

## **HTML :**

HTML stands for Hyper Text Markup Language.

It is developed by Tim Berner's Lee at W3C (World Wide Web Consortium).

It is used to create Static Web pages.

It is a global language i.e., it can be understood by all browsers.

It is not case-sensitive language.

The latest version of HTML is HTML 5

HTML documents are also called as web pages.

HTML pages can run on any browser.

### **HTML Example**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>First HTML</title>
```

```
</head>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

### **HTML Tags**

HTML markup tags are usually called HTML tags

HTML tags are keywords (tag names) surrounded by angle brackets like <html>

HTML tags normally come in pairs like <b> and </b>

The first tag in a pair is the start tag, the second tag is the end tag The end tag is written like the start

tag, with a forward slash before the tag name

Start and end tags are also called opening tags and closing tags `<tagname>content</tagname>`

## **HTML Elements**

"HTML tags" and "HTML elements" are often used to describe the same thing.

But strictly speaking, an HTML element is everything between the start tag and the end tag, including the tags:

HTML Element:

```
<p>This is a paragraph.</p>
```

## **HTML Editors**

Writing HTML

HTML code can be written in:

- Notepad
- Microsoft Visual Studio
- Adobe Dreamweaver
- Microsoft Expression
- Web Edit Plus
- Text Pad
- Notepad++
- sublimetext

etc.

However, for learning HTML we recommend a text editor like Notepad (PC) or TextEdit (Mac). We believe using a simple text editor is a good way to learn HTML.

Follow the 4 steps below to create your first web page with Notepad.

### **Step 1: Start Notepad**

To start Notepad go to:

Start

All Programs Accessories Notepad

## **Step 2: Edit Your HTML with Notepad**

Type your HTML code into your Notepad:

## **Step 3: Save Your HTML**

Select Save as.. in Notepad's file menu.

When you save an HTML file, you can use either the .htm or the .html file extension. There is no difference, it is entirely up to you.

Save the file in a folder that is easy to remember, like web.

## **Step 4: Run the HTML in Your Browser**

Start your web browser and open your html file from the File, Open menu,

or just browse the folder and double-click your HTML file.

## **The HTML <head> Element**

The <head> element is a container for all the head elements. Elements inside <head> can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

The following tags can be added to the head section: <title>, <style>, <meta>, <link>, <script>, <noscript>, and <base>.

## **The HTML <title> Element**

The <title> tag defines the title of the document.

The <title> element is required in all HTML/XHTML documents. The <title> element: defines a title in the browser toolbar

provides a title for the page when it is added to favorites displays a title for the page in searchengine results

## **A simplified HTML document:**

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Title of the document</title> </head>

<body>
The content of the document..... </body>

</html>
```

## **HTML Headings**

HTML headings are defined with the <h1> to <h6> tags.

### **Example**

```
<h1>This is a heading</h1> <h2>This is a heading</h2> <h3>This is a heading</h3> <h4>This is
a heading</h4> <h5>This is a heading</h5>

<h6>This is a heading</h6>
```

## **HTML Paragraphs**

HTML paragraphs are defined with the <p> tag.

### **Example**

```
<p>This is a paragraph.</p> <p>This is another paragraph.</p>
```

## **HTML Links**

HTML links are defined with the <a> tag.

### **Example**

```
<a href="http://www.peerstech.com">Peerstech</a> Note: The link address is specified in the href
attribute.
```

## **HTML Attributes**

HTML elements can have attributes

Attributes provide additional information about an element Attributes are always specified in the start tag

Attributes come in name/value pairs like: name="value"

## **Example**

HTML links are defined with the <a> tag. The link address is specified in the href attribute: <a href="http://www.peerstech.com">This is a link</a>

Attribute values should always be enclosed in quotes.

Double style quotes are the most common, but single style quotes are also allowed.

### **class**

Specifies one or more classnames for an element (refers to a class in a style sheet)

### **id**

Specifies a unique id for an element

### **style**

Specifies an inline CSS style for an element

### **title**

Specifies extra information about an element (displayed as a tool tip)

## **HTML Lines**

The <hr>tag creates a horizontal line in an HTML page. The hr element can be used to separate content:

```
<p>This is a paragraph</p>
```

```
<hr>
```

```
<p>This is a paragraph</p>
```

```
<hr>
```

```
<p>This is a paragraph</p>
```

## **HTML Comments**

Comments can be inserted into the HTML code to make it more readable and understandable.

Comments are ignored by the browser and are not displayed.

Comments are written like this:

<!-- This is a comment -->

## HTML Text Formatting Tags

<b> Defines bold text

<em> Defines emphasized text

<i> Defines a part of text in an alternate voice or mood

<small> Defines smaller text <strong> Defines important text <sub> Defines subscripted text <sup> Defines superscripted text

<ins> Defines inserted text

<del> Defines deleted text <u> Underline for text

## HTML Images

HTML images are defined with the <img> tag.

### Example

```

```

Note: The filename and the size of the image are provided as attributes.

### HTML Images - The <img> Tag and the Src Attribute

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

Syntax for defining an image:

```

```

### HTML Images - Set Height and Width of an Image

The height and width attributes are used to specify the height and width of an image.

The attribute values are specified in pixels by default:



## HTML Tables

Tables are defined with the <table> tag.

A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.

```
<table border="1">
```

```
<tr>
```

```
<td>row 1, cell 1</td>
```

```
<td>row 1, cell 2</td> </tr>
```

```
<tr>
```

```
<td>row 2, cell 1</td>
```

```
<td>row 2, cell 2</td>
```

```
</tr> </table>
```

```
<table border="4" width="100%" height="200px" bordercolor="red">
```

```
<thead><tr><th>EmpName</th> <th>EmpAge</th> <th>EmpSal</th> </tr></thead> <tbody>
```

```
<tr><td>Sandeep</td> <td>23</td> <td>23000</td> </td></tr> <tr><td>Sandeep</td> <td>23</td>
```

```
<td>23000</td> </td></tr> <tr><td>Sandeep</td> <td>23</td> <td>23000</td> </td></tr>
```

```
<tr><td>Sandeep</td> <td>23</td> <td>23000</td> </td></tr>
```

```
</tbody> </table>
```

cellspacing cellpadding border bordercolor width: % colspan rowspan

## HTML Lists

The most common HTML lists are ordered and unordered lists:

### HTML Unordered Lists

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag. The list items are

marked with bullets (typically small black circles).

```
<h4>unOdered List</h4> <ul>
```

```
<li type="circle">HTML</li> <li type="none">CSS</li>
```

```
<li>Javascript</li> <li>JQuery</li> </ul>
```

### **HTML ordered Lists**

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag. The list items are marked with numbers.

```
<h4>Odered List</h4>
```

```
<ol>
```

```
<li number="3">HTML</li> <li>CSS</li>
```

```
<li>Javascript</li> <li>JQuery</li> </ol>
```

### **HTML Definition Lists**

A definition list is a list of items, with a description of each item. The `<dl>` tag defines a definition list.

The `<dl>` tag is used in conjunction with `<dt>` (defines the item in the list) and `<dd>` (describes the item in the list):

```
<h4>Definitions List</h4> <dl>
```

```
<dt>HTML</dt> <dd>HyperTextMarkupLanguage</dd> <dt>CSS</dt> <dd>Cascading  
styleSheets</dd> <dt>Javascript</dt> <dd>its pa programming language</dd> </dl>
```

### **The HTML `<div>` Element**

The HTML `<div>` element is a block level element that can be used as a container for grouping other HTML elements.

The `<div>` element has no special meaning. Except that, because it is a block level element, the browser will display a line break before and after it.

When used together with CSS, the `<div>` element can be used to set style attributes to large blocks of content.



Another common use of the <div> element, is for document layout. It replaces the "old way" of defining layout using tables. Using tables is not the correct use of the <table> element. The purpose of the <table> element is to display tabular data.

### **The HTML <span> Element**

The HTML <span> element is an inline element that can be used as a container for text. The <span> element has no special meaning.

When used together with CSS, the <span> element can be used to set style attributes to parts of the text.

### **HTML Iframes**

An iframe is used to display a web page within a web page.

Syntax for adding an iframe:

```
<iframe src="URL"></iframe>
```

The URL points to the location of the separate page.

#### **Iframe - Set Height and Width**

The height and width attributes are used to specify the height and width of the iframe.

The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

#### **Iframe - Remove the Border**

The frameborder attribute specifies whether or not to display a border around the iframe. Set the attribute value to "0" to remove the border:

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

#### **Use iframe as a Target for a Link**

An iframe can be used as the target frame for a link.

The target attribute of a link must refer to the name attribute of the iframe:

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p>
```

```
<a href="http://www.peerstech.com" target="iframe_a">PeersTech</a> </p>
```

## HTML Forms

HTML forms are used to pass data to a server.

HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc.

Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for processing.

The <form> tag is used to create an HTML form:

```
<form>
input elements . </form>

<form> <fieldset>

<legend>Log In</legend>
<label>Username: <input type="text"></label> <label>Password: <input type="password"></
label> <input type="submit" value="Submit">

</fieldset> </form>
```

## Text Fields

Text fields are one line areas that allow the user to input text.

Single-line text input controls are created using an <input> element, whose type attribute has a value of text. Here's an example of a singleline text input used to take username:

### Example

```
<form>
<label for="username">Username:</label>
<input type="text" name="username" id="username">

</form>
```

## Password Field

Password fields are similar to text fields. The only difference is; characters in a password field are masked i.e. shown as asterisks or dots. This is to prevent others from reading the password on the screen. This is also a single-line text input controls created using an `<input>` element whose type attribute has a value of password.

Here's an example of a single-line password input used to take user password:

### **Example**

```
<form>
<label for="user-pwd">Password:</label>
<input type="password" name="user-password" id="user-pwd">

</form>
```

### **Radio Buttons**

Radio buttons are used to let the user select exactly one option from a pre-defined set of options. It is created using an `<input>` element whose type attribute has a value of radio.

### **Example**

```
<form>
<input type="radio" name="gender" id="male"> <label for="male">Male</label>
<input type="radio" name="gender" id="female"> <label for="female">Female</label>

</form>
```

### **Checkboxes**

Checkboxes allows the user to select one or more option from a pre-defined set of options. It is created using an `<input>` element whose type attribute has a value of checkbox.

### **Example**

```
<form>
<input type="checkbox" name="sports" id="soccer"> <label for="soccer">Soccer</label>
<input type="checkbox" name="sports" id="cricket"> <label for="cricket">Cricket</label>
<input type="checkbox" name="sports" id="baseball"> <label for="baseball">Baseball</label>

</form>
```

## **File Select box**

The file fields allow a user to browse for a local file and send it as an attachment to the form data. It normally rendered as a text box with a button that enables the user to browse for a file. However, the user can also type the path and name of the file in the text box.

This is also created using an `<input>` element, whose type attribute value is set to file.

### **Example**

```
<form>
<label for="file-select">Upload:</label>
<input type="file" name="upload" id="file-select">

</form>
```

## **Textarea**

Textarea is a multiple-line text input control that allows a user to enter more than one line of text. Multi-line text input controls are created using an `<textarea>` element.

### **Example**

```
<form>
<label for="address">Address:</label>
<textarea rows="3" cols="30" name="address" id="address"></textarea>

</form>
```

## **Select Boxes**

A select box is a dropdown list of options that allows user to select one or more option from a pull-down list of options. Select box is created using the `<select>` element and `<option>` element. The option elements within the `<select>` element define each list item.

### **Example**

```
<form>
<label for="city">Course:</label>

<select name="course" id="course"> <option value="Wad">Wad</option> <option
```

```
value="iOS">iOS</option> <option value="Android">Anroid</option>

</select> </form>
```

### Submit and Reset Buttons

A submit button is used to send the form data to a web server. When submit button is clicked the form data is sent to the file specified in the form's action attribute to process the submitted data. A reset button resets all the forms control to default values.

### Example

```
<form action="action.php" method="post" id="users"> <label for="first-name">First Name:</label>
<input type="text" name="first-name" id="first-name"> <input type="submit" value="Submit">

<input type="reset" value="Reset"> </form>

<!DOCTYPE html> <html lang="en"> <head>

<title>Example of HTML From Controls</title> <style type="text/css">

legend {

color: red; }

fieldset

{

color: green;

} </style>

</head> <body>

<form> <fieldset>

<legend>Sign In</legend>
<label for="user-name">Username:</label>

<input type="text" name="username" id="user-name"> <label for="user-pwd">Password:</label>

<input type="password" name="user-password" id="user-pwd"> </fieldset>
```

<fieldset> <legend>Gender</legend>

<input type="radio" name="sex" id="male"> <label for="male">Male</label>

<input type="radio" name="sex" id="female"> <label for="female">Female</label>

</fieldset>

<fieldset> <legend>Hobbies</legend>

<input type="checkbox" name="sports" id="soccer"> <label for="soccer">Soccer</label>

<input type="checkbox" name="sports" id="Cricket">

<label for="Cricket">Cricket</label>

<input type="checkbox" name="sports" id="Baseball"> <label for="baseball">Baseball</label>

</fieldset>

<fieldset> <legend>Address</legend>

<textarea rows="10" cols="100"></textarea> </fieldset>

<fieldset>

<legend>Upload file</legend>

<label for="file-select">Upload:</label>

<input type="file" name="upload" id="file-select">

</fieldset>

<fieldset>

<legend>Select Your City</legend>

<label for="city">City:</label> <select name="city" id="city">

<option value="Singapore">Singapore</option> <option value="Hyderabad">Hyderabad</option>

<option value="Chennai">Chennai</option> <option value="Banglore">Banglore</option>

</select> </fieldset>

<fieldset>

<legend>Action</legend>

<input type="submit" value="Submit">

```
<input type="reset" value="Reset"> </fieldset>
```

```
</form> </body>
```

```
</html>
```

## **HTML Meta Element**

The meta elements used to provide structured metadata about a web page. The <meta> element typically provide metadata such as a document's keywords, description, and author last modified, and other metadata. Any number of META elements can be placed in the head section of an HTML or XHTML document. META's name attribute provides a property name while the content attribute gives the corresponding value. The content attribute value may contain text and entities, but it may not contain HTML tag

### **Define Character Encoding in HTML**

Meta tag typically used to declare character encoding inside HTML document.

### **Keywords and Description for Search Engines:**

Some search engines use metadata, especially keywords and descriptions to index web pages; however this may not necessarily be true. Keywords giving extra weight to a document's keywords and description provide a short synopsis of the page.

### **Enable Zooming on Mobile Devices**

You can use the meta viewport tag to enable zooming of your websites on mobile devices.

```
<!DOCTYPE html> <html lang="en"> <head>
```

```
<title>Meta-tags </title>
```

```
<meta name="Peerstech" content="Peerstech"> <!--Defining Document's Author-->
```

```
<meta charset="UTF-8">
```

```
<!--Declaring Character Encoding-->
```

```
<meta name="keywords" content="HTML, CSS, References">
```

```
<meta name="description" content="Tutorials on HTML and CSS">

<!--Defining Keywords and Description-->


<meta name="viewport" content="width=device-width, initial-scale=1">

<!--Enable Zooming in Mobile Devices-->


</head> <body>

<p>Hello World!</p> </body>

</html>
```

## **HTML Script**

The script element is used to embed or reference JavaScript within an HTML document to add interactivity or affect the behaviour of web pages.

### **Client-side Scripting**

Client-side scripting refers to the type of computer programs that are executed by the user's web browser. JavaScript is the most popular client-side scripting language on the web.

### **Adding a Script to HTML Document**

The <script> element is used to define a client-side script.

### **Calling an External Script**

You can also place your scripts into a separate file, and then call that file through the src attribute in your HTML document. This is useful if you want the same scripts available to multiple documents — it saves you from repeating the same task over and over again, and makes your website much easier to maintain.

### **The HTML noscript Element**

The <noscript> element is used to provide an alternate content for users that have either disabled scripts in their browser or have a browser that doesn't support client-side scripting.

The noscript element can contain all the HTML elements that you'd include inside the <body> element of a normal HTML page.



```

<!DOCTYPE html> <html lang="en"> <head>

<meta charset="UTF-8">

<title>Embedded JavaScript Example</title> </head>

<body>
<!-- No Script-->

<noscript>
<p>Sorry, your browser does not support JavaScript!</p>

</noscript>
<!-- Internal JavaScript-->

<script type="text/javascript"> document.write("Hello World!"); alert("hello JS");

</script>
<!-- External JavaScript-->

<script type="text/javascript" src=" ../js/hello.js"></script> </body>
</html>

```

### **<style> tag**

The HTML <style> tag is used to specify style sheet for the current HTML document. Following is an example to define few style sheet rules inside <style> tag:

```

<!DOCTYPE html>
<html>
<head>
<title>HTML style Tag Example</title> <base href="http://www.google.com/" /> <style
type="text/ css">

.myclass{ background-color: #aaa;

padding: 10px; }

</style>
</head>
<body>

```

```
<p class="myclass">Hello, World!</p> </body>
```

```
</html>
```

You can use CSS in three ways in your HTML document:

- External Style Sheet - Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML <link> tag.
- Internal Style Sheet - Define style sheet rules in header section of the HTML document using <style> tag.
- Inline Style Sheet - Define style sheet rules directly along-with the HTML elements using style attribute.

Let's see all the three cases one by one with the help of suitable examples.

### **External Style Sheet**

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as .css and it will be included in HTML files using <link> tag.

### **Example**

Consider we define a style sheet file style.css which has following rules:

```
.red{  
color: red; } .thick{ font-size:20px; } .green{ color:green; }
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML External CSS</title>  
<link rel="stylesheet" type="text/css" href="/html/style.css"> </head>  
  
<body>
```

```
<p class="red">This is red</p>
<p class="thick">This is thick</p> <p class="green">This is green</p>

<p class="thick green">This is thick and green</p> </body> </html>
```

## Internal Style Sheet

If you want to apply Style Sheet rules to a single document only then you can include those rules in header section of the HTML document using `<style>` tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

## Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using `<style>` tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Internal CSS</title> <style type="text/css">

.red{
color: red; } .thick{ font-size:20px; }

.green{

color:green; }

</style>
</head>
<body>
<p class="red">This is red</p>
<p class="thick">This is thick</p>
<p class="green">This is green</p>
<p class="thick green">This is thick and green</p> </body>

</html>
```

## Inline Style Sheet

You can apply style sheet rules directly to any HTML element using style attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

### **Example**

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using style attribute of those elements.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Inline CSS</title> </head>

<body>
<p style="color:red;">This is red</p>
<p style="font-size:20px;">This is thick</p>
<p style="color:green;">This is green</p>
<p style="color:green;font-size:20px;">This is thick and
green</p> </body> </html>
```

### **HTML Entities**

Character entity references, or entities for short, enable you to use characters that cannot be expressed in the document's character encoding or that cannot be entered by a keyboard , e.g. copyright symbol ©.

Some characters are reserved in HTML, e.g. you cannot use the less than (<) or greater than (>) signs or angle brackets within your text, because the browser could mistake them for markup, while some characters are not present on the keyboard like copyright symbol ©.

To display these special characters, they must be replaced with the character entities.

### **Frequently Used HTML Character Entities**

< less than &lt; > greater than &gt;

& ampersand &amp;

&#60; &#62; &#38;

Result	Description	Entity Name	Numerical reference
Result	Description	Entity Name	Numerical reference
	non-breaking space	&nbsp;	&#160;

Result Description Entity Name Numerical reference

non-breaking &nbsp; &#160; space

<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	quotation mark	&quot;	&#34;
'	apostrophe	&apos;	&#39;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	trademark	&trade;	&#8482;

## Creating Website Layouts

Creating a website layout is the activity of positioning the various elements that make a web page in a well-structured manner and give appealing look to the website.

You can find, most of the websites on the internet have put their content in multiple rows and columns — formatted like a magazine or newspaper, to give general web users a better reading and writing environment. This can be easily achieved by using the <table> <div> or <span> tags and

adding some <styles> to them.

## 1.HTML Layout Using Tables:

Tables are the simplest method for creating layouts in HTML. Generally, this involves the process of putting the content into rows and columns.

## 2.HTML Layouts Using Div and Span

The HTML <div> (stands for division) element is used for marking out a block of content, or set of other elements inside an HTML document. It can contain further div element if required, but it cannot be contained within an inline element. The <span> element on the other hand is used for marking out sections within a block element or other inline elements.

## 3.Creating Website Layouts Using HTML5 Structural Elements

HTML5 has introduced new structural elements like <header> <footer> <nav> <section> etc. to define different parts of the web pages in a more semantic way. You can consider these elements as a replacement for commonly used classes such as .header, .footer, .nav,.section etc.

### Table Layout ::

```
<!DOCTYPE html> <html lang="en"> <head>

<meta charset="UTF-8"> <title>HTML Table Layout</title>

</head>
<body style="margin:0px;">

<table cellpadding="10px;" cellspacing="0" style="font:12px verdana, sans-serif; width:80%;
margin:0 auto;">

<!-- Header--> <tr>

<td colspan="2" style="background-color:#679BB7;">
<h1 style="font-size:18px; margin:10px 0;">Peers Tech</h1>

</td> </tr>

<!-- Container-->

<tr style="height:170px;">
<td style="background-color:#bbd2df; width:20%; vertical-align:top;">

<ul style="list-style:none; padding:0px; margin:0px;">
```

```

<li style="margin-bottom:5px;"><a href="#" style="color:#3d677e;">Home</a></li> <li
style="margin-bottom:5px;"><a href="#" style="color:#3d677e;">About</a></li> <li
style="margin-bottom:5px;"><a href="#" style="color:#3d677e;">Contact</a></li>

</ul> </td>

<td style="background-color:#f0f0f0; width:80%; vertical-align:top;"> <h2 style="font-size:16px;
margin:0px;">Welcome to our site</h2> <p>Here you will learn to create websites...</p>

</td> </tr>

<!-- footer--> <tr>

<td colspan="2" style="background-color:#679BB7;">

<p style="text-align:center; margin:5px;">copyright &copy; peerstech.com</p> </td>

</tr> </table>

</body>

</html>

```

### Div Span Layout :

```

<!DOCTYPE html> <html lang="en"> <head>

<meta charset="UTF-8"> <title>HTML Div Layout</title> <style type="text/css">

body{
font: 12px verdana, sans-serif; margin: 0px;

} .header{

padding: 10px 0;

background-color: #679BB7; }

.header h1{ font-size: 18px; margin: 10px; color: white;

} .container{

width: 80%;

```

```
margin: 0 auto; /* Align container DIV horizontally center */
```

```
background-color: #f0f0f0; }
```

```
.sidebar{ float: left;
```

```
width: 20%;
```

```
min-height: 170px; background-color: #bbd2df;
```

```
}
```

```
.sidebar .nav{
```

```
padding: 10px; }
```

```
.nav ul{
```

```
list-style: none; padding: 0px; margin: 0px;
```

```
}
```

```
.nav ul li{
```

```
margin-bottom: 5px; }
```

```
.nav ul li a{
```

```
color: #3d677e;
```

```
}
```

```
.nav ul li a:hover{
```

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

```
.content{ float: left;
```

```
width: 80%;
```

```
min-height: 170px; }
```

```
.content .section{ padding: 10px;
```

```
}
```

```
.content h2{
```

```
font-size: 16px;
```



```
margin: 0px; }
```

```
.clearfix{ clear: both;
```

```
} .footer{
```

```
background-color: #679BB7;
```

```
padding: 10px 0; }
```

```
.footer p{ text-align: center; margin: 5px;
```

```
} </style>
```

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

```
<div class="container"> <!-- header-->
```

```
<div class="header">
```

```
<h1>peers tech</h1> </div>
```

```
<!-- Container-->
```

```
<div class="wrapper">
```

```
<div class="sidebar"> <div class="nav">
```

```
<ul>
```

```
<li><a href="#">Home</a></li> <li><a href="#">About</a></li> <li><a href="#">Contact</a></li>
```

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

```
</div>
```

```
<div class="content">
```

```
<div class="section">
```

```
<h2>Welcome to our site</h2>
```

```
<p>Here you will learn to create websites...</p>
```

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

```
<div class="clearfix"></div> </div>
```

```
<!-- footer--> <div class="footer">

<p>copyright &copy; peerstech.com</p> </div>

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

## HTML5 Layout

```
<!DOCTYPE html>

<html lang="en"> <head>

<meta charset="UTF-8"> <title>HTML5 Web Page Layout</title> <style type="text/css">

body{
font: 12px verdana, sans-serif; margin: 0px;

} header{

background-color: #679BB7;

padding: 10px; }

header h1{ font-size: 18px; margin: 10px 0;

} .container{

width: 80%;

margin: 0 auto; /* Align container DIV horizontally center */

background-color: #f0f0f0; }

.sidebar{ float: left;

width: 20%;

min-height: 170px; background-color: #bbd2df;

}

.sidebar nav{

padding: 10px; }

nav ul{
```

```
list-style: none; padding: 0px; margin: 0px;

}
nav ul li{

margin-bottom: 5px;

}
nav ul li a{

color: #3d677e; }

nav ul li a:hover{ text-decoration: none;

} .content{

float: left;
width: 80%; min-height: 170px;

}
.content section{

padding: 10px; }

section h2{ font-size: 16px; margin: 0px;

}

.clearfix:after{ content: "."; display: block; height: 0;
clear: both; visibility: hidden;

} footer{

background-color: #679BB7;

padding: 10px; }

footer p{
text-align: center; margin: 5px;

} </style>

</head> <body>
```

```

<div class="container">

<header>
<h1>peers tech</h1>

</header>
<div class="wrapper clearfix">

<div class="sidebar"> <nav>

<ul>
<li><a href="#">Home</a></li> <li><a href="#">About</a></li> <li><a href="#">Contact</a></li>
</ul> </nav>

</div>
<div class="content">

<section>
<h2>Welcome to our site</h2>
<p>Here you will learn to create websites...</p>

</section> </div>

</div> <footer>

<p>copyright &copy; peerstech.com</p> </footer>

</div> </body> </html>

```

### 1) Unpaired tags list:

```

<br> <hr> <img> <meta> <input> <link>

```

### 2) Paired tags list:

```

<html> <body> <head>

```

### (all remaining)

```

<title>

```

```

<h1> <h2> <h3> <h4> <h5> <h6>

```

<b>  
<i>  
<u> <strike> <sup> <sub> <span> <div> <form> <textarea> <iframe> etc.

**List of html tags:**

1. <html> 2. <body> 3. <br>
4. <hr>
5. <h1>
6. <h2>
7. <h3>
8. <h4>
9. <h5>
10. <h6> 11. <p>
12. <b>
13. <i>
14. <u>
15. <strong> 16. <em> 17. <strike> 18. <sup>
19. <sub>
20. <img> 21. <a>
22. <ol>
23. <ul>
24. <li>
25. <dl>
26. <dt>
27. <dd>
28. <abbr> 29. <bdo> 30. <pre>
31. <table> 32. <tr>
33. <th>
34. <td>
35. <caption> 36. <iframe> 37. <title>
38. <meta>
39. <head>
40. <input>
41. <button> 42. <form>

43. <textarea> 44. <select> 45. <option> 46. <optgroup> 47. <label>

48. <fieldset> 49. <legend> 50. <link> 51. <script>

---