.
rotate3d(x, y, z, angle)	Defines a 3D rotation.

rotate X

rotate Y

rotate Z	Defines a rotation along the x, y, or z axis, respectively.
skew(x-angle, y-angle)	Defines a 2D skew transformation along the x and y axes.
skewX(angle) skewY(angle)	Defines a 2D skew transformation along the x or y axis, respectively.
perspective(n)	Defines a perspective view for a 3D element.

Let's start using the *transform* properties. The *rotate* value allows you to rotate an element by an *angle* expressed in degrees. The *scale* property lets you scale an element along the *x* and *y* axes. Listing 9.5 shows the use of these two properties.

Listing 9.5: c09transform-1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Transform </title> <style>
div {
float: left;

}

div.original {
margin: 60px;
width: 100px;
height: 100px;
background-color: red;

}

div.transformed {
margin: 60px;
width: 100px;
```

```

height: 100px;
background-color: red;
transform: rotate(10deg) scale(1.5); }
</style>
</head>

<body>
<div class="original">Rotate by 10 degrees, scale by 1.5.</div> <div
class="transformed">Rotated 10 degrees and scaled by 1.5.</div> </body>

</html>

```

Notice that the original *div* element is *100px* wide and *100px* tall. I have used the *rotate* and *scale* values to *transform* the element. Notice the effect of *transform* show in Figure 1.



Figure 1

The *transform-origin* value lets you specify the *pivot point* or the *origin* of the transformation. Listing 9.6 shows use of this value. In this example, I am trying to rotate the cube around the *top-right* corner of the box. The effect is shown in Figure 2.

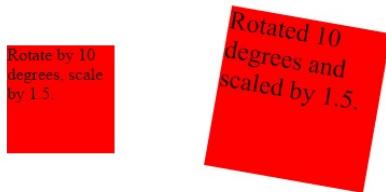


Figure 2

Listing 9.6: c09transform-2.html

```
<!DOCTYPE html>
<html>

<head>
<title> Transform </title> <style>
div {
float: left;

}

div.original {
margin: 60px;
width: 100px;
height: 100px;
background-color: red;

}

div.transformed {
margin: 60px;
width: 100px;
height: 100px;
background-color: red;
transform: rotate(30deg);
transform-origin: right top;

}

</style>
</head>

<body>
<div class="original">Rotate by 30 degrees !</div> <div
class="transformed">Rotated 30 degrees. The pivot is right top corner of the
box!</div> </body>
```

</html>

The *transform-origin* property uses the following syntax:

transform-origin: x-axis y-axis z-axis|initial|inherit;

Table 9.4 describes the *transform-origin* properties values.

Table 9.4: The *transform-origin* properties

Value

Description	
x-axis	Defines the X coordinate. Possible values are: left, center, right, length, and %.
y-axis	Defines the Y coordinate. Possible values are: top, center, bottom, length, and %.
z-axis	Defines the z axis location. The possible value is length.

Exercises

Exercise -1

Use the Listing Ex9.1. Assign six values shown in Table 9.2 to the six **div** boxes of the Listing Ex9.1 and then see the effect in the browser.

Listing Ex9.1: c09ex1.html

```
<!DOCTYPE html>
<html>

<head>
<title> Animation </title> <style>
div {
width: 100px;
height: 30px;
background-color: beige;
transition-property: width, background-color; transition-duration: 1s;
transition-delay: 1s;
transition-timing-function: linear; }

div:hover {
width: 300px;
background-color: bisque;
}

</style>
</head>

<body>
<div class="one">Linear</div>   <div class="two">Ease</div>   <div
class="three">Ease-In</div>   <div class="four">Ease-Out</div>   <div
class="five">Ease-In-Out</div>   <div class="six">Cubic-Bezier</div>
</body>

</html>
```

Exercise 2

Use Listing 9.3. Make changes to the `div` rule in CSS to ensure that the animation is played *two* times. Also, set the direction to alternate reverse.

Exercise 3

Use Listing Ex9.2. Rotate the box by *360* degrees when the user hover mouse over the box. Also, change the *background color* to *lightcoral* and add a border of *10px* border with *lightgreen* color and *ridge* style.

Listing Ex9.2: c09ex3.html

```
<!DOCTYPE html>
<html>

<head>
<title> Transform </title> <style>
div {
float: left;

}

div.original {
margin: 60px;
width: 100px;
height: 100px;
background-color: red;

}

</style>
</head>

<body>
```

```
<div class="original">Rotate it by 360 degrees !</div>
</body>
```

```
</html>
```

Summary

As you have seen, CSS gives absolute freedom to create animations. You have full control over the animation, transitions, and transformations. In this chapter, I demonstrated use of animation and transition properties.

Tutorials

Tutorial - 1: Styling a Simple Table In this tutorial, you will create and style a simple table (see Figure 1).

Salesman	Appointments	Sales	Expenses
Phillip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Chicago	50	4500

Figure 1

Follow these steps:

1. Open the code editor you are using and then create a new HTML document. Save the document with the name *tut1.html*.

2. Type the following code and then save the html file:

```
<!DOCTYPE html>
<html>

<head>
<title> Styling a Simple Table </title>
<style>
</style>
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

3. Add the following code below the *body* tag to create the table:

```
<table class="simpleTable">
<tr>
<th>Salesman</th>
<th>Appointments</th>
<th>Sales</th>
<th>Expanses</th>
</tr>
<tr>
<td>Phillip</td>
<td>NewYork</td>
<td>57</td>
<td>2500</td>
</tr>
<tr>
<td>Brent</td>
<td>Miami</td>
<td>48</td>
<td>2000</td>
</tr>
<tr>
<td>Bill</td>
<td>Washington</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>Los Angeles</td>
<td>56</td>
<td>4000</td>
```

```
</tr>
<tr>
<td>Tim</td>
<td>Utah</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td>Shicago</td>
<td>50</td>
<td>4500</td>
</tr>
</table>
```

The *simpleTable* class will be used to style the table. Let's start styling the table.

4. Add the following rule to the *style* element to set the *font family*:

```
body {
  font-family: helvetica, verdana, sans-serif; }
```

5. Add the rule to the *style* element for the *simpleTable* class:

```
.simpleTable {
  font-size: 12px;
  color: #3c3c3c;
  width: 100%;
  border-width: 1px;
  border-color: #b9a986; }
```

6. Add the following rule to style the *th* element:

```
.simpleTable th {
  font-size: 13px;
  background-color: #d4c4a0;
```

```
border-width: 1px;  
padding: 6px;  
border-style: solid;  
border-color: #cbba95;  
  
}
```

Figure 2 shows the output. Notice that the text in the first row is center aligned. Next, you will change the *alignment*.

Salesman	Appointments	Sales	Expenses
Phillip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Chicago	50	4500

Figure 2

7. In the *.simpleTable th* rule, add the following declarations to *align* the text:

```
text-align: left;  
vertical-align: middle;
```

8. Add the following rule to the *style* element to assign a background color to the rows:

```
.simpleTable tr {  
background-color: #e5cf9e;
```

```
}
```

Figure 3 shows the result. Notice that there is space between the cells. Next, you will fix it.

Salesman	Appointments	Sales	Expenses
Philip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Chicago	50	4500

Figure 3

- 9.** In the *.simpleTable* rule, add the following declaration to remove the spacing:
 border-collapse: collapse;

Next, its time to style the cells using the *td* element.

- 10.** Add the following rule to the *style* tag:

```
.simpleTable td {
  font-size: 12px;
  border-width: 1px;
  padding: 8px;
  border-style: solid;
  border-color: #af9f7a;
```

}

Finally, you will write a rule for changing the color of a row to white when user hovers mouse over it.

- 11.** Add the following rule to the *style* tag:

```
.simpleTable tr:hover {
  background-color: aliceblue;
```

}

Below is the complete code that you have created in this tutorial:

[File:/tutorials/tutorial-1.html](#)

```
<!DOCTYPE html>
<html>

<head>
<title> Styling a Simple Table </title> <style>
body {
font-family: helvetica, verdana, sans-serif; }

.simpleTable {
font-size: 12px;
color: #3c3c3c;
width: 100%;
border-width: 1px;
border-color: #b9a986;
border-collapse: collapse;

}

.simpleTable th {
font-size: 13px;
background-color: #d4c4a0;
border-width: 1px;
padding: 6px;
border-style: solid;
border-color: #cbba95;
text-align: left;
vertical-align: middle;

}

.simpleTable tr {
```

```
background-color: #e5cf9e;
```

```
}
```

```
.simpleTable td {  
font-size: 12px;  
border-width: 1px;  
padding: 8px;  
border-style: solid;  
border-color: #af9f7a;
```

```
}
```

```
.simpleTable tr:hover {  
background-color: aliceblue;
```

```
}
```

```
</style>  
</head>
```

```
<body>  
<table class="simpleTable">  
<tr>  
<th>Salesman</th>  
<th>Appointments</th>  
<th>Sales</th>  
<th>Expanses</th>  
</tr>  
<tr>  
<td>Phillip</td>  
<td>NewYork</td>  
<td>57</td>  
<td>2500</td>  
</tr>
```

```
<tr>
<td>Brent</td>
<td>Miami</td>
<td>48</td>
<td>2000</td>
</tr>
<tr>
<td>Bill</td>
<td>Washington</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>Los Angeles</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>Utah</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td>Shicago</td>
<td>50</td>
<td>4500</td>
</tr>
</table>

</body>

</html>
```

Tutorial - 2: Styling a table with rounded corner In this tutorial, you

will style a table and make its corners round (see Figure 1).

Salesman	Appointments	Sales	Expenses
Phillip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Chicago	50	4500

Figure 1

Follow these steps:

1. Open the code editor you are using and then create a new HTML document. Save the document with the name *tut2.html*.

2. Type the following code and then save the html file:

```
<!DOCTYPE html>
<html>

<head>
<title> Styling a Table </title> <style>
</style>
</head>

<body>

</body>

</html>
```

3. Copy the code that you have used to create the table in Tutorial 1 and paste immediately below the opening *body* tag. Assign class, *grid* to the table: *<table class="grid">*

4. Add the following rule to the *style* tag to specify the font family:

```
body {  
    font-family: helvetica, verdana, sans-serif; }
```

5. Add the following rule using the *table* selector to define table width and to *remove spacing* between cells:

```
table {  
    border-collapse: separate;  
    border-spacing: 0;  
    width: 100%;  
  
}
```

6. Add the style code for the *grid* class:

```
.grid {  
    border: solid #c3c0c0 1px;  
    border-radius: 10px;  
    box-shadow: 0 1px 1px #ccc;  
  
}
```

7. Add the following rules to the *style* tag to create *1px* border. Also, align *text* to the left:

```
.grid td,  
.grid th {  
    border-left: 1px solid #ccc;  
    border-top: 1px solid #ccc;  
    padding: 10px;  
    text-align: left;  
  
}
```

Figure 2 shows the output. Next, you will style the table header using ***th*** element.

Salesman	Appointments	Sales	Expanses
Phillip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Shicago	50	4500

Figure 2

8. Add the following code to the ***style*** element to set ***background***, ***drop-shadow***, and ***text-shadow*** effects:

```
.grid th {  
background-color: #ECECEC;  
background-image: linear-gradient(#f4ebeb, #e6fdfd); box-shadow: 0 1px 0  
#ffffff inset;  
border-top: none;  
text-shadow: 0 1px 0 #ffffff;  
}  
}
```

Figure 3 shows the result. Notice that the corners are not ***round***. Now, you will round the corners.

9. Add the following code to the ***style*** element:

```
.grid th:first-child {  
border-radius: 10px 0 0 0;  
}  
}
```

The ***th:first-child*** selector targets the first cell of the table.

10. Now, target the last cell of the first row. Add the following code:

```
.grid th:last-child {  
border-radius: 0 10px 0 0;  
}  
  
}
```

Salesman	Appointments	Sales	Expenses
Phillip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Chicago	50	4500

Figure 3

11. Add the following code to the *style* tag:

```
.grid tr:last-child td:last-child {  
border-radius: 0 0 10px 0;  
}  
  
}
```

The grid *tr:last-child td:last-child* rule targets the cell which has the placeholder as *Graham*. Next, you will target the right most cell of the last row.

12. Add the following code to the style tag:

```
.grid tr:last-child td:last-child {  
border-radius: 0 0 10px 0; }
```

Figure 4 shows the result. If you zoom in on the cell (see Figure 5) that has the placeholder *Graham*, you will see an additional border

line. Next, you will fix it.

Salesman	Appointments	Sales	Expenses
Phillip	NewYork	57	2500
Brent	Miami	48	2000
Bill	Washington	75	3700
Fred	Los Angeles	56	4000
Tim	Utah	45	6000
Graham	Shicago	50	4500

Figure 4

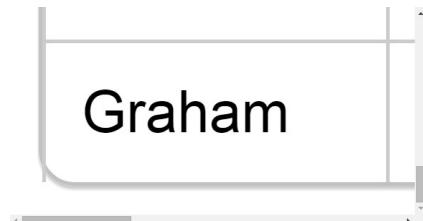


Figure 5

13. Add the following code to the style tag:

```
.grid td:first-child,  
th:first-child {  
border-left: none;
```

```
}
```

Finally, let's add a background transition to the rows when use hovers mouse over them.

14. Add the following code to the style tag:

```
.grid tr:hover {  
background: #e2dddd;  
transition: all 0.1s ease-in-out;
```

```
}
```

Below is the complete code for the tutorial.

File:/tutorials/tutorial-2.html

```
<!DOCTYPE html>
<html>

<head>
<title> Styling a Table </title> <style>
body {
font-family: helvetica, verdana, sans-serif; }

table {
border-collapse: separate;
border-spacing: 0;
width: 100%; }

.grid {
border: solid #c3c0c0 1px;
border-radius: 10px;
box-shadow: 0 1px 1px #ccc;

}

.grid td,
th {
border-left: 1px solid #ccc;
border-top: 1px solid #ccc;
padding: 10px;
text-align: left;
```

}

```
.grid th {  
background-color: #ECECEC;  
background-image: linear-gradient(#f4ebeb, #e6fdfd); box-shadow: 0 1px 0  
#ffffff inset;  
border-top: none;  
text-shadow: 0 1px 0 #ffffff;
```

}

```
.grid th:first-child {  
border-radius: 10px 0 0 0;
```

}

```
.grid th:last-child {  
border-radius: 0 10px 0 0;
```

}

```
.grid tr:last-child td:last-child {  
border-radius: 0 0 10px 0;
```

}

```
.grid tr:last-child td:last-child {  
border-radius: 0 0 10px 0;
```

}

```
.grid td:first-child,  
th:first-child {
```

```
border-left: none;
```

```
}
```

```
.grid tr:hover {  
background: #e2ddd;  
transition: all 0.1s ease-in-out;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>  
<table class="grid">  
<tr>  
<th>Salesman</th>  
<th>Appointments</th>  
<th>Sales</th>  
<th>Expenses</th>  
</tr>  
<tr>  
<td>Phillip</td>  
<td>New York</td>  
<td>57</td>  
<td>2500</td>  
</tr>  
<tr>  
<td>Brent</td>  
<td>Miami</td>  
<td>48</td>  
<td>2000</td>  
</tr>  
<tr>  
<td>Bill</td>
```

```
<td>Washington</td>
<td>75</td>
<td>3700</td>
</tr>
<tr>
<td>Fred</td>
<td>Los Angeles</td>
<td>56</td>
<td>4000</td>
</tr>
<tr>
<td>Tim</td>
<td>Utah</td>
<td>45</td>
<td>6000</td>
</tr>
<tr>
<td>Graham</td>
<td>Shicago</td>
<td>50</td>
<td>4500</td>
</tr>
</table>
</body>

</html>
```

Tutorial - 3: Styling a login form In this tutorial, you will style a login form (see Figure 1).



Figure 1

Follow these steps:

- 1.** Open the code editor you are using and then create a new HTML document. Save the document with the name *tut3.html*.
- 2.** Type following code and then save the html file:

```
<!DOCTYPE html>
<html>

<head>
<title> Styling a Table </title> <style>
</style>
</head>

<body>

</body>

</html>
```

- 3.** Add the following code between the *<body>* and *</body>* tags to create the HTML code for the tutorial:

```
<div class="box">

<form action="example.php">
<fieldset>
```

```

<legend>Login to your account:</legend> <div>
  <label for="email" class="title">Email:</label> <input type="email"
  id="email" name="email" />
</div>
<div>
  <label for="pwd" class="title">Password:</label> <input type="password"
  id="pwd" name="pwd" />
</div>
<div>

<input type="checkbox" name="reme" id="reme" checked /> <label
for="reme" class="title">Remember me</label>

</div>
<div>
  <input type="submit" value="LOGIN" id="submit" /> </div>
<div class="link">

<a href="example.php">Forget Password?</a> </div>

</fieldset>

</form>

</div>

```

Next, you will start styling the form elements. Add the code inside the ***style*** element.

4. Add the following code to specify the font family:

```

body {
  font-family: helvetica, verdana, sans-serif;
}

```

5. Add the following code to define the ***base*** of the form:

```
.box {
```

```
background: #3361a1;  
border-radius: 15px;  
width: 290px;  
height: 300px;  
  
}
```

Figure 2 shows the result.



Figure 2

6. Add the following code to change the *color* of the text to *white*:

```
.title {  
color: white;  
  
}
```

7. Add the following code to style *fieldset* and *legend* elements:

```
fieldset {  
margin: 25px;  
width: 200px;  
border: 1px solid #dcdcdc;  
border-radius: 10px;  
padding: 20px;  
text-align: left;
```

```
}
```

```
legend {  
color: white;  
font-weight: bold;  
background-color: #2989d8;  
border: 1px solid #dcdcdc;  
border-radius: 10px;  
padding: 10px 10px;  
text-align: right;  
font-size: 14px;
```

```
}
```

```
}
```

Figure 3 shows the result.



Figure 3

8. Add the following code to style the input elements of type *email* and *password*:

```
input[type="email"],  
input[type="password"] {  
border-style: solid;
```

```
border-width: 1px;  
border-color: rgb(64, 114, 183);  
background-color: rgb(0, 52, 113);  
border-radius: 5px;  
padding: 5px 5px 5px 5px;  
margin-bottom: 10px;  
display: block;  
  
}
```

- 9.** Add the following code to define the behavior of the *input boxes* and when user *hovers* mouse over them:

```
input:focus {  
background-color: #c4d8f2;  
border: 1px solid #f6fbfc;
```

```
}
```

Figure 4 shows the result.



Figure 4

- 10.** Add the following code to style the *submit* button:

```
input[type="submit"] {  
width: 80px;
```

```

height: 33px;
margin-top: 10px;
color: white;
border-radius: 5px;
border: 1px solid rgb(10, 61, 129);
box-shadow: 0.5px 0.866px 2px 0px rgba(2, 23, 51, 0.259); text-shadow: 0px
1px 1px #ffffff;
background: -webkit-linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca
51%, #7db9e8 100%); border-bottom: 2px solid #2a5be0;

}


```

```

input[type="submit"]:hover {
color: bisque;
border: 1px solid #a4a4a4;
border-top: 2px solid #b2b2b2;
background: linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca 51%,
#7db9e8 100%); }


```

Figure 5 shows the result.



Figure 5

11. Finally, add the following code to style the Forget password link:

```

.link {
margin-top: 10px;


```

}

```
.link [href] {  
margin-top: 25px;  
color: #89f889;
```

}

Below is the complete code for the tutorial:

[File:/tutorials/tutorial-3.html](#)

```
<!DOCTYPE html>  
<html>  
  
<head>  
<title> Styling a Login Form </title> <style>  
body {  
font-family: helvetica, verdana, sans-serif; }
```

```
.box {  
background: #3361a1;  
border-radius: 15px;  
width: 290px;  
height: 300px;
```

}

```
.title {  
color: white;
```

}

```
fieldset {  
margin: 25px;  
width: 200px;  
border: 1px solid #dcdcdc;  
border-radius: 10px;  
padding: 20px;  
text-align: left;  
}  
}
```

```
legend {  
color: white;  
font-weight: bold;  
background-color: #2989d8;  
border: 1px solid #dcdcdc;  
border-radius: 10px;  
padding: 10px 10px;  
text-align: right;  
font-size: 14px;  
}  
}
```

```
input[type="email"],  
input[type="password"] {  
border-style: solid;  
border-width: 1px;  
border-color: rgb(64, 114, 183);  
background-color: rgb(0, 52, 113);  
border-radius: 5px;  
padding: 5px 5px 5px 5px;  
margin-bottom: 10px;  
display: block;  
}  
}
```

```
input:focus {  
background-color: #c4d8f2;  
border: 1px solid #f6fbfc;  
  
}  
  
input[type="submit"] {  
width: 80px;  
height: 33px;  
margin-top: 10px;  
color: white;  
border-radius: 5px;  
border: 1px solid rgb(10, 61, 129);  
box-shadow: 0.5px 0.866px 2px 0px rgba(2, 23, 51, 0.259); text-shadow: 0px  
1px 1px #ffffff;  
background: -webkit-linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca  
51%, #7db9e8 100%); border-bottom: 2px solid #2a5be0;  
  
}
```

```
input[type="submit"]:hover {  
color: bisque;  
border: 1px solid #a4a4a4;  
border-top: 2px solid #b2b2b2;  
background: linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca 51%,  
#7db9e8 100%); }
```

```
.link {  
margin-top: 10px;  
  
}  
  
.link [href] {  
margin-top: 25px;
```

```
color: #89f889;

}

</style>

</head>

<body>
<div class="box">

<form action="example.php">
<fieldset>
<legend>Login to your account:</legend> <div>
<label for="email" class="title">Email:</label> <input type="email"
id="email" name="email" />
</div>
<div>
<label for="pwd" class="title">Password:</label> <input type="password"
id="pwd" name="pwd" />
</div>
<div>

<input type="checkbox" name="reme" id="reme" checked /> <label
for="reme" class="title">Remember me</label>

</div>
<div>
<input type="submit" value="LOGIN" id="submit" /> </div>
<div class="link">

<a href="example.php">Forget Password?</a> </div>

</fieldset>
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

Tutorial - 4: Creating fixed width Layout In this tutorial, you will create a **fixed** width layout, see Figure 1. Figure 2 shows the **class** and **id** structure of the layout. You can use a grid system such as **960 Grid System** to create the fix width layout. The fixed width layout stays the same even if you resize the browser window. To create a fixed width layout, the width of the main element is generally specified using **pixels**.

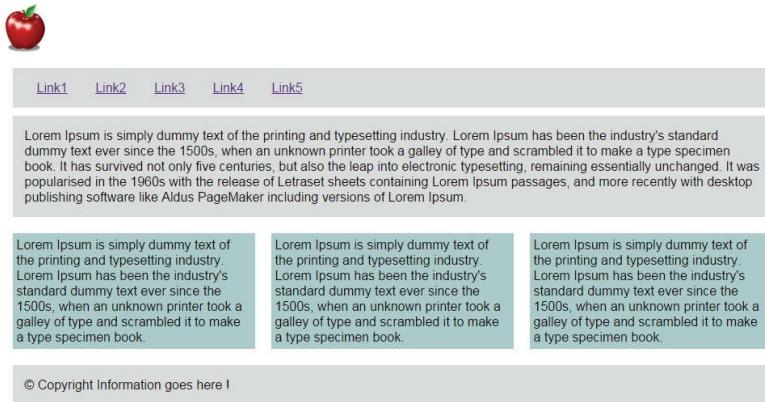


Figure 1

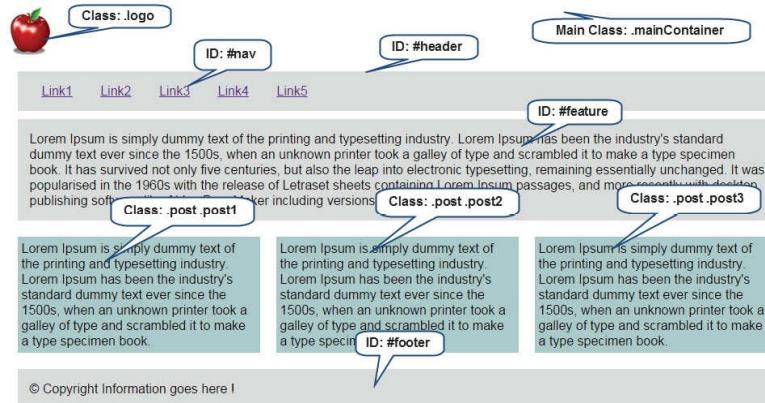


Figure 2



960 Grid System

For more information on this *Grid* system, visit the following link:
<http://960.gs>.

Follow these steps:

1. Let's start styling the layout shown in Figure 1. Open the code editor of your choice and then create a new HTML document. Save the document with the name *tut4.html*. Type following code and then save the html file:

```
<!DOCTYPE html>
<html>

<head>
<title> Fixed Width Layout </title> <style>
</style>
</head>

<body>

</body>

</html>
```

2. This is the basic HTML template that you will be using to create the layout. Add the following code between the *<body>* and *</body>* tags to create the HTML code for the tutorial:

```
<div class="mainContainer">
<div id="header">
 <div id="nav">
<ul>
<li><a href="#">Link1</a> </li> <li><a href="#">Link2</a> </li> <li><a href="#">Link3</a> </li> <li><a href="#">Link4</a> </li> <li><a
```

```
href="#">Link5</a> </li> </ul>
</div>
</div>
<div id="feature">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p> </div>
<div class="post post1">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post2">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post3">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
</div>
<div id="footer">
<p>&copy; Copyright Information goes here !</p> </div>
</body>
```

Next, you will *reset* the elements. You can use a reset stylesheet to get rid of the *browser inconsistencies*. It clears properties like the *default line heights, margins and font sizes of headings*, and so forth. I am using the *Reset CSS stylesheet* available at <http://meyerweb.com>.

3. Add the following code to the style element to *reset* the elements:

```
/* http://meyerweb.com/eric/tools/css/reset/
```

```
v2.0 | 20110126
```

```
License: none (public domain)
```

```
*/
```

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,  
blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img,  
ins, kbd, q, s, samp,  
small, strike, strong, sub, sup, tt, var,  
b, u, i, center,  
dl, dt, dd, ol, ul, li,  
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details,  
embed,  
figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section,  
summary,  
time, mark, audio, video {  
margin: 0;  
padding: 0;  
border: 0;  
font-size: 100%;  
font: inherit;  
vertical-align: baseline;
```

}

```
/* HTML5 display-role reset for older browsers */
```

```
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav,  
section {  
display: block;
```

}

```
body {  
line-height: 1;
```

}

```
ol, ul {  
list-style: none;
```

}

```
blockquote, q {  
quotes: none;
```

}

```
blockquote:before, blockquote:after,  
q:before, q:after {  
content: "";  
content: none;
```

}

```
table {  
border-collapse: collapse;  
border-spacing: 0;
```

}

```
/* END RESET */
```

Add the following code to the `style` element:

```
body {  
font-family: helvetica, verdana, sans-serif; width: 960px;  
margin: 0 auto;  
line-height: 1.2;  
  
}
```

This code sets the fixed width of `960px` for the layout. The `margin: 0 auto;` declaration centers the content on the page. Also, this code defines `font family` and `height` of the line. Next, you will set the height of the main container to `100%` and `overflow` to `auto`.

4. Add the following code to the `style` element:

```
.mainContainer {  
height: 100%;  
overflow: auto;  
  
}
```

The `overflow` property lets you control the `overflowed content` that is not displayed to the user. Next, you will set the `background color` for the `nav`, `features`, and `footer` ids.

5. Add the following code:

```
#nav,  
#feature,  
#footer {  
background-color: #dadbdb;  
padding: 15px;  
margin: 10px;
```

}

Figure 3 shows the result. Next, you will style the three columns.



Link1
Link2
Link3
Link4
Link5

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Copyright information goes here !

Figure 3

6. Add the following code:

```
.post1,  
.post2,  
.post3 {  
background-color: #aecbcb;  
float: left;  
margin: 10px;  
width: 290px;  
padding: 5px;
```

}

This code floats the columns to the *left* and assigns *padding*, *margin*, and *background color* properties to them. The *width* of the box is set to *290px*. So the total width is equal to *960px*. Notice in Figure 4, the links are not *inline*. Next, you will make them *inline* and add a padding of *15px*.

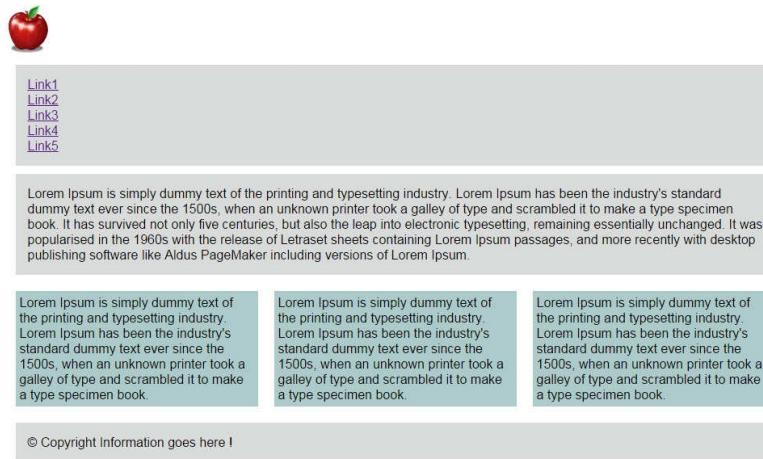


Figure 4

7. Add the following code:

```
li {  
display: inline;  
padding: 15px;  
  
}
```

Finally, you will give some *margin* to the *logo*.

8. Add the following rule for logo:

```
.logo {  
margin-bottom: 5px;  
margin-top: 10px;  
  
}
```

Below is the complete code for the layout:

File:/tutorials/tutorial-4.html

```
<!DOCTYPE html>
<html>

<head>
<title> Fixed Width Layout </title> <style>
/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126
License: none (public domain)
```

```
 */
```

```
html,
body,
div,
span,
applet,
object,
iframe,
h1,
h2,
h3,
h4,
h5,
h6,
p,
blockquote,
pre,
a,
abbr,
acronym,
address,
big,
cite,
code,
del,
dfn,
```

em,
img,
ins,
kbd,
q,
s,
samp,
small,
strike,
strong,
sub,
sup,
tt,
var,
b,
u,
i,
center,
dl,
dt,
dd,
ol,
ul,
li,
fieldset,
form,
label,
legend,
table,
caption,
tbody,
tfoot,
thead,
tr,
th,
td,
article,

```
aside,  
canvas,  
details,  
embed,  
figure,  
figcaption,  
footer,  
header,  
hgroup,  
menu,  
nav,  
output,  
ruby,  
section,  
summary,  
time,  
mark,  
audio,  
video {  
margin: 0;  
padding: 0;  
border: 0;  
font-size: 100%;  
font: inherit;  
vertical-align: baseline;  
}  
/* HTML5 display-role reset for older browsers */
```

```
article,  
aside,  
details,  
figcaption,  
figure,
```

```
footer,  
header,  
hgroup,  
menu,  
nav,  
section {  
display: block;  
  
}
```

```
body {  
line-height: 1;  
  
}
```

```
ol,  
ul {  
list-style: none;  
  
}
```

```
blockquote,  
q {  
quotes: none;  
  
}
```

```
blockquote:before,  
blockquote:after,  
q:before,  
q:after {  
content: " ";  
content: none;
```

}

```
table {  
border-collapse: collapse;  
border-spacing: 0;
```

}

```
/* END RESET */
```

```
body {  
font-family: helvetica, verdana, sans-serif; width: 960px;  
margin: 0 auto;  
line-height: 1.2;
```

}

```
.mainContainer {  
height: 100%;  
overflow: auto;
```

}

```
#nav,  
#feature,  
#footer {  
background-color: #dadbdb;  
padding: 15px;  
margin: 10px;
```

}

```
.post1,
```

```
.post2,  
.post3 {  
background-color: #aecbcb;  
float: left;  
margin: 10px;  
width: 290px;  
padding: 5px;  
  
}  
  
li {  
display: inline;  
padding: 15px;  
  
}  
  
.logo {  
margin-bottom: 5px;  
margin-top: 10px;  
  
}  
  
</style>  
</head>  
  
<body>  
<div class="mainContainer">  
<div id="header">  
 <div id="nav">  
<ul>  
<li><a href="#">Link1</a> </li> <li><a href="#">Link2</a> </li> <li><a href="#">Link3</a> </li> <li><a href="#">Link4</a> </li> <li><a href="#">Link5</a> </li> </ul>  
</div>
```

```
</div>
<div id="feature">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p> </div>
<div class="post post1">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post2">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post3">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
</div>
<div id="footer">
<p>&copy; Copyright Information goes here !</p> </div>
</body>

</html>
```

Tutorial - 5: Creating a liquid Layout In this tutorial, you will create a **liquid** layout, see Figure 1. In liquid layouts, the content is stretched and contracted as the user increases or decreases the size of the browser window. Generally, units are specified using **percentages** in

liquid layouts.

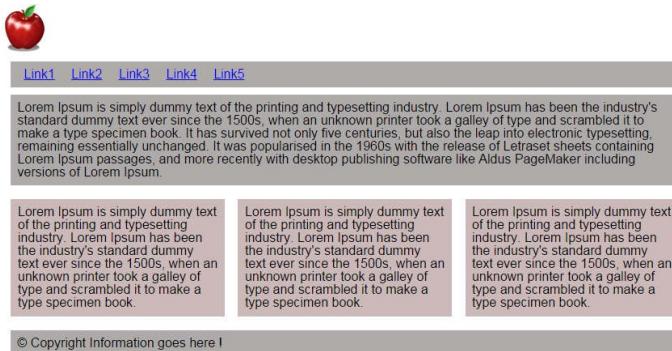


Figure 1

1. Let's start styling the layout shown in Figure 1. Open the code editor of your choice and then create a new HTML document. Save the document with the name **tut5.html**. Type the following code and then *save* the **html** file:

```
<!DOCTYPE html>
<html>

<head>
<title> Liquid Layout </title>
<style>
</style>
</head>

<body>

</body>

</html>
```

2. This is the basic HTML template that you will be using to create the layout. Add the following code between the **<body>** and **</body>** tags to create the HTML code for the tutorial:

```
<div class="mainContainer">
<div id="header">
```

```
 <div id="nav">
<ul>
<li><a href="#">Link1</a> </li> <li><a href="#">Link2</a> </li> <li><a href="#">Link3</a> </li> <li><a href="#">Link4</a> </li> <li><a href="#">Link5</a> </li> </ul>
</div>
</div>
<div id="feature">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p> </div>
<div class="post post1">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post2">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post3">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
</div>
<div id="footer">
<p>&copy; Copyright Information goes here !</p> </div>
```

This is the same code used in Tutorial 4. Next, you will *reset* the

elements. You can use a *reset* stylesheet to get rid of the browser inconsistencies. It clears properties like the *default line heights*, *margins* and *font sizes* of *headings*, and so forth. I am using the Reset CSS stylesheet available at <http://meyerweb.com>.

3. Add the following code to the style element to reset the elements:

```
/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126
License: none (public domain)
```

*/

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img,
ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details,
embed,
figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section,
summary,
time, mark, audio, video {
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
```

}

```
/* HTML5 display-role reset for older browsers */
```

```
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav,  
section {  
display: block;
```

```
}
```

```
body {  
line-height: 1;
```

```
}
```

```
ol, ul {  
list-style: none;
```

```
}
```

```
blockquote, q {  
quotes: none;
```

```
}
```

```
blockquote:before, blockquote:after,  
q:before, q:after {  
content: " ";  
content: none;
```

```
}
```

```
table {  
border-collapse: collapse;  
border-spacing: 0;
```

```
}
```

```
/* END RESET */
```

4. Add the following code to the *style* element:

```
body {  
    font-family: helvetica, verdana, sans-serif; width: 90%;  
    margin: 0 auto;  
}
```

This code defines width as *90%*. The remaining *10%* width creates a small gap between the *left* and *right* side of the browser window and main content of the page. The *margin: 0 auto;* declaration *centers* the content on the page.

5. Add the following style code:

```
.mainContainer {  
    overflow: auto;  
}
```

This code sets the overflowing of the content to the *auto* mode. Next, you will specify width of the columns.

6. Add the following style code:

```
.post1,  
.post2,  
.post3 {  
    width: 29.3%;  
    margin: 1%;  
    padding: 1%;  
    float: left;
```

```
background-color: #ccbaba;
```

```
}
```

The width of a post column is set to **31.3%** with a margin of **1%**. Therefore, the total width of the columns will be **$29.3*3+1*6$ (margin) + $1*6$ (padding)=99.9%**. Figure 2 shows the result.

7. Add the following code to set **nav** list to **inline** display:

```
li {  
    display: inline;  
    padding: 0.5em;
```

```
}
```

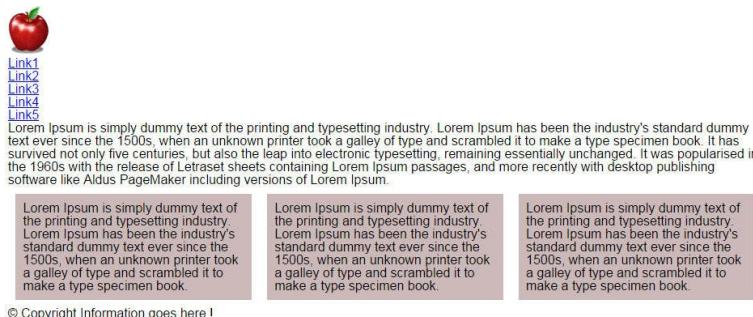


Figure 2

8. Add the following code to set the background colors and specify padding for **nav**, **feature**, and **footer** ids.

```
#nav,  
#footer,  
#feature {
```

```
background-color: #afabab;
```

```
}
```

```
#nav,  
#footer,  
#feature {  
padding: 0.5em;
```

```
}
```

Figure 3 shows the result. Notice that you need to add some space between the *nav* and *feature* ids. Also, you need to *align* the columns with rest of the content.

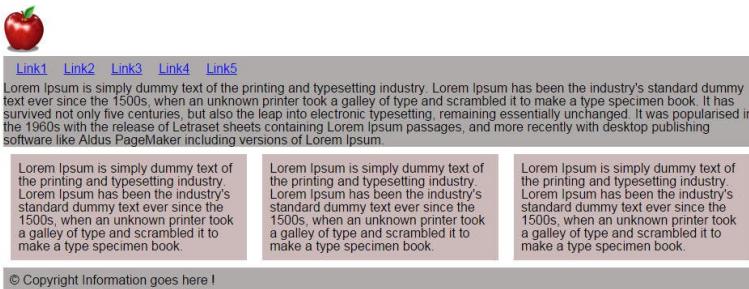


Figure 3

9. Add the following code to achieve it:

```
#nav,  
#footer,  
#feature {  
padding: 0.5em;
```

```
}
```

Now, try to resize the browser, the content within will stretch or contract as you resize the browser. Figures 4 and 5 show layout at different browser resolutions.

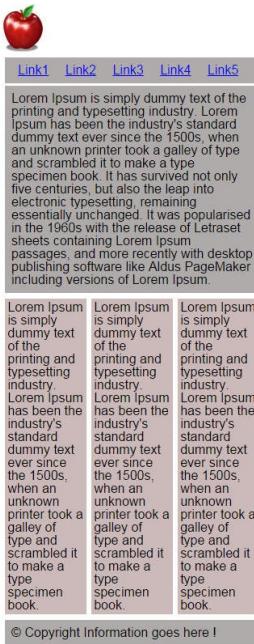


Figure 4

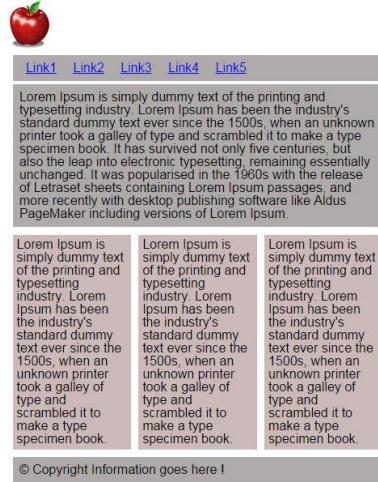


Figure 5

Below is the complete code for the tutorial:
[File:/tutorials/tutorial-5.html](#)

```
<!DOCTYPE html>
<html>

<head>
<title> Liquid Layout </title>
<style>
/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126
License: none (public domain)
```

```
 */
```

```
html,
body,
div,
span,
applet,
object,
iframe,
h1,
h2,
h3,
h4,
h5,
h6,
p,
blockquote,
pre,
a,
abbr,
acronym,
address,
big,
cite,
code,
del,
```

dfn,
em,
img,
ins,
kbd,
q,
s,
samp,
small,
strike,
strong,
sub,
sup,
tt,
var,
b,
u,
i,
center,
dl,
dt,
dd,
ol,
ul,
li,
fieldset,
form,
label,
legend,
table,
caption,
tbody,
tfoot,
thead,
tr,
th,
td,

```
article,  
aside,  
canvas,  
details,  
embed,  
figure,  
figcaption,  
footer,  
header,  
hgroup,  
menu,  
nav,  
output,  
ruby,  
section,  
summary,  
time,  
mark,  
audio,  
video {  
margin: 0;  
padding: 0;  
border: 0;  
font-size: 100%;  
font: inherit;  
vertical-align: baseline;  
}  
/* HTML5 display-role reset for older browsers */
```

```
article,  
aside,  
details,  
figcaption,
```

```
figure,  
footer,  
header,  
hgroup,  
menu,  
nav,  
section {  
display: block;  
  
}
```

```
body {  
line-height: 1;  
  
}
```

```
ol,  
ul {  
list-style: none;  
  
}
```

```
blockquote,  
q {  
quotes: none;  
  
}
```

```
blockquote:before,  
blockquote:after,  
q:before,  
q:after {  
content: " ";  
content: none;
```

}

```
table {  
border-collapse: collapse;  
border-spacing: 0;
```

}

```
/* END RESET */
```

```
body {  
font-family: helvetica, verdana, sans-serif; width: 90%;  
margin: 0 auto;  
font-size: 16px;
```

}

```
.mainContainer {  
overflow: auto;
```

}

```
.post1,  
.post2,  
.post3 {  
width: 29.3%;  
margin: 1%;  
padding: 1%;  
float: left;  
background-color: #ccbaba;
```

}

```
li {  
display: inline;  
padding: 0.5em;  
  
}  
  
#nav,  
#footer,  
#feature {  
background-color: #afabab;  
  
}  
  
#nav,  
#footer,  
#feature {  
padding: 0.5em;  
  
}  
  
#nav,  
#feature,  
#footer {  
margin: 1%;  
  
}  
  
</style>  
</head>  
  
<body>  
<div class="mainContainer">  
<div id="header">  
 <div id="nav">
<ul>
<li><a href="#">Link1</a> </li> <li><a href="#">Link2</a> </li> <li><a href="#">Link3</a> </li> <li><a href="#">Link4</a> </li> <li><a href="#">Link5</a> </li> </ul>
</div>
</div>
<div id="feature">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p> </div>
<div class="post post1">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post2">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
<div class="post post3">
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> </div>
</div>
<div id="footer">
<p>&copy; Copyright Information goes here !</p> </div>
</body>

</html>
```

