BALLARI INSTITUTE OF TECHNOLOGY AND MANAGEMENT , BALLARI.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING.

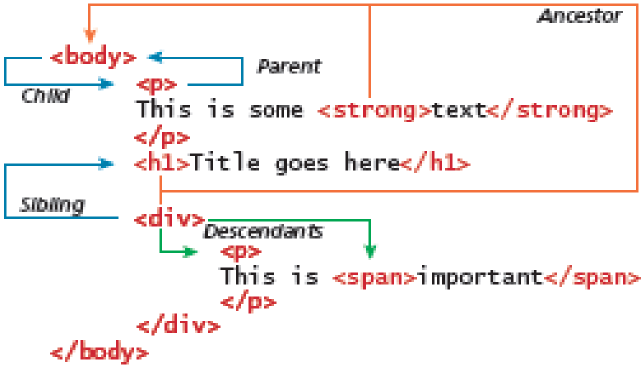
WTA 18CS63

FIRST IA TEST QUESTION BANK WITH ANSWERS

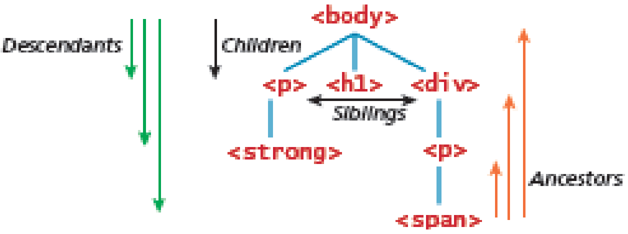
Department	**Computer Science & Engineering**	**COURSE TYPE**	**Programme Core**
Course Title	**WEB TECHNOLOGY AND ITS APPLICATION**	**Course Code**	**18CS63**
Assignment	**01**	**Marks**	
Answer the following questions in brief:			

1. Explain HTML syntax in detail.

- HTML documents are composed of textual content and HTML elements.
- The term **HTML element** is often used interchangeably with the term **tag**.
- However, an HTML element consists of the element name within angle brackets (i.e., the tag) and the content within the tag.
- A tag consists of the element name within angle brackets.



```
<body>
   <p>
   This is some <strong>text</strong>
   </p>
   <h1>Title goes here</h1>
   <div>
      <p>
      This is <span>important</span>
      </p>
   </div>
</body>
```



- The element name appears in both the
- beginning tag and the closing tag.
- The closing tag contains a forward slash followed by the element name, all enclosed within angle brackets.

Opening tag **content** **closing tag**

<p style = “font-size:40”> Hello I am a paragraph </p>

In the above example, <p> is the tag and “Hello I am a paragraph” is the content. HTML elements can also contain attributes.

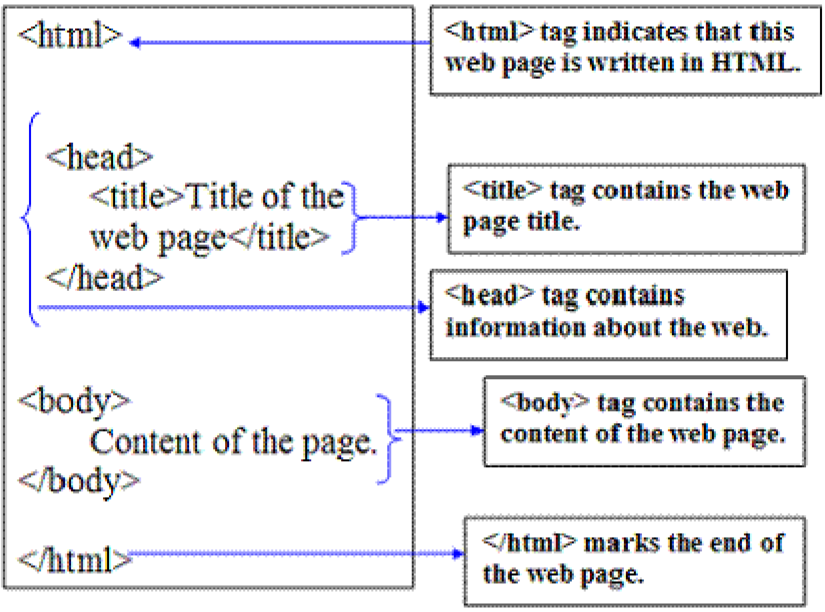
An **HTML attribute** is a ‘name = value’ pair that provides more information about the HTML element. In the above example, style is an attribute

2. What are the advantages of writing semantic HTML markup?

Importance of writing HTML markup and its advantages:

- **Maintainability**. Semantic markup is easier to update and change than web pages that contain a great deal of presentation markup. More time is spent maintaining and modifying existing code than in writing the original code.
- **Faster**. Semantic web pages are typically quicker and faster to download.
- **Accessibility**. Not all web users are able to view the content on web pages. Users with sight disabilities experience the web using voice reading software. Visiting a web page using voice reading software can be a very frustrating experience if the site does not use semantic markup.
- **Search engine optimization**. The most important users of a website are the various search engine crawlers. These crawlers are automated programs that cross the web scanning sites for their content, which is then used for users' search queries. Semantic markup provides better instructions for these crawlers.

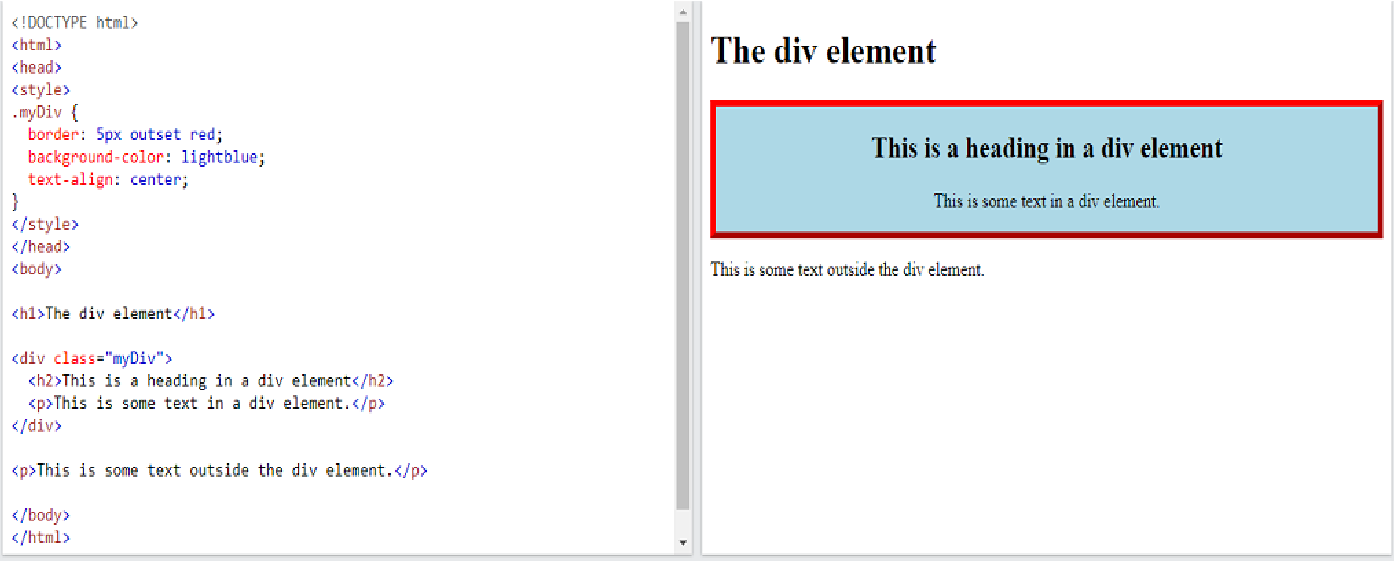
3. Illustrate structure of HTML document in detail


<html>
<head>
<title>Title of the web page</title>
</head>
<body>
Content of the page.
</body>
</html>
<html> tag indicates that this web page is written in HTML.
<title> tag contains the web page title.
<head> tag contains information about the web.
<body> tag contains the content of the web page.
</html> marks the end of the web page.

- **<!DOCTYPE html>:** This tag is used to tells the HTML version. This currently tells that the version is HTML 5.
  <html>: This is called HTML root element and used to wrap all the code.
- The **<title>** element is used to provide a broad description of the content. It is displayed by the browser in its window and/or tab.
  <head>: Head tag contains metadata, title, page CSS etc. All the HTML elements that can be used inside the <head> element are:
  <style>
- <title>
- <base>
- <noscript>
- <script>
- <meta>
- **<body>:** Body tag is used to enclose all the data which a web page has from texts to links. All the content that you see rendered in the browser is contained within this element.
- There are six levels of **headings,** specified by the tags <h1>, <h2>, <h3>, <h4>,<h5>, and <h6>, where <h1> specifies the highest-level heading.

4. Explain sample <div>-based XHTML layout (with HTML5 equivalents).

- The <div> tag defines a division or a section in an HTML document.
- The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- The <div> tag is easily styled by using the class or id attribute.
- Any sort of content can be put inside the <div> tag!
- **Note:** By default, browsers always place a line break before and after the <div> element.



```
<!DOCTYPE html>
<html>
<head>
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;
}
</style>
</head>
<body>

<h1>The div element</h1>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>

<p>This is some text outside the div element.</p>

</body>
</html>
```

```
The div element

This is a heading in a div element
This is some text in a div element.

This is some text outside the div element.
```

5. Describe benefits of CSS.

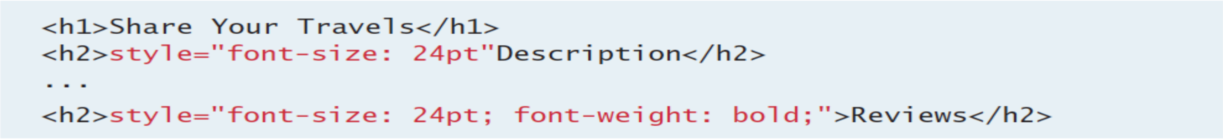
1. Easier to maintain and update
2. Greater consistency in design
3. More formatting options
4. Lightweight code
5. Faster download times
6. Search engine optimization benefits
7. Ease of presenting different styles to different viewers
8. Greater accessibility

6. **Explain Location of styles in CSS with suitable program.**

CSS style rules can be located in three different locations.

1. Inline- with in one line
2. Internal- applicable with single html file.
3. External-applicable multiple html files.

1.**Inline- with in one line**



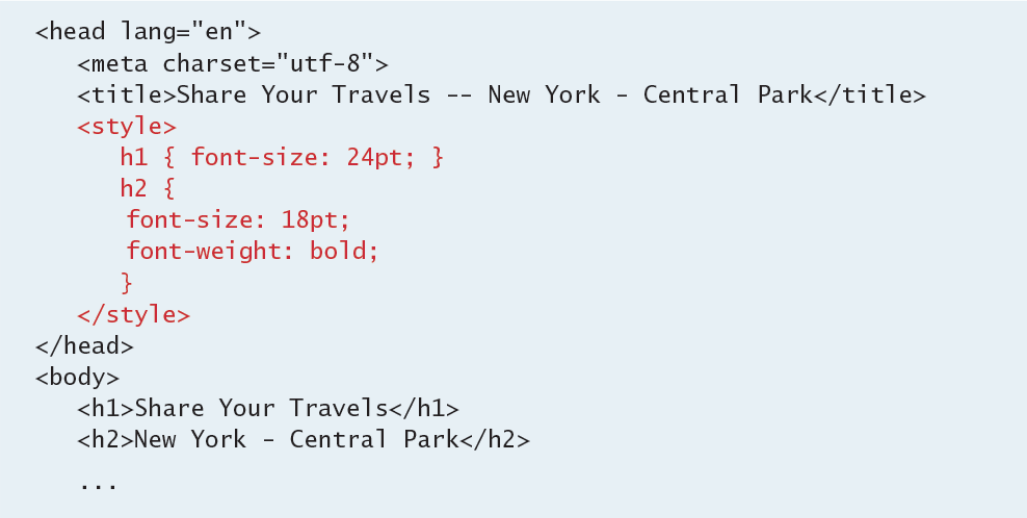
```
<h1>Share Your Travels</h1>
<h2>style="font-size: 24pt"Description</h2>
...
<h2>style="font-size: 24pt; font-weight: bold;">Reviews</h2>
```

LISTING 3.1 Internal styles example

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

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

2. Internal- applicable with single html file.(Embbeded)

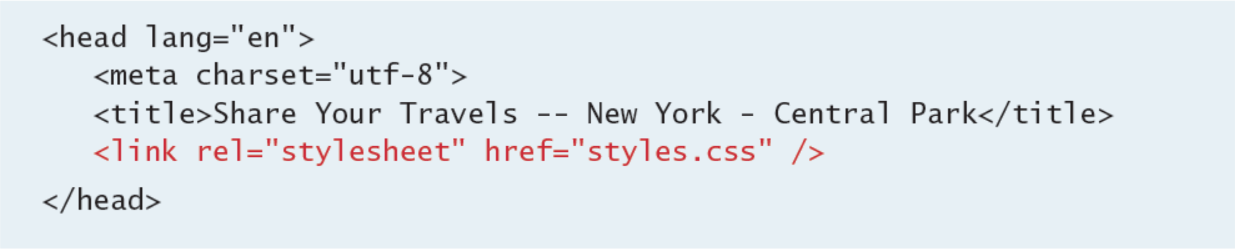


```
<head lang="en">
   <meta charset="utf-8">
   <title>Share Your Travels -- New York - Central Park</title>
   <style>
      h1 { font-size: 24pt; }
      h2 {
       font-size: 18pt;
       font-weight: bold;
      }
   </style>
</head>
<body>
   <h1>Share Your Travels</h1>
   <h2>New York - Central Park</h2>
   ...
```

LISTING 3.2 Embedded styles example

While better than inline styles, using embedded styles is also by and large discouraged.
Since each HTML document has its own <style> element, it is more difficult to consistently style multiple documents when using embedded styles.

3. External-applicable multiple html files



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

LISTING 3.3 Referencing an external style sheet

When you make a change to an external style sheet, all HTML documents that reference that style sheet will automatically use the updated version.

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

7. Describe the classification of CSS selectors with suitable snippet of code.

Selectors

The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. They are a pattern that is used by the browser to select the HTML elements that will receive the style.

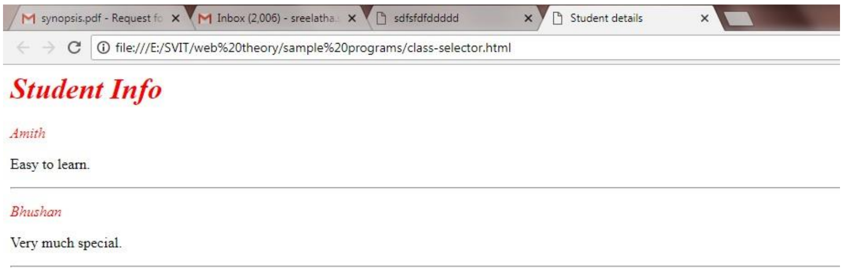
- Element Selectors
- Class Selectors
- Id Selectors
- Attribute Selectors
- Pseudo-Element and Pseudo-Class Selectors
- Contextual Selectors

Element Selectors

Element selectors select an element or group of elements of the HTML document, and the properties are applied on it.

The group of elements are separated using commas is called **grouped selector**.

Universal element selector - All elements of the document can be selected by using the * (asterisk) character.


synopsis.pdf - Request fo
Inbox (2,006) - sreelatha
sdfsfdfddddd
Student details
file:///E:/SVIT/web%20theory/sample%20programs/class-selector.html

Student Info

Amith

Easy to learn.

Bhushan

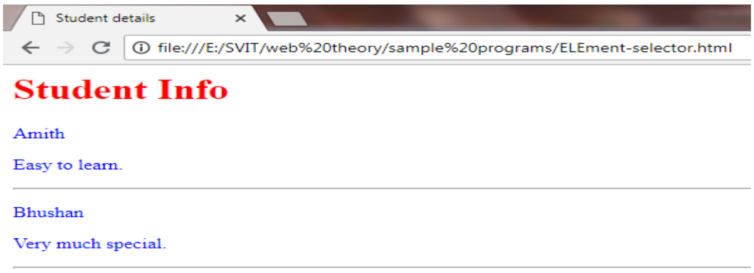
Very much special.

Eg of element selector -

```
<head>
<title>Student details </title>
<style>
*{
color:blue;}
h1 {
```

Output:

```
color: red;
}
</style>
</head>
<body>
<h1 >Student Info</h1>
<div>
<p >Amith</p>
<p>Easy to learn.</p>
</div>
<hr/>
<div>
<p >Bhushan</p>
<p>Very much special.</p>
</div>
<hr/>
</body>
```


Student details
file:///E:/SVIT/web%20theory/sample%20programs/ELement-selector.html

Student Info

Amith

Easy to learn.

Bhushan

Very much special.

Eg of grouped selector

```
p, div,
h3 {
margin:
0;
padding: 0;
}
```

Eg of universal element selector

```
*{
    color : red;
  }
```

Class Selectors

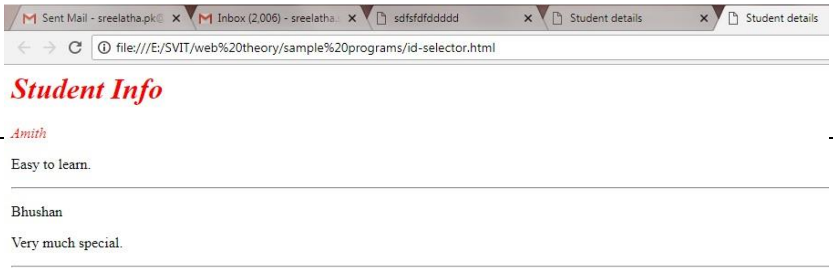
A **class selector** allows to simultaneously target different HTML elements. The HTML elements with the same class attribute value, can be styled by using a class selector.
Syntax: period (.)classname{ styles}

```
Eg:
<head>
    <title>Student details </title>
    <style>
    .first {
    font-style:
    italic; color:
    red;
    }
    </style>
</head>
<body>                                    Output:
     <h1 class="first">Student Info</h1>
     <div>
     <p class="first">Amith</p>
     <p>Easy to learn.</p>
     </div>
     <hr/>
     <div>
     <p class="first">Bhushan</p>
     <p>Very much special.</p>
  </div>
  <hr/>
</body>
```

Id Selectors

An **id selector** allows to assign style to a specific element by its id attribute. Syntax: hash (#)id name

```
Eg:
<head>
    <title>Student details </title>
    <sty
    le>
    #firs
    t {
    font-style:
```



file:///E:/SVIT/web%20theory/sample%20programs/id-selector.html

Student Info

Amith

Easy to learn.

Bhushan

Very much special.

```
        italic; color:
        red;
        }
        </style>
    </head>
    <body>
        <h1 id="first">Student Info</h1>
        <div>
        <p id="first">Amith</p>                    Output:
        <p>Easy to learn.</p>
        </div>
        <hr/>
        <div>
        <p >Bhushan</p>
        <p>Very much special.</p>
        </div>
        <hr/>
    </body>
```

Attribute Selectors

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

Eg: [src], [src$=".jpg"] , a[href*="gala"] etc.

[src] – selects all the elements which have 'src' as an attribute

[src$=".jpg"] – selects all the elements with 'src' value ending with .jpg a[href*="gala"] – selects <a> tag with 'href' value having text 'gala'.

Pseudo-Element and Pseudo-Class Selectors

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

Contextual Selectors

A **contextual selector** (in CSS3 also called **combinators**) allows to select elements based on their *ancestors*, *descendants*, or *siblings*. It selects elements based on their context or relation to other elements in the document tree.

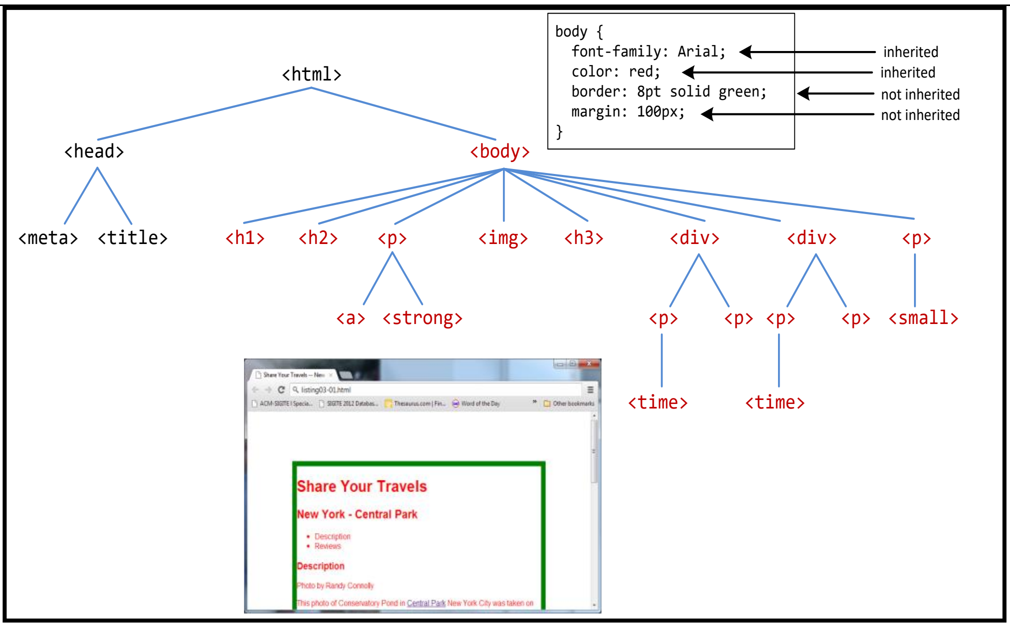
8. Explain CSS cascade principles in detail.

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

1. inheritance,
2. specificity,
3. location

1. Inheritance

- Many (but not all) CSS properties affect not only themselves but their descendants as well.
- Font, color, list, and text properties are inheritable.
- Layout, sizing, border, background and spacing properties are not.



```
body {
  font-family: Arial;
  color: red;
  border: 8pt solid green;
  margin: 100px;
}
```

Share Your Travels

New York - Central Park

- Description
- Reviews

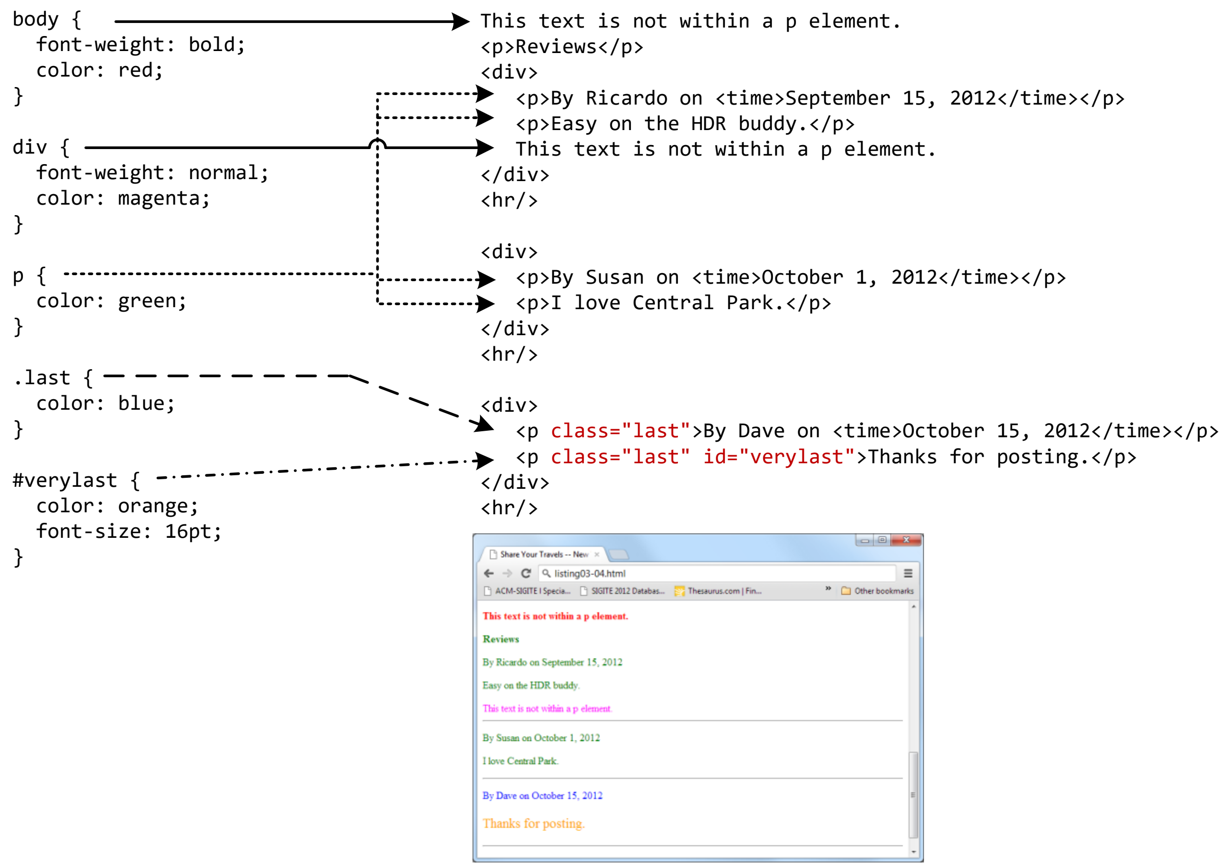
Description

Photo by Randy Connolly

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

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

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



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

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

p {
  color: green;
}

.last {
  color: blue;
}

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

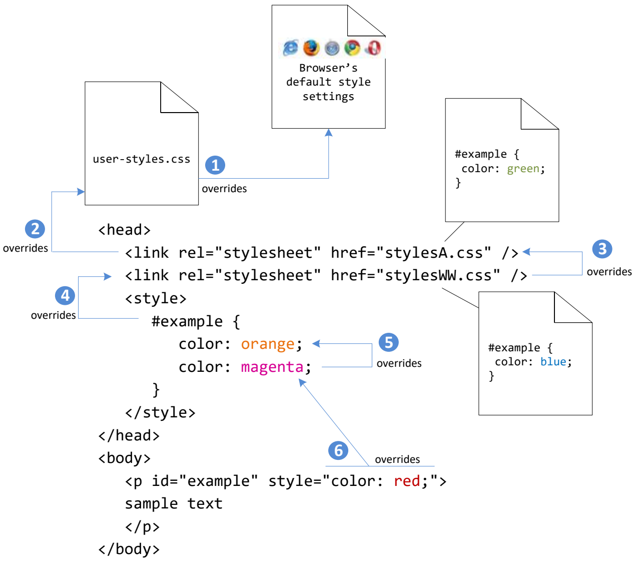
```
This text is not within a p element.
<p>Reviews</p>
<div>
   <p>By Ricardo on <time>September 15, 2012</time></p>
   <p>Easy on the HDR buddy.</p>
   This text is not within a p element.
</div>
<hr/>

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

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

```
This text is not within a p element.
Reviews
By Ricardo on September 15, 2012
Easy on the HDR buddy.
This text is not within a p element.
By Susan on October 1, 2012
I love Central Park.
By Dave on October 15, 2012
Thanks for posting.
```

3. Location

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

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



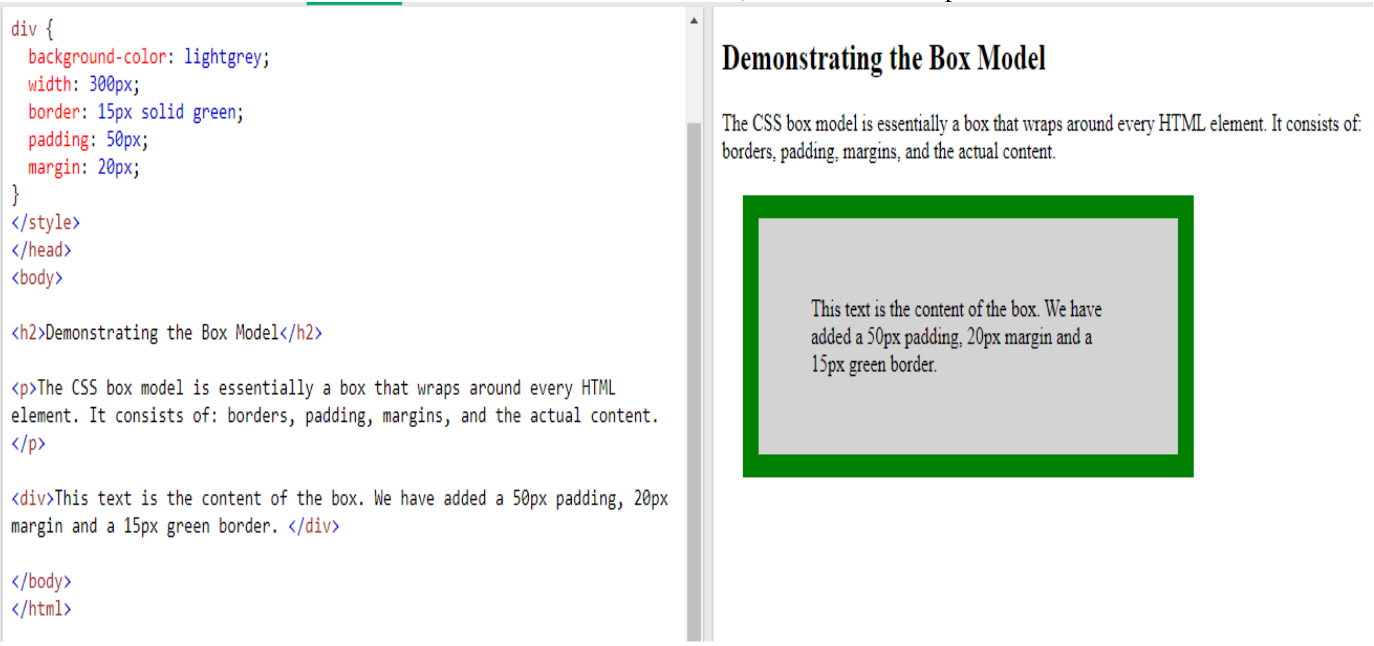
```
<head>
   <link rel="stylesheet" href="stylesA.css" />
   <link rel="stylesheet" href="stylesWW.css" />
   <style>
      #example {
         color: orange;
         color: magenta;
      }
   </style>
</head>
<body>
   <p id="example" style="color: red;">
   sample text
   </p>
</body>
```

9. Write a program to demonstrate various BOX model properties of CSS.

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.



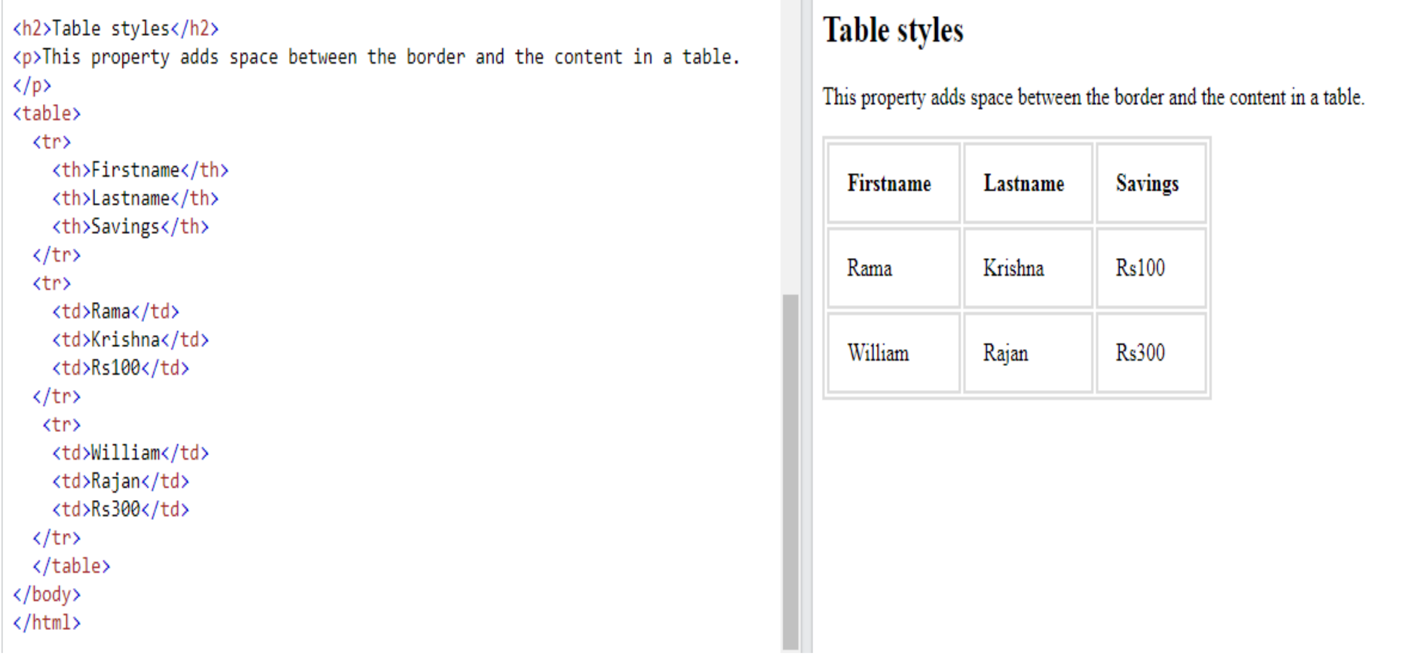
```
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

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

<div>This text is the content of the box. We have added a 50px padding, 20px
margin and a 15px green border. </div>

</body>
</html>
```

Demonstrating the Box Model

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

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.

10. Write a program to demonstrate HTML5 tables and Styling tables.



```
<h2>Table styles</h2>
<p>This property adds space between the border and the content in a table.
</p>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Rama</td>
    <td>Krishna</td>
    <td>Rs100</td>
  </tr>
   <tr>
    <td>William</td>
    <td>Rajan</td>
    <td>Rs300</td>
  </tr>
  </table>
</body>
</html>
```

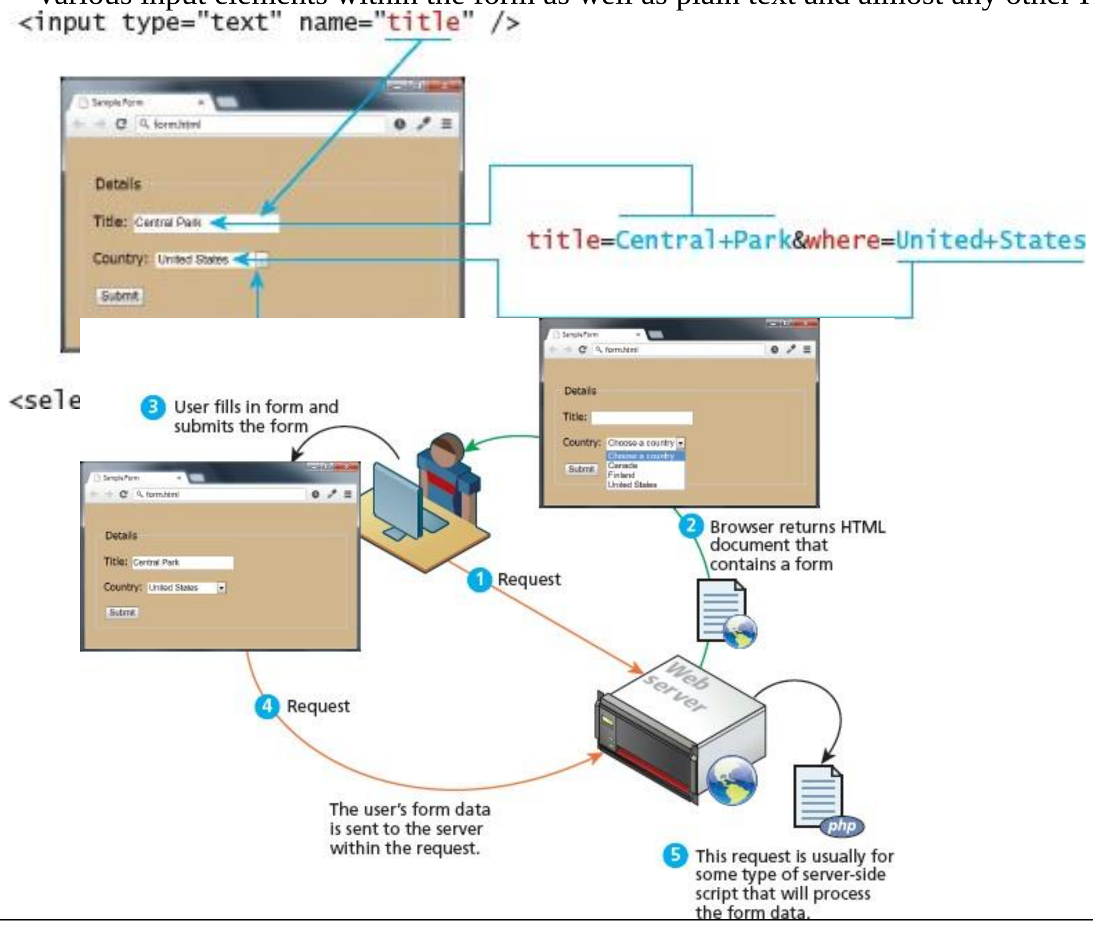
Table styles

This property adds space between the border and the content in a table.

Firstname	Lastname	Savings
Rama	Krishna	Rs100
William	Rajan	Rs300

11. Explain HTML5 Form and its various input widgets.

- **Forms** provide the user with an alternative way to interact with a web server.
- Using a form, the user can enter text, choose items from lists, and click buttons. Programs running on the server will take the input from HTML forms and processes it, or save it.
- A form is defined by a <form> element, which is a container for other elements that represent the various input elements within the form as well as plain text and almost any other HTML element.



<input type="text" name="title" />
Details
Title: Central Park
Country: United States
Submit
title=Central+Park&where=United+States
<sele
3 User fills in form and submits the form
Details
Title:
Country: Choose a country
Choose a country
Canada
Finland
United States
Submit
2 Browser returns HTML document that contains a form
1 Request
Details
Title: Central Park
Country: United States
Submit
4 Request
Web server
php
The user's form data is sent to the server within the request.
5 This request is usually for some type of server-side script that will process the form data.

The <input> Element

The HTML <input> element is the most used form element.
An <input> element can be displayed in many ways, depending on the type attribute.
Here are some examples:

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

12. Illustrate table & form accessibility and Micro-formats.

The most important guidelines for web accessibility are:

- ■ Provide text alternatives for any nontext content so that it can be changed into other forms people need, such as large print, braille, speech, symbols, or simpler language.
- ■ Create content that can be presented in different ways (for example simpler layout) without losing information or structure.
- ■ Provide ways to help users navigate, find content, and determine where they are.

Accessible Tables

HTML tables can be quite frustrating, for people with visual disability. One important way to improve the accessibility is to only use tables for tabular data, not for layout

Accessible Forms

HTML forms are also potentially problematic with respect to accessibility. The use of the <fieldset>, <legend>, and <label> elements, provide a connection between the input elements in the form.

Microformats

The web has millions of pages and there are many similar information from site to site. Most sites have Contact Us page, in which addresses and other information are displayed; calendar of upcoming events or information about products or news. These types of common information can be tagged in a similar way, and automated tools can be used to gather and transform the information.

A **microformat** is a small pattern of HTML markup and to represent common blocks of information such as people, events, and news stories so that the information in them can be extracted and indexed by software agents.

13. Explain positioning and floating of elements with suitable programs.

Floating Elements

It is possible to displace an element out of its position in the normal flow via the CSS float **property**. An element can be floated to the left or floated to the right.

When an item is floated, it is moved all the way to the far left or far right of its containing block and the rest of the content is "**re-flowed**" around the floated element.

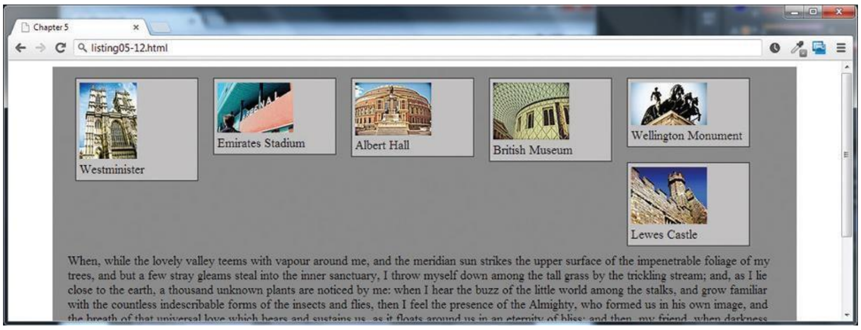
Notice that a floated block-level element must have a width specified; otherwise, the width will be set to auto, which will mean it implicitly fills the entire width of the containing block, and there will be no room available to flow content around the floated item.

Floating within a Container

It should be reiterated that a floated item moves to the left or right of its container.
The floated figure contained within an <article> element that is indented from the browser's edge. The relevant margins and padding areas are color coded to help make it clearer how the float interacts with its container.

Floating Multiple Items Side by Side

A common use of float property is to place multiple items side by side on the same line. When multiple items are floated, each element will be nestled up beside the previously floated item. All other content in the containing block will flow around all the floated elements.

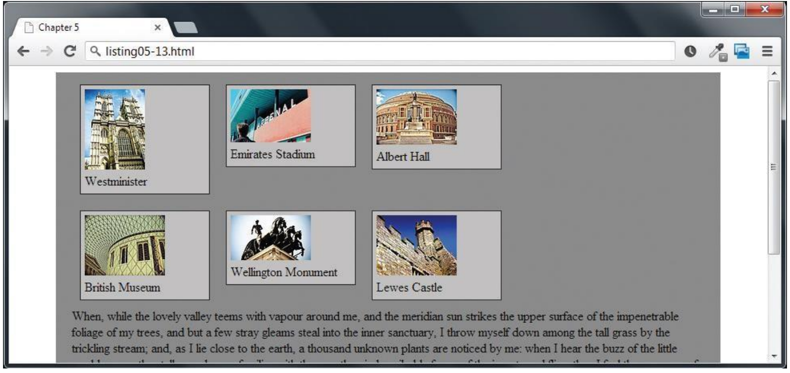


When, while the lovely valley teems with vapour around me, and the meridian sun strikes the upper surface of the impenetrable foliage of my trees, and but a few stray gleams steal into the inner sanctuary, I throw myself down among the tall grass by the trickling stream; and, as I lie close to the earth, a thousand unknown plants are noticed by me: when I hear the buzz of the little world among the stalks, and grow familiar with the countless indescribable forms of the insects and flies, then I feel the presence of the Almighty, who formed us in his own image, and the breath of that universal love which bears and sustains us, as it floats around us in an eternity of bliss; and then, my friend, when darkness

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

This arrangement of images floated changes as the browser window size changes. If suppose any element has to be stopped from flowing around a floated element, it can be done by using the **clear** CSS **property**.

By setting the clear property of third image to left, it means that there should be no elements to its left. { clear : left;}



When, while the lovely valley teems with vapour around me, and the meridian sun strikes the upper surface of the impenetrable foliage of my trees, and but a few stray gleams steal into the inner sanctuary, I throw myself down among the tall grass by the trickling stream; and, as I lie close to the earth, a thousand unknown plants are noticed by me: when I hear the buzz of the little

The other values for clear property are described below –

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

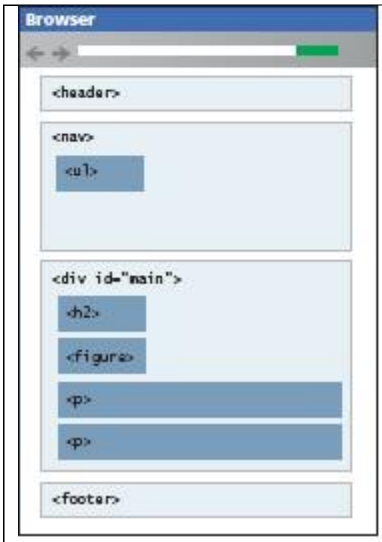
14. Demonstrate constructing of multicolumn Layouts in CSS.

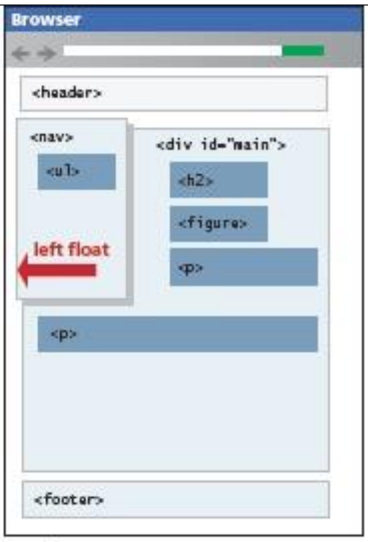
Constructing Multicolumn Layouts

Using Floats to Create Columns

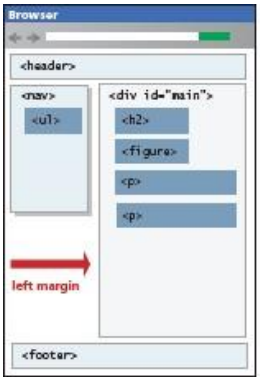
Using floats is the most common way to create columns of content. The steps for this approach are as follows –

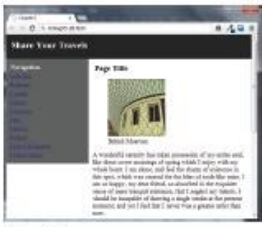
1. float the content container that will be on the left-hand side. (the floated container needs to have a width specified).
2. The other content will flow around the floated element.
3. Set the left – hand side margin for the non-floated element.



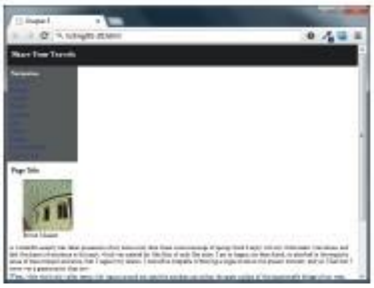


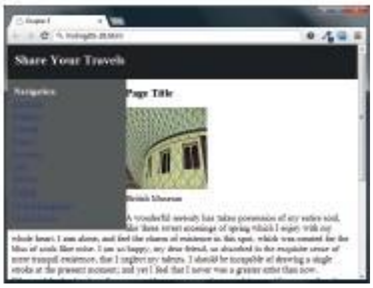
```
nav {
   …
   width: 12em;
   float: left;
}
```





```
div#main {
   …
   margin-left: 220px;
}
```





Layout without float
Property

Layout after using the float property for
left side element