WEB TECHNOLOGY AND DESIGN

ISIBOR O.O.
COMPUTER SCIENCE DEPARTMENT
LAGOS CITY POLYTECHNIC
&
SOUTHWESTERN UNIVERSITY
IKEJA-NIGERIA.
osesisibor@gmail.com, 08063546421.

FIRST PUBLISHED ON MAY 2017
REVIEWED ON MARCH 2019

1

DEFINITIONS

THE INTERNET is a global network of networks with a massive store of multimedia and shared information. It seems to be everywhere and allows many people and devices to connect to it via phone line, cable, digital subscriber lines or wireless.

The Internet also referred to as the net, in simplest terms, consists of large a group of millions of computers around the world that are connected to one another. It is a network of networks that consists of millions of private, public, academic, business, and government networks, local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies such as phone lines, fibre optic lines, coaxial cable, satellites, and wireless connections.

The Internet originated as a proposal from the Advanced Research Project Agency (ARPA). The idea was to see how computers connected in a network i.e. (ARPANET) could be used to access information from research facilities and universities. In 1969, four computers (located at UCLA, Stanford Research Institute, University of California Santa Barbara and the University of Utah) were successfully connected. As time went on, other networks were connected. With four nodes by the end of 1969, the ARPANET spanned the continental United States (US) by 1971 and had connections to Europe by 1973. Though the Interconnected Network, or Internet, was originally limited to the military, government, research, and educational purposes it was eventually opened to the public. Today there are hundreds of millions of computers and other devices connected to the Internet worldwide.

INTRANET: The term “Intranet” is used to describe a network of personal computers (PC) without any personal computers on the network connected to the world outside of the Intranet. The Intranet resides behind a firewall; if it allows access from the Internet, it becomes an Extranet. The firewall helps to control access between the intranet and Internet so that only authorized users will have access to the Intranet. Usually these people are members of the same company or organization. Like the Internet itself, intranets are used to share information. Secure intranets are now the fastest-growing segment of the Internet because they are much less expensive to build and manage than private network based on proprietary protocols.

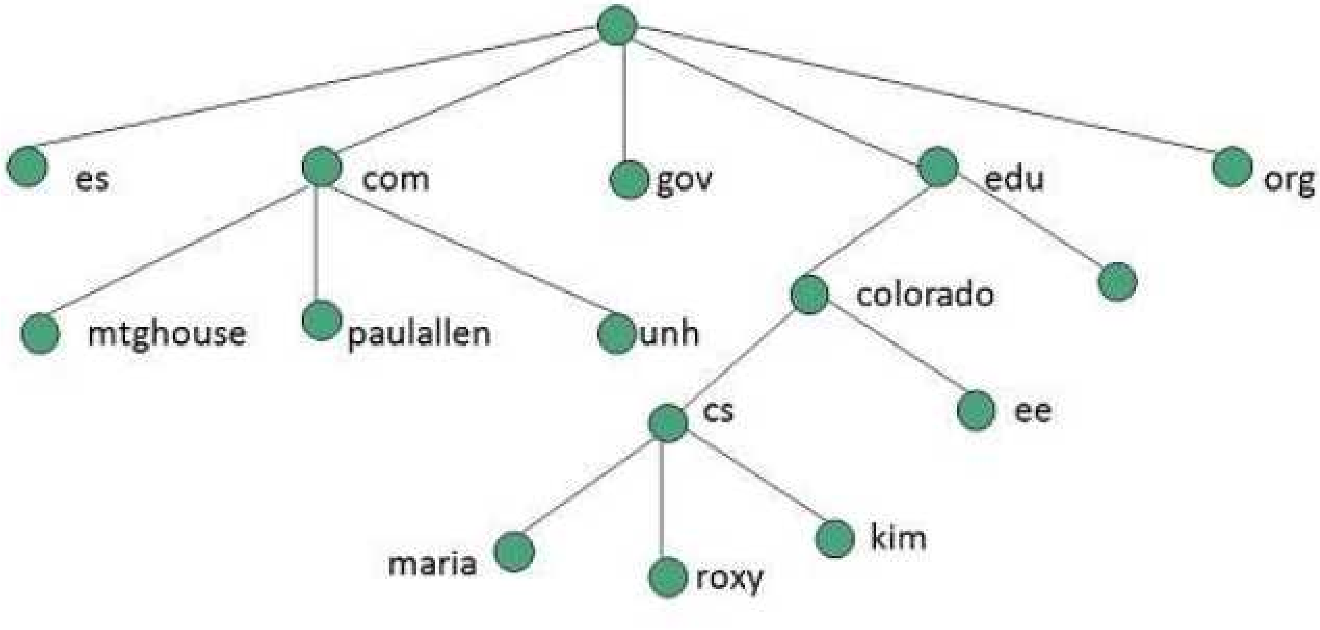
EXTRANET: Extranets are becoming a very popular means for business partners to exchange information. An Extranet is a term used to refer to an intranet that is partially accessible to authorized outsiders. Privacy and security are important issues in extranet use. A firewall is usually provided to help control access between the Intranet and Internet. In this case, the actual server will reside behind a firewall. The level of access can be set to different levels for individuals or groups of outside users.

INTERNET SERVICES

Services on the Internet: The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail.

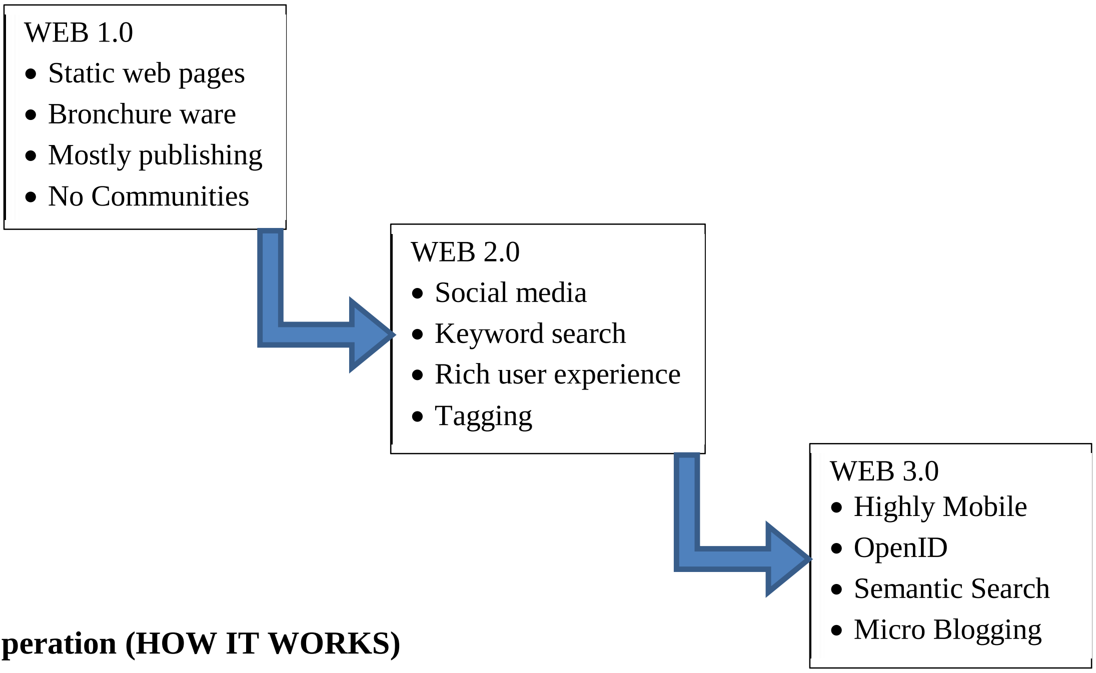
1. **World Wide Web (WWW):** The World Wide Web is a repository of information spread all over the world and linked together for easy access. It is made up of documents called pages that combine text, pictures, forms, sound, animation and hypertext links into rich communication medium.

WWW stands for **World Wide Web.** A technical definition of the World Wide Web is all the resources and users on the Internet that are using the Hypertext Transfer Protocol *HTTP*. A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium** ***W*3*C*.**
The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge. In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.
(*Internet and Web is not the same thing: Web uses internet to pass over the information.*)



World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.** The following diagram briefly defines evolution of World Wide Web:

The following diagram briefly defines evolution of World Wide Web:



WEB 1.0

- Static web pages
- Bronchure ware
- Mostly publishing
- No Communities

WEB 2.0

- Social media
- Keyword search
- Rich user experience
- Tagging

WEB 3.0

- Highly Mobile
- OpenID
- Semantic Search
- Micro Blogging

WWW Operation (HOW IT WORKS)

WWW works on client- server approach. Following steps explains how the web works:

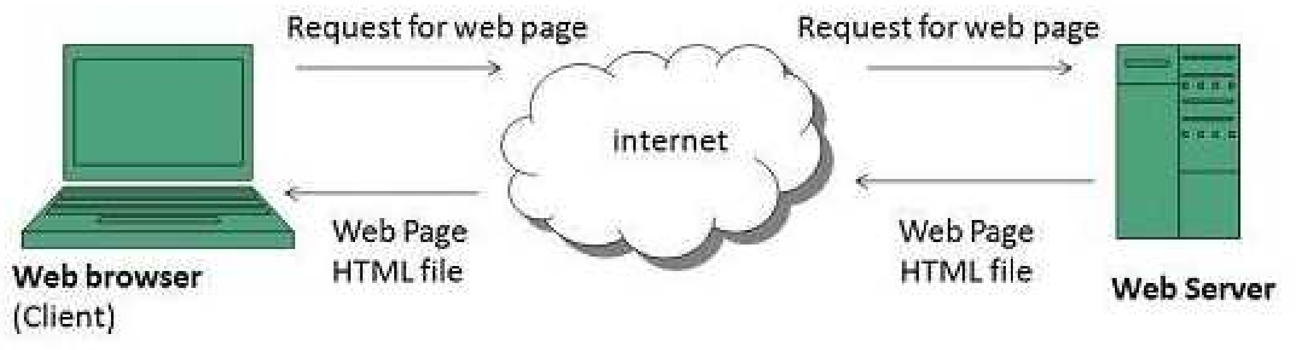
1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.

2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.

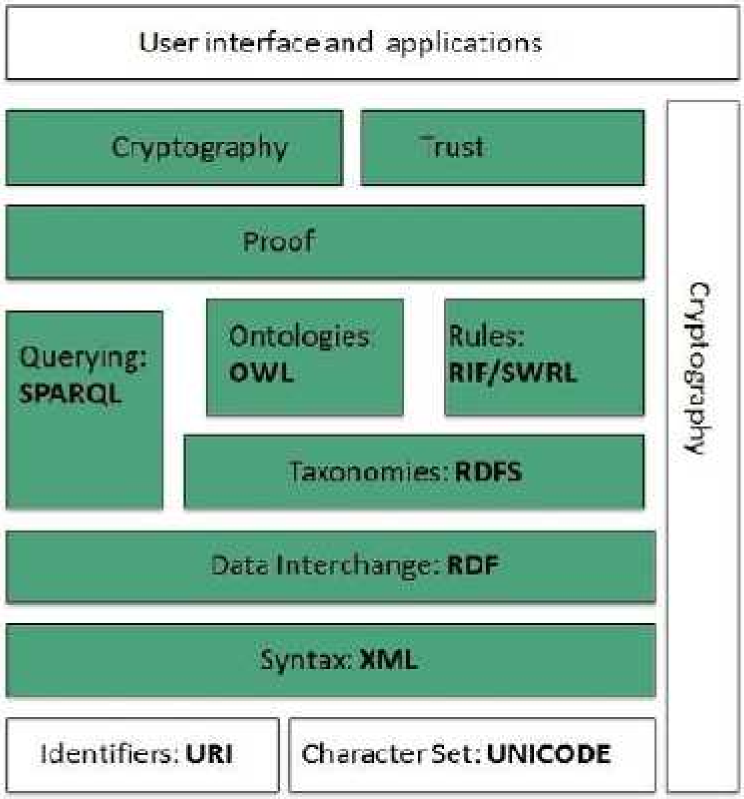
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.

4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.

5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



WWW architecture is divided into several layers as shown in the following diagram:



2. **Electronic Mail (e-mail):** Electronic mail, commonly called email or e-mail, is a method of exchanging digital messages from an author to one or more recipients. E-mail operates across the Internet or other computer networks. An e- mail message consists of three components namely:
   (i) the message header
   (ii) the message envelop, and
   (iii) the message body.

2

COMMUNICATION PROTOCOLS AND SERVERS

Protocols are rules that describe how a client and a server communicate with each other over a network. No single protocol makes the Internet and Web work; rather a number of protocols with unique functions are required. The most commonly used protocols are:

1. Transmission Control/Internet Protocol (TCP/IP)
2. File Transfer Protocol (FTP)
3. Hypertext Transfer Protocol (HTTP)
4. Real Time Streaming Protocol (RTSP)
5. Email Protocol, these includes:

SMTP - Simple Mail Transport Protocol SMTP controls the transfer of e-mail messages on the Internet. SMTP defines the interaction between Internet hosts that participate in forwarding e-mail from a sender to its destination.

POP - Post Office Protocol POP allows you to fetch email that is waiting in a mail server mailbox. POP defines a number of operations for how to access and store email on your server.

IMAP - Internet Message Access Protocol IMAP - Internet Message Access Protocol is an Internet protocol

SERVER

Servers are usually high-performance computers connected to the Internet by high-speed communication lines. Depending on your application, you may deploy it on less-powered machine with less substantial connections. The following are variation of servers:

a) Web server: This is use to store and deliver the elements of web pages.

b) Application server: This is used to run specialised Internet application, such as e-commerce or e-health's engine. It is designed to process requests and deliver dynamic results.

c) Streaming server: This is used to deliver audio or video to the visitors to a site real-time.

d) Mail server: This is used to send and receive e-mail.

e) Name server: This is a specialised server that stores huge directories of web servers. It keeps track of all the registered domain names on the Internet.

f) Secure server: This is a Web server that encrypts data before transmitting it, to prevent unauthorised access. They are commonly used to secure for financial transactions in the Internet.

WEB APPLICATION (WEBAPPS)

Webapps are applications that are accessed with a web browser over a network such as the Internet or an intranet. They are popular because of the ubiquity of the browser as a client (thin client). Similarly, its popularity is equally due to the possibility of updating and maintaining the application without necessarily distributing and installing it on every available client. Webapps or weblications as they are sometimes called are used to implement webmail, online retail sales, online auctions, discussion boards, and weblogs and so on. Web developers often use client-side scripting to add functionality to the webapps by creating an interactive site that does not require page reloading. Webapps generate a series of web pages dynamically in a standard format such as Hypertext Markup Language (HTML) supported by common browsers.

WEB SERVER

A web server can be referred to as either the hardware (the computer) or the software (the computer application) that helps to deliver content that can be accessed through the Internet. The most common use of Web servers is to host Web sites but there are other uses like data storage or for running enterprise applications. The primary function of a web server is to deliver web pages on the request to clients. This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and JavaScript.

Servers are slave programs. They act only when requests are made to them by browsers running on other computers and the Internet. The most commonly used Web Servers are Apache, which has been implemented for variety of computer platforms, and Microsoft's Internet Information Server (IIS), which runs under windows operating systems.

- **Apache HTTP Server** This is most popular web server. It is a free software/open source like Linux, PHP and MySQL. Apache runs on Unix, Linux, MS Windows, Novell Netware and some other platforms. Apache serves over 68 per cent of websites and serves both static and dynamic contents on the web in a very reliable and secure manner. It is an excellent server because it is both fast and reliable. Furthermore, it is open-sources software, which means it is free and managed by a large team of volunteers, a process that efficiently and effectively maintains the systems.
- **The Internet information services (IIS)** The Internet information services (IIS) is a server or system based services for servers using Microsoft Windows operation system. It is a major component of the Microsoft Server operating system and particularly, a component of its Active Server Pages (ASPs). IIS is recommended if both the middleware (ASP) and the database Server (SQL Server) are Microsoft products. Though the Apache server may be installed on Windows platforms, it is not the most popular server on those systems. IIS remains the most popular on Windows platform because it is supplied as part of Windows and because it is a reasonable good server.

DIFFERENCES BETWEEN WEB CLIENT AND WEB SERVERS:

Web Client

- Connected to the Internet when needed
- Usually runs Web browser(client) software such as Internet Explorer or Netscape
- Uses HTTP
- Request Web pages from a server
- Receives Web Pages and files from as server

Web Server

- Continually connected to the Internet
- Runs Web server software (such as Apache or Internet Information Services (IIS)

- Uses HTTP
- Receives a request for the Web page
- Responds to the request and transmits the status code, Web page, and associated files

3

UNIFORM RESOURCE LOCATORS (URLS).

A URL is a formatted text string used by Web browsers, email clients and other software to identify a network resource on the Internet. Network resources are files that can be plain Web pages, other text documents, graphics, or programs.

URLs are used by Web browser to locate and access information on the WWW. A URL is also known as a Web address. Think of URLs as a postal addresses for the Internet.

The first part of the URL parts is known as the *protocol*. This is almost always *http://*, which is short for Hypertext Transfer Protocol. Some URLs start with a different protocol, such as *ftp://* or *news://*. If you're accessing a document on your local machine instead of on the Web, the URL will begin with *file://*.

The second part of the URL is known as the *domain name*. If you've used e-mail on the Internet, you're probably already familiar with domains. The domain represents the name of the server that you're connecting to. A domain name, to put it simple, is your address on the World Wide Web. This is where you put up your website and it is what internet users will type in their address bar in order to locate your site while online. Your domain name should be short, simple, and easy to remember. But, one must keep in mind that domain names are only available for one individual or business. This is to maintain uniqueness and to avoid confusion among the millions of websites and internet users.

A most common example of a domain name is www yahoo.com. The first part, the www identifies the server name of the domain. Yahoo, the second element, is the name of the company, individual or organization; and the suffix .com is the domain name extension, which identifies the purpose of the website.
Another example of a domain name is www nasa.gov. This is the NASA website, and since it is a government office, it uses the extension dot gov. Users need to bear in mind that the domain name extensions are there for a purpose. It indicates the purpose why the website exists.

The third part of the URL is called the *directory path*. This is the specific area on the server where the item resides. Directory paths on Web servers work a lot like they do on your desktop computer. To locate a particular file on a server, you need to indicate its directory path first.

The fourth part of the URL is called the *document file name*. This indicates the specific file being accessed. This is usually an HTML file, but it can also be an image, sound, or another file.

ABSOLUTE VS. RELATIVE URLS

Full URLs featuring all substrings are called *absolute* URLs. In some cases, such as within Web pages, URLs can contain only the one location element. These are called *relative* URLs. Relative URLs are used for efficiency by Web servers and a few other programs when they already know the correct URL protocol and host.

4

WEB BROWSERS, WEBSITES AND WEB PAGES

Browser A Web browser is a software program that interprets the coding language of the World Wide Web in graphic form, displaying the translation rather than the coding. A browser acts as an interface between the user and the inner working of the web. The browser software such as the Internet Explorer, Google Chrome, Firefox, Mozilla, Safari, and so on, interpret HTML codes and presents the information contained in the web pages in a readable format on the users' computer. A browser does not display HTML tags. Browsers function as client programs by contacting the web server and sending the request for information received to the users' computes.

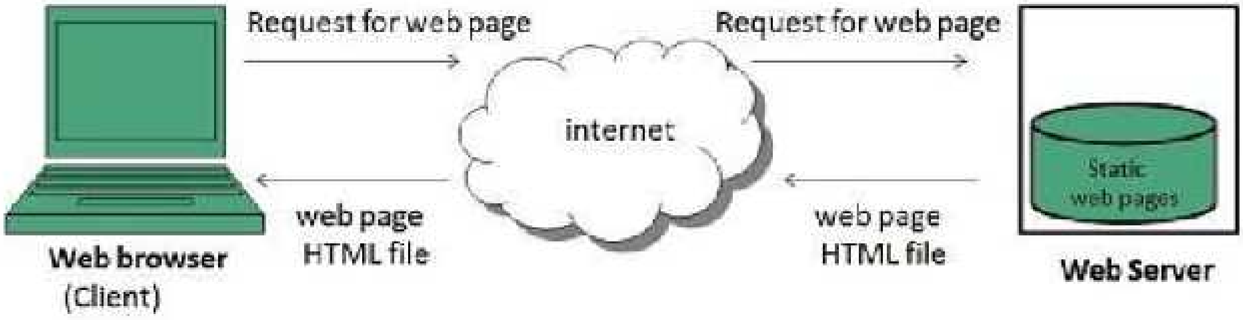
web Browser is an application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into the address bar. Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page. Earlier the web browsers were text-based while nowadays graphical-based or voice-based web browsers are also available. Following are the most common web browser available today:

WEBSITE

A Website is a location on web and is hosted on a web server. It is a set of related web pages. It is accessed using Internet address known as Uniform Resource Locator.

Static Websites

Static websites are also known as flat or stationary websites. They are loaded on the client's browser as exactly they are stored on the web server. Such websites contain only static information. User can only read the information but can't do any modification or interact with the information. Static websites are created using only HTML. Static websites are only used when the information is no more required to be modified.



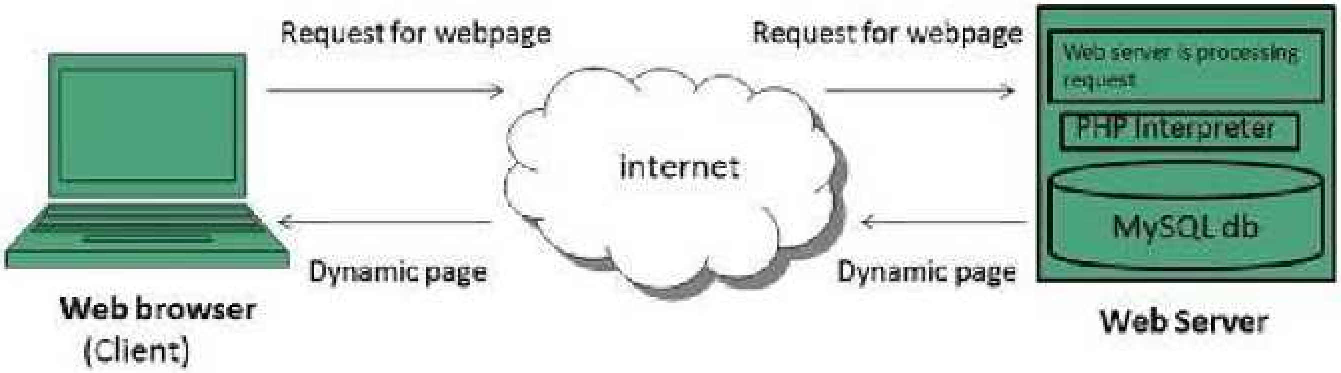
Dynamic websites shows different information at different point of time. It is possible to change a portion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

Server-side dynamic web page

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

Client-side dynamic web page

It is processed using client side scripting such as javascript. And then passed in to **Document Object Model *DOM*.**



5

UNDERSTANDING HYPERTEXT MARKUP LANGUAGE (HTML)

HTML stands for Hypertext Mark-up Language. It is the language for building Web pages and consists of standardized codes or "tags" that are used to define the structure of information on the Web page. Web pages come in many different varieties. In their simplest form, they contain static information, which is made up of simple texts.

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the World Wide Web Consortium, W3C, the organization charged with designing and maintaining the language.

- ***HyperText*** is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is ***hyper*** just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
- ***Markup*** is what HTML tags do to the text inside them. They mark it as a certain type of text (*italicised* text, for example).
- HTML is a *Language*, as it has code-words and syntax like any other language.

HTML TAGS

HTML consists of standardized "tags" that are used to define the structure of information on the Web pages. The decision about the structure of the text is made by the browser based on the tags, which are marks that are embedded into the text.

HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<html>** has its closing tag**</html>** and **<body>** tag has its closing tag **</body>** tag etc.

Tags are generally used to specify "mark-up" regions of HTML documents for the web browser to interpret. Tags are composed of the name of the element, surrounded by angle brackets. An end tag also has a slash after the opening angle bracket, to distinguish it from the start tag. To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML4

Tags can be**: Empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

[**Tag Types and Attributes and Elements**]

The structure of WEB Page The basic structure for all HTML documents is simple and should include the following minimum elements or tags known as the **basic HTML document**
In its simplest form, following is an example of an HTML document:

```
<! DOCTYPE html>
<html>
        <head>
                <title>This is document title</title>
        </head>
        <body>
                <h1>This is a heading</h1>
                <p>Document content goes here.....</p>
        </body>
</html>
```

- The **<! DOCTYPE>** declaration tag is used by the web browser to understand the version of the HTML used in the document.
- **<html>-**This is the main container for HTML pages
- **<head>-**This is the container for page header information
- **<title>-**The is used for the title of the page
- **<body>-**This is a container of the main body of the page

HOW TO CREATE AND RUN HTML CODES

Creating an HTML document is easy. HTML allows us use only ASCII characters for both the main text and formatting instructions. To begin coding HTML you need a standard text editor. Notepad is readily available on computer systems. You could also use Dreamweaver or other text editors if installed. It is not advisable to use a word processor.

Steps 1. Launch Notepad application on your computer
2. Type in your HTML codes
3. Save the document onto a location in your computer drive with a name and the extension “html” (for example firstnoun.html)
4. Click on the filename created.

THINGS YOU CAN DO WITH HTML

1. You can create a personal home page and leave your mark on the World Wide Web.
2. You can create a page for your company to advertise and promote products and services.
3. You can build a catalog on the World Wide Web, complete with product descriptions and photographs. You can even incorporate fill-in order forms so that your customers can order products from you on line.
4. You can create a searchable phone directory for your company or organization.
5. You can create a newsletter on the Web, with pictures and sounds. Using some of the advanced HTML tricks explained in this book, you can format the newsletter to give it a slick, professional appearance.

HTML – HEADER

We have learnt that a typical HTML document will have following structure:

```
Document declaration tag
<html>
<head>
Document header related tags
</head>
<body>
Document body related tags
</body>
</html>
```

The HTML <title> Tag

The HTML <title> tag is used for specifying the title of the HTML document. Following is an example to give a title to an HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Title Tag Example</title>
</head>
<body>
<p>Hello, World!</p>
</body>
</html>
```

The HTML <meta> Tag

The HTML <meta> tag is used to provide metadata about the HTML document which includes information about page expiry, page author, list of keywords, page description etc.

Following are few of the important usages of <meta> tag inside an HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Meta Tag Example</title>
<!-- Provide list of keywords -->
<meta name="keywords" content="C, C++, Java, PHP, Perl, Python">
<!-- Provide description of the page -->
<meta name="description" content="Simply Easy Learning by Tutorials Point">
<!-- Author information -->
```

```
<meta name="author" content="Tutorials Point">
<!-- Page content type -->
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<!-- Page refreshing delay -->
<meta http-equiv="refresh" content="30">
<!-- Page expiry -->
<meta http-equiv="expires" content="Wed, 21 June 2006 14:25:27 GMT">
<!-- Tag to tell robots not to index the content of a page -->
<meta name="robots" content="noindex, nofollow">
</head>
<body>
<p>Hello, World!</p>
</body>
</html>
```

The HTML <base> Tag

The HTML <base> tag is used for specifying the base URL for all relative URLs in a page, which means all the other URLs will be concatenated into base URL while locating for the given item.

For example, all the given pages and images will be searched after prefixing the given URLs with base URL http://www.tutorialspoint.com/ directory:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Base Tag Example</title>
<base href="http://www.tutorialspoint.com/" />
</head>
<body>
<img src="/images/logo.png" alt="Logo Image"/>
<a href="/html/index.htm" title="HTML Tutorial"/>HTML Tutorial</a>
</body>
</html>
```

The HTML <link> Tag

The HTML <link> tag is used to specify relationships between the current document and external resource. Following is an example to link an external style sheet file available in **css** sub-directory within web root:

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

```
<title>HTML link Tag Example</title>
<base href="http://www.tutorialspoint.com/" />
<link rel="stylesheet" type="text/css" href="/css/style.css">
</head>
<body>
<p>Hello, World!</p>
</body>
</html>
```

The 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.tutorialspoint.com/" />
<style type="text/css">
.myclass{
background-color: #aaa;
padding: 10px;
}
</style>
</head>
<body>
<p class="myclass">Hello, World!</p>
</body>
</html>
```

The HTML <script> Tag

The HTML <script> tag is used to include either external script file or to define internal script for the HTML document. Following is an example where we are using JavaScript to define a simple JavaScript function:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML script Tag Example</title>
<base href="http://www.tutorialspoint.com/" />
<script type="text/JavaScript">
```

```
function Hello(){
alert("Hello, World");
}
</script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="OK" />
</body>
</html>
```

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>**. While displaying any heading, browser adds one line before and one line after that heading.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Heading Example</title>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
```

Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag as shown below in the example:

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph Example</title>
</head>
```

```
<body>
<p>Here is a first paragraph of text.</p>
<p>Here is a second paragraph of text.</p>
<p>Here is a third paragraph of text.</p>
</body>
</html>
```

Line Break Tag

Whenever you use the **<br />** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them. The <br /> tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid in XHTML.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Line Break Example</title>
</head>
<body>
<p>Hello<br />
You delivered your assignment on time.<br />
Thanks<br />
Mahnaz</p>
</body>
</html>
```

Centering Content

You can use **<center>** tag to put any content in the center of the page or any table cell.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Centering Content Example</title>
</head>
<body>
<p>This text is not in the center.</p>
<center>
<p>This text is in the center.</p>
</center>
```

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

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly. For example, you may want to give a line between two paragraphs as in the given example below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Horizontal Line Example</title>
</head>
<body>
<p>This is paragraph one and should be on top</p>
<hr />
<p>This is paragraph two and should be at bottom</p>
</body>
</html>
```

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The **<hr />** element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use **<hr>** it is not valid in XHTML

Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag **<pre>**.

Any text between the opening **<pre>** tag and the closing **</pre>** tag will preserve the formatting of the source document.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Preserve Formatting Example</title>
</head>
<body>
<pre>
function testFunction( strText ){
```

```
alert (strText)
}
</pre>
</body>
</html>
```

Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines:

An example of this technique appears in the movie "12 Angry Men."

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity **&nbsp;** instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code:

Example

```
<! DOCTYPE html>
<html>
<head>
<title>Nonbreaking Spaces Example</title>
</head>
<body>
<p>An example of this technique appears in the movie "12 &nbsp; Angry &nbsp; Men."</p>
</body>
```

HTML – FORMATTING

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

Bold Text

Anything that appears within **<b>...</b>** element, is displayed in bold as shown below:

Example

```
<! DOCTYPE html>
<html>
<head>
<title>Bold Text Example</title>
</head>
<body>
<p>The following word uses a <b>bold</b> typeface. </p>
</body>
</html>
```

Italic Text

Anything that appears within **<i>...</i>** element is displayed in italicized as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Italic Text Example</title>
</head>
<body>
<p>The following word uses a <i>italicized</i> typeface.</p>
</body>
</html>
```

Underlined Text

Anything that appears within **<u>...</u>** element, is displayed with underline as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Underlined Text Example</title>
</head>
<body>
<p>The following word uses a <u>underlined</u> typeface.</p>
</body>
</html>
```

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Strike Text Example</title>
</head>
<body>
<p>The following word uses a <strike>strikethrough</strike> typeface.</p>
</body>
</html>
```

Monospaced Font

The content of a **<tt>...</tt>** element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Monospaced Font Example</title>
</head>
<body>
<p>The following word uses a <tt>monospaced</tt> typeface.</p>
</body>
</html>
```

Superscript Text

The content of a **<sup>...</sup>** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Superscript Text Example</title>
</head>
<body>
<p>The following word uses a <sup>superscript</sup> typeface.</p>
</body>
</html>
```

Subscript Text

The content of a **<sub>...</sub>** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Subscript Text Example</title>
</head>
<body>
<p>The following word uses a <sub>subscript</sub> typeface.</p>
```

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

Computer Code

Any programming code to appear on a Web page should be placed inside **<code>...</code>**tags. Usually the content of the <code> element is presented in a monospaced font, just like the code in most programming books.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Computer Code Example</title>
</head>
<body>
<p>Regular text. <code>This is code.</code> Regular text.</p>
</body>
</html>
```

Address Text

The **<address>...</address>** element is used to contain any address.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Address Example</title>
</head>
<body>
<address>388A, Road No 22, Jubilee Hills - Hyderabad</address>
</body>
</html>
```

HTML – Comments

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code and increases code readability. HTML comments are placed in between **<!-- ... -->** tags. So, any content placed with-in <!-- ... --> tags will be treated as comment and will be completely ignored by the browser. Never add a space between the left angle bracket and the exclamation mark.

Comments do not nest which means a comment cannot be put inside another comment. Second the double-dash sequence "--" may not appear inside a comment except as part of the closing --> tag. You must also make sure that there are no spaces in the start-of-comment string.

Example

```
<!DOCTYPE html>
<html>
<head> <!-- Document Header Starts -->
<title>This is document title</title>
</head> <!-- Document Header Ends -->
<body>
<p>Document content goes here.....</p>
</body>
</html>
```

Multiline Comments

So far we have seen single line comments, but HTML supports multi-line comments as well.

You can comment multiple lines by the special beginning tag <!-- and ending tag --> placed before the first line and end of the last line as shown in the given example below.

Example

```
<!DOCTYPE html><html>
<head>
<title>Multiline Comments</title>
</head>
<body>
<!--
This is a multiline comment and it can
span through as many as lines you like.
-->
<p>Document content goes here.....</p>
</body>
</html>
```

HTML – MARQUEES

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML <marquees> tag.

Note: The HTML <marquee> tag may not be supported by various browsers so it is not recommended to rely on this tag, instead you can use JavaScript and CSS to create such effects.

Syntax

A simple syntax to use HTML <marquee> tag is as follows:

```
<marquee attribute_name="attribute_value"....more attributes>
One or more lines or text message or image
</marquee>
```

The <marquee> Tag Attributes

Following is the list of important attributes which can be used with <marquee> tag.

Attribute	**Description**
width:	This specifies the width of the marquee. This can be a value like 10 or 20% etc.
height:	This specifies the height of the marquee. This can be a value like 10 or 20% etc.
direction:	This specifies the direction in which marquee should scroll. This can be a value like up, down, left or right.
behavior:	This specifies the type of scrolling of the marquee. This can have a value like scroll, slide and alternate.
scrolldelay:	This specifies how long to delay between each jump. This will have a value like 10 etc.
scrollamount:	This specifies the speed of marquee text. This can have a value like 10 etc.
loop:	This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
bgcolor:	This specifies background color in terms of color name or color hex value.
hspace:	This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
vspace:	This specifies vertical space around the marquee. This can be a value like 10 or 20% etc.

Below are few examples to demonstrate the usage of marquee tag.

Examples - 1

```
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee>This is basic example of marquee</marquee>
</body>
</html>
```

Examples - 2

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

```
<title>HTML marquee Tag</title>
</head>
<body>
<marquee width="50%">This example will take only 50% width</marquee>
</body>
</html>
```

Examples - 3

```
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee direction="right">This text will scroll from left to right</marquee>
</body>
</html>
```

Examples - 4

```
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee direction="up">This text will scroll from bottom to up</marquee>
</body>
</html>
```

This will produce the following result:

HTML – FONTS

Fonts play a very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML **<font>** tag to add style, size, and color to the text on your website. You can use a **<basefont>** tag to set all of your text to the same size, face, and color.

The font tag is having three attributes called **size, color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the <font> tag. The text that follows will remain changed until you close with the </font> tag. You can change one or all of the font attributes within one <font> tag.

Set Font Size

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Setting Font Size</title>
</head>
<body>
<font size="1">Font size="1"</font><br />
<font size="2">Font size="2"</font><br />
<font size="3">Font size="3"</font><br />
<font size="4">Font size="4"</font><br />
<font size="5">Font size="5"</font><br />
<font size="6">Font size="6"</font><br />
<font size="7">Font size="7"</font>
</body>
</html>
```

Setting Font Face

You can set font face using *face* attribute but be aware that if the user viewing the page doesn't have the font installed, they will not be able to see it. Instead user will see the default font face applicable to the user's computer.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Font Face</title>
</head>
<body>
<font face="Times New Roman" size="5">Times New Roman</font><br />
<font face="Verdana" size="5">Verdana</font><br />
```

```
<font face="Comic sans MS" size="5">Comic Sans MS</font><br />
<font face="WildWest" size="5">WildWest</font><br />
<font face="Bedrock" size="5">Bedrock</font><br />
</body>
</html>
```

Setting Font Color

You can set any font color you like using *color* attribute. You can specify the color that you want by either the color name or hexadecimal code for that color.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Setting Font Color</title>
</head>
<body>
<font color="#FF00FF">This text is in pink</font><br />
<font color="red">This text is red</font>
</body>
</html>
```

HTML – BACKGROUNDS

By default, your webpage background is white in color. You may not like it, but no worries. HTML provides you following two good ways to decorate your webpage background.

- Html Background with Colors
- Html Background with Images

Now let's see both the approaches one by one using appropriate examples.

Html Background with Colors

The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds. Following is the syntax to use bgcolor attribute with any HTML tag.

```
<tagname bgcolor="color_value"...>
```

This color_value can be given in any of the following formats:

-
  ```
  <!-- Format 1 - Use color name -->
  <table bgcolor="lime" >
  ```

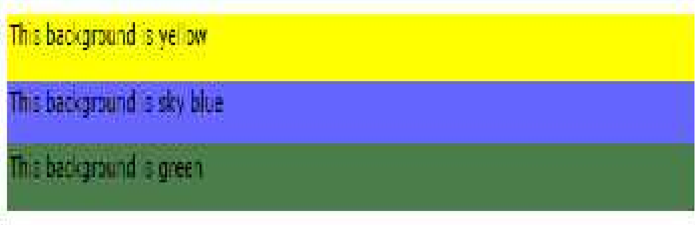
-
  ```
  <!-- Format 2 - Use hex value -->
  <table bgcolor="#f1f1f1" >
  ```

- <!-- Format 3 - Use color value in RGB terms -->
  <table bgcolor="rgb(0,0,120)" >

Example

Here are the examples to set background of an HTML tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Colors</title>
</head>
<body>
<!-- Format 1 - Use color name -->
        <table bgcolor="yellow" width="100%">
<tr><td>
This background is yellow
</td></tr>
</table>
<!-- Format 2 - Use hex value -->
        <table bgcolor="#6666FF" width="100%">
<tr><td>
This background is sky blue
</td></tr>
</table>
<!-- Format 3 - Use color value in RGB terms -->
        <table bgcolor="rgb(255,0,255)" width="100%">
<tr><td>
This background is green
</td></tr>
</table>
</body>
</html>
```



This background is yellow

This background is sky blue

This background is green

Html Background with Images

The **background** attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table. Following is the syntax to use background attribute with any HTML tag.

Note: The *background* attribute is deprecated and it is recommended to use Style Sheet for background setting.

<tagname background="Image URL"...>

The most frequently used image formats are JPEG, GIF and PNG images.

Example

Here are the examples to set background images of a table.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Images</title>
</head>
<body>
<!-- Set table background -->
<table background="/images/html.gif" width="100%" height="100">
<tr><td>
This background is filled up with HTML image.
</td></tr>
</table>
</body>
</html>
```

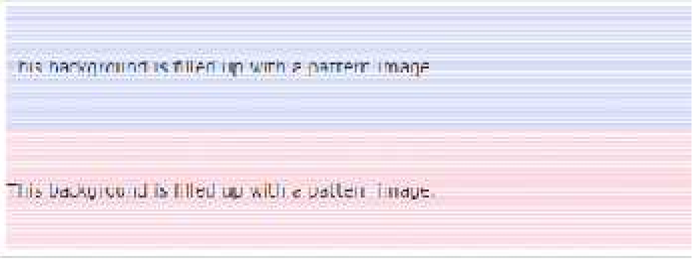
Patterned & Transparent Backgrounds

You might have seen many pattern or transparent backgrounds on various websites. This simply can be achieved by using patterned image or transparent image in the background.

It is suggested that while creating patterns or transparent GIF or PNG images, use the smallest dimensions possible even as small as 1x1 to avoid slow loading.

Example

Here are the examples to set background pattern of a table:



This background is filled up with a pattern image

This background is filled up with a pattern image.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Images</title>
</head>
<body>
<!-- Set a table background using pattern -->
<table background="/images/pattern1.gif" width="100%" height="100">
<tr><td>
This background is filled up with a pattern image.
</td></tr>
</table>
<!-- Another example on table background using pattern -->
<table background="/images/pattern2.gif" width="100%" height="100">
<tr><td>
```

```
This background is filled up with a pattern image.
</td></tr>
</table>
</body>
</html>
```

HTML – IMAGES

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive. The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

Insert Image

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

```
<img src="Image URL" ... attributes-list/>
```

The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory:

```
<!DOCTYPE html>
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>
<img src="images/test.png" alt="Test Image" /> </body>
</html>
```

Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Width and Height</title>
</head>
<body>
```

```
<p>Setting image width and height</p>
<img src="test.png" alt="Test Image" width="150" height="100"/>
</body>
</html>
```

Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Border</title>
</head>
<body>
<p>Setting image Border</p>
<img src="test.png" alt="Test Image" border="3"/>
</body>
</html>
```

Set Image Alignment

By default, image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Alignment</title>
</head>
<body>
<p>Setting image Alignment</p>
<img src="test.png" alt="Test Image" border="3" align="right"/>
</body>
</html>
```

HTML – COLORS

Colors are very important to give a good look and feel to your website. You can specify colors on page level using <body> tag or you can set colors for individual tags using **bgcolor** attribute.

The <body> tag has following attributes which can be used to set different colors:

- **bgcolor -** sets a color for the background of the page.
- **text -** sets a color for the body text.
- **alink -** sets a color for active links or selected links.
- **link -** sets a color for linked text.
- **vlink -** sets a color for *visited links* - that is, for linked text that you have already clicked on.

HTML Color Coding Methods

There are following three different methods to set colors in your web page:

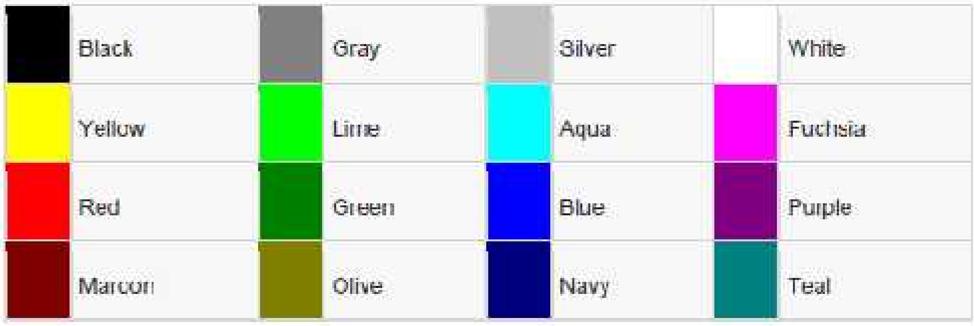
1. **Color names -** You can specify color names directly like green, blue or red.
2. **Hex codes -** A six-digit code representing the amount of red, green, and blue that makes up the color.
3. **Color decimal or percentage values -** This value is specified using the rgb() property.

HTML Colors - Color Names

You can specify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by major browsers.

W3C Standard 16 Colors

Here is the list of W3C Standard 16 Colors names and it is recommended to use them.



Black	Gray	Silver	White
Yellow	Lime	Aqua	Fuchsia
Red	Green	Blue	Purple
Maroon	Olive	Navy	Teal

Example

Here are the examples to set background of an HTML tag by color name:

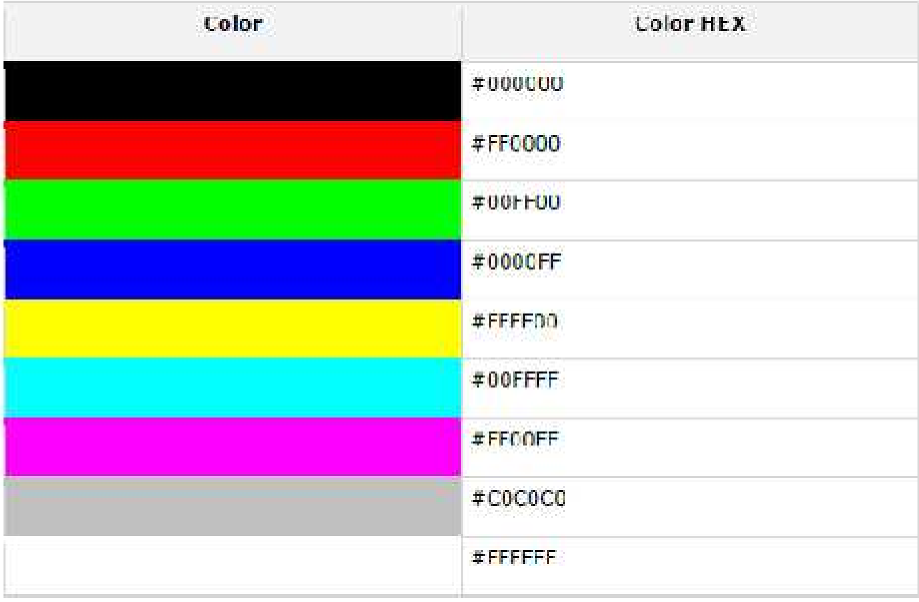
```
<!DOCTYPE html>
<html>
<head>
<title>HTML Colors by Name</title>
</head>
<body text="blue" bgcolor="green">
<p>Use different color names for for body and table and see the result.</p>
```

```
<table bgcolor="black">
<tr>
<td>
<font color="white">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```

<u>HTML Colors - Hex Codes</u>

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).
A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Paintshop Pro or MS Paint.

Each hexadecimal code will be preceded by a pound or hash sign #. Following is a list of few colors using hexadecimal notation.



Color	Color HEX
⬛	#000000
🟥	#FF0000
🟩	#00FF00
🟦	#0000FF
🟨	#FFFF00
🟦	#00FFFF
🟪	#FF00FF
▨	#C0C0C0
⬜	#FFFFFF

Example

Here are the examples to set background of an HTML tag by color code in hexadecimal:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Colors by Hex</title>
</head>
```

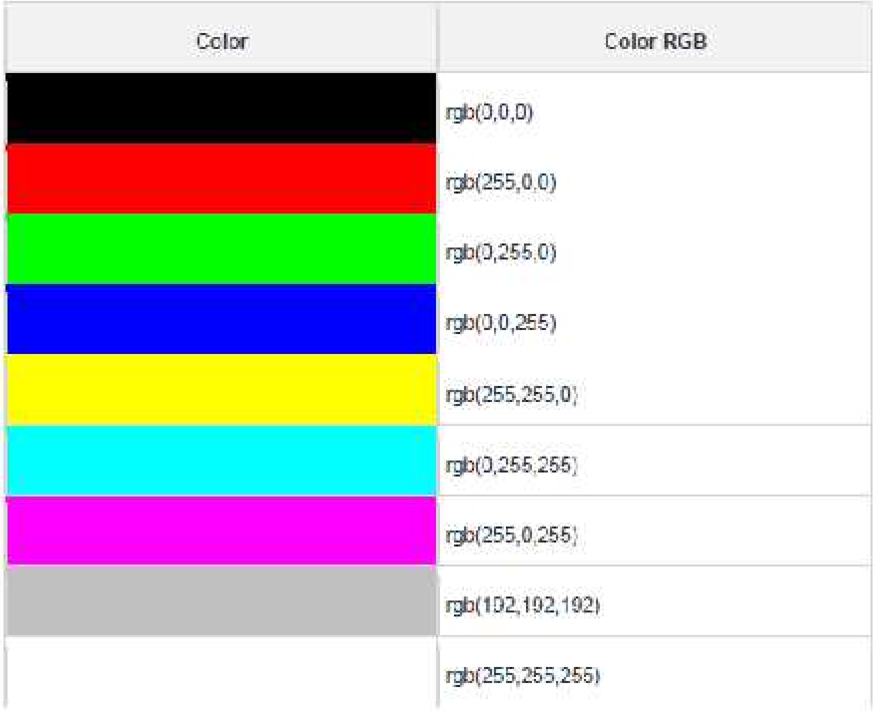
```
<body text="#0000FF" bgcolor="#00FF00">
<p>Use different color hexa for for body and table and see the result.</p>
<table bgcolor="#000000">
<tr>
<td>
<font color="#FFFFFF">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```

HTML Colors - RGB Values

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

Note: All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is a list to show few colors using RGB values.



Color	Color RGB
⬛	rgb(0,0,0)
🟥	rgb(255,0,0)
🟩	rgb(0,255,0)
🟦	rgb(0,0,255)
🟨	rgb(255,255,0)
🟦	rgb(0,255,255)
🟪	rgb(255,0,255)
▨	rgb(192,192,192)
⬜	rgb(255,255,255)

Example

Here are the examples to set background of an HTML tag by color code using rgb() values:

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

```
<head>
<title>HTML Colors by RGB code</title>
</head>
<body text="rgb(0,0,255)" bgcolor="rgb(0,255,0)">
<p>Use different color code for for body and table and see the result.</p>
<table bgcolor="rgb(0,0,0)">
<tr>
<td>
<font color="rgb(255,255,255)">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF

CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

HTML – LISTS

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

- **<ul>** - An unordered list. This will list items using plain bullets.
- **<ol>** - An ordered list. This will use different schemes of numbers to list your items.
- **<dl>** - A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **<ul>** tag. Each item in the list is marked with a bullet.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

The type Attribute

You can use **type** attribute for <ul> tag to specify the type of bullet you like. By default, it is a disc. Following are the possible options:

```
<ul type="square">
<ul type="disc">
<ul type="circle">
```

Example

Following is an example where we used <ul type="square">

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul type="square">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

Example

Following is an example where we used <ul type="disc">:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul type="disc">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

Example

Following is an example where we used <ul type="circle">:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul type="circle">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

The type Attribute

You can use **type** attribute for <ol> tag to specify the type of numbering you like. By default, it is a number. Following are the possible options:

- <ol type="1"> - Default-Case Numerals.
- <ol type="I"> - Upper-Case Numerals.
- <ol type="i"> - Lower-Case Numerals.
- <ol type="a"> - Lower-Case Letters.
- <ol type="A"> - Upper-Case Letters.

Example

Following is an example where we used **<ol type="I">**

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="I">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Example

Following is an example where we used **<ol type="i">**

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="i">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
```

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

Example

Following is an example where we used **<ol type="A">**

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="A">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Example

Following is an example where we used **<ol type="a">**

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="a">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

HTML – TEXT LINKS

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks.

Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

Linking Documents

A link is specified using HTML tag <a>. This tag is called **anchor tag** and anything between the opening <a> tag and the closing </a> tag becomes part of the link and a user can click that part to reach to the linked document. Following is the simple syntax to use <a> tag.

```
<a href="Document URL" ... attributes-list>Link Text</a>
```

Example

Let's try following example which links http://www.tutorialspoint.com at your page:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.tutorialspoint.com" target="_self">Tutorials Point</a>
</body>
</html>
```

The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are the possible options:

Option	**Description**
_blank	Opens the linked document in a new window or tab.
_self	Opens the linked document in the same frame.
_parent	Opens the linked document in the parent frame.
_top	Opens the linked document in the full body of the window.
Targetframe	Opens the linked document in a named targetframe.

Example

Try following example to understand basic difference in few options given for target attribute.

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

```
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body>
<p>Click any of the following links</p>
<a href="/html/index.htm" target="_blank">Opens in New</a> |
<a href="/html/index.htm" target="_self">Opens in Self</a> |
<a href="/html/index.htm" target="_parent">Opens in Parent</a> |
<a href="/html/index.htm" target="_top">Opens in Body</a>
</body>
</html>
```

Use of Base Path

When you link HTML documents related to the same website, it is not required to give a complete URL for every link. You can get rid of it if you use **<base>** tag in your HTML document header. This tag is used to give a base path for all the links. So your browser will concatenate given relative path to this base path and will make a complete URL.

Example

Following example makes use of <base> tag to specify base URL and later we can use relative path to all the links instead of giving complete URL for every link.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body>
<p>Click following link</p>
<a href="/html/index.htm" target="_blank">HTML Tutorial</a>
</body>
</html>
```

Linking to a Page Section

You can create a link to a particular section of a given webpage by using **name** attribute. This is a two-step process.

First create a link to the place where you want to reach with-in a webpage and name it using <a...> tag as follows:

```
<h1>HTML Text Links <a name="top"></a></h1>
```

Second step is to create a hyperlink to link the document and place where you want to reach:

```
<a href="/html/html_text_links.htm#top">Go to the Top</a>
```

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.

Setting Link Colors

You can set colors of your links, active links and visited links using **link**, **alink** and **vlink** attributes of <body> tag.

Example

Save the following in test.htm and open it in any web browser to see how **link**, **alink** and **vlink** attributes work.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body alink="#54A250" link="#040404" vlink="#F40633">
<p>Click following link</p>
<a href="/html/index.htm" target="_blank" >HTML Tutorial</a>
</body>
</html>
```

Download Links

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple; you just need to give complete URL of the downloadable file as follows:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<a href="http://www.tutorialspoint.com/page.pdf">Download PDF File</a>
</body>
</html>
```

HTML – IMAGE LINKS

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now, we will learn how to use images to create hyperlinks.

Example

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below:

```
<!DOCTYPE html>
<html>
<head>
<title>Image Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.tutorialspoint.com" target="_self">
<img src="/images/logo.png" alt="Tutorials Point" border="0"/>
</a>
</body>
</html>
```

HTML Email Tag

HTML **<a>** tag provides you option to specify an email address to send an email. While using <a> tag as an email tag, you will use **mailto: email address** along with *href* attribute. Following is the syntax of using **mailto** instead of using http.

```
<a href= "mailto: abc@example.com">Send Email</a>
```

This code will generate the following link which you can use to send email.

Send Email

Now, if a user clicks this link, it launches one Email Client (like Lotus Notes, Outlook Express etc.) installed on your user's computer. There is another risk to use this option to send email because if user do not have email client installed on their computer then it would not be possible to send email.

Default Settings

You can specify a default *email subject* and *email body* along with your email address. Following is the example to use default subject and body.

```
<a href="mailto:abc@example.com?subject=Feedback&body=Message">
Send Feedback
</a>
```

The <div> tag

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now <div> tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page using <div> tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

Example

Following is a simple example of <div> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <div> tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML div Tag</title>
</head>
<body>
<!-- First group of tags -->
<div style="color:red">
<h4>This is first group</h4>
<p>Following is a list of vegetables</p>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</div>
<!-- Second group of tags -->
<div style="color:green">
<h4>This is second group</h4>
<p>Following is a list of fruits</p>
<ul>
<li>Apple</li>
<li>Banana</li>
<li>Mango</li>
<li>Strawberry</li>
</ul>
</div>
</body>
</html>
```

HTML – FORMS

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax:

```
<form action="Script URL" method="GET|POST">
form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	**Description**
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are: application/x-www-form-urlencoded - This is the standard method most forms use in simple scenarios. mutlipart/form-data - This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms:

1. **Single-line text input controls -** This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.

2. **Password input controls -** This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.

3. **Multi-line text input controls -** This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name:

```
<!DOCTYPE html>
<html>
<head>
<title>Text Input Control</title>
</head>
<body>
<form >
First name: <input type="text" name="first_name" />
<br>
Last name: <input type="text" name="last_name" />
</form>
</body>
</html>
```

Attributes

Following is the list of attributes for <input> tag for creating text field.

Attribute	**Description**
type:	Indicates the type of input control and for text input control it will be set to text.
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
value:	This can be used to provide an initial value inside the control.
size:	Allows to specify the width of the text-input control in terms of characters.
maxlength:	Allows to specify the maximum number of characters a user can enter into the text box.

Password Input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password:

```
<!DOCTYPE html>
<html>
<head>
<title>Password Input Control</title>
</head>
<body>
<form >
User ID : <input type="text" name="user_id" />
<br>
Password: <input type="password" name="password" />
</form>
</body>
</html>
```

Attributes

Following is the list of attributes for <input> tag for creating password field.

Attribute	**Description**
type:	Indicates the type of input control and for password input control it will be set to password.
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
value:	This can be used to provide an initial value inside the control.
size:	Allows to specify the width of the text-input control in terms of characters.
maxlength:	Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description:

```
<!DOCTYPE html>
<html>
<head>
<title>Multiple-Line Input Control</title>
</head>
```

```
<body>
<form>
Description: <br />
<textarea rows="5" cols="50" name="description">
Enter description here...
</textarea>
</form>
</body>
</html>
```

Attributes

Following is the list of attributes for <textarea> tag.

Attribute	**Description**
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows:	Indicates the number of rows of text area box.
cols:	Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox**.

Example

Here is an example HTML code for a form with two checkboxes:

```
<!DOCTYPE html>
<html>
<head>
<title>Checkbox Control</title>
</head>
<body>
<form>
<input type="checkbox" name="maths" value="on"> Maths
<input type="checkbox" name="physics" value="on"> Physics
</form>
</body>
</html>
```

Attributes

Following is the list of attributes for <checkbox> tag.

Attribute	**Description**
type:	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
value:	The value that will be used if the checkbox is selected.
checked:	Set to checked if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPE html>
<html>
<head>
<title>Radio Box Control</title>
</head>
<body>
<form>
<input type="radio" name="subject" value="maths"> Maths
<input type="radio" name="subject" value="physics"> Physics
</form>
</body>
</html>
```

Attributes

Following is the list of attributes for radio button.

Attribute	**Description**
type:	Indicates the type of input control and for checkbox input control it will be set to radio.
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
value:	The value that will be used if the radio box is selected.
checked:	Set to checked if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Select Box Control</title>
</head>
<body>
<form>
<select name="dropdown">
<option value="Maths" selected>Maths</option>
<option value="Physics">Physics</option>
</select>
</form>
</body>
</html>
```

Attributes

Following is the list of important attributes of **<select> tag:**

Attribute	**Description**
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
size:	This can be used to present a scrolling list box.
multiple:	If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of **<option> tag:**

Attribute	**Description**
value:	The value that will be used if an option in the select box box is selected.
selected:	Specifies that this option should be the initially selected value when the page loads.
label:	An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

Example

Here is example HTML code for a form with one file upload box:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
```

```
<form>
<input type="file" name="fileupload" accept="image/*" />
</form>
</body>
</html>
```

Attributes

Following is the list of important attributes of file upload box:

Attribute	Description
name:	Used to give a name to the control which is sent to the server to be recognized and get the value.
accept:	Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input> tag by setting its type attribute to button. The type attribute can take the following values:

Type	Description
submit:	This creates a button that automatically submits a form.
reset:	This creates a button that automatically resets form controls to their initial values.
button:	This creates a button that is used to trigger a client-side script when the user clicks that button.
image:	This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
<input type="button" name="ok" value="OK" />
<input type="image" name="imagebutton" src="/html/images/logo.png" />
</form>
</body>
</html>
```

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<p>This is page 10</p>
<input type="hidden" name="pagename" value="10" />
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
</form>
</body>
</html>
```

HTML – FRAMES

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages:

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back button* might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

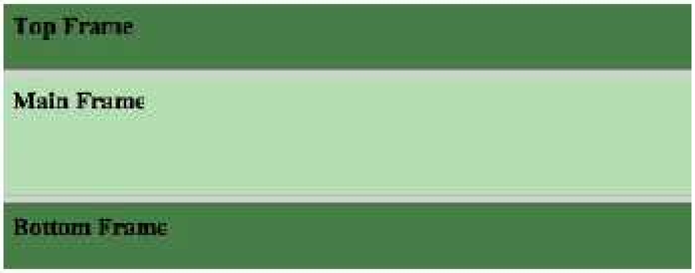
To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Example

Following is the example to create three horizontal frames:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
<frame name="top" src="/html/top_frame.htm" />
<frame name="main" src="/html/main_frame.htm" />
<frame name="bottom" src="/html/bottom_frame.htm" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

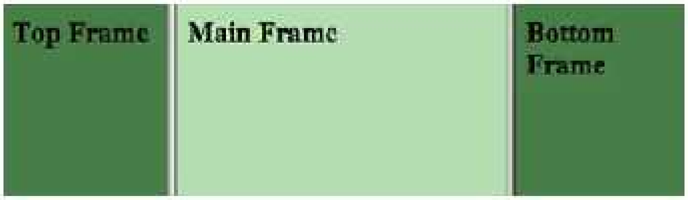
This will produce



Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset cols="25%,50%,25%">
<frame name="left" src="/html/top_frame.htm" />
<frame name="center" src="/html/main_frame.htm" />
<frame name="right" src="/html/bottom_frame.htm" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```



The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag:

Attribute **Description**

cols Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways:

Absolute values in pixels. For example, to create three vertical frames, use cols="100, 500,100".

A percentage of the browser window. For example, to create three vertical frames, use cols="10%, 80%,10%".

Using a wildcard symbol. For example, to create three vertical frames, use cols="10%, *,10%". In this case wildcard takes remainder of the window.

As relative widths of the browser window. For example, to create three vertical frames, use cols="3*,2*,1*". This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.

rows This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use rows="10%, 90%". You can specify the height of each row in the same way as explained above for columns.

border This attribute specifies the width of the border of each frame in pixels. For example, border="5". A value of zero means no border.

frameborder This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder="0" specifies no border.

framespacing This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are the important attributes of <frame> tag:

Attribute **Description**

src This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file available in html directory.

name This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.

frameborder This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).

marginwidth This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".

marginheight This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight="10".

noresize By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".

scrolling This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.

longdesc This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

Browser Support for Frames

If a user is using any old browser or any browser, which does not support frames then <noframes> element should be displayed to the user.

So you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand <frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element.

You can put some nice message for your user having old browsers. For example, *Sorry!! your browser does not support frames.*

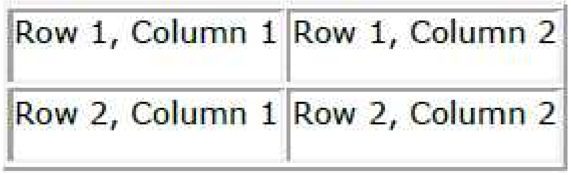
HTML – TABLES

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Tables</title>
</head>
<body>
<table border="1">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```



Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

Here, the **border** is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border, then you can use border="0".

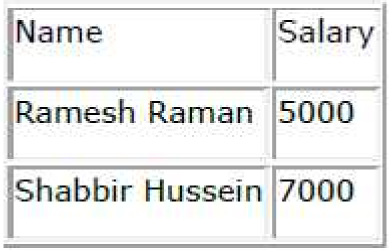
Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row.

Example

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

```
<title>HTML Table Header</title>
</head>
<body>
<table border="1">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```



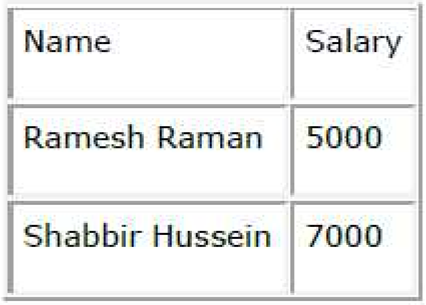
Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

<u>Cellpadding and Cellspacing Attributes</u>

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines the width of the border, while cellpadding represents the distance between cell borders and the content within a cell.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Cellpadding</title>
</head>
<body>
<table border="1" cellpadding="5" cellspacing="5">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
```



Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

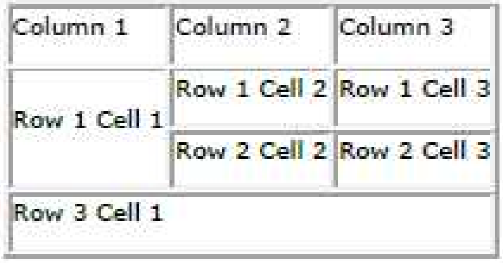
```
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```

Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Colspan/Rowspan</title>
</head>
<body>
<table border="1">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>
```



Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

Tables Backgrounds

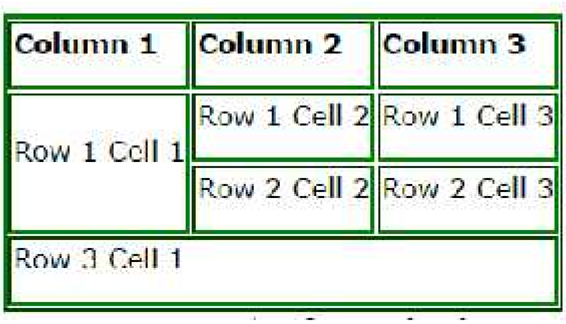
You can set table background using one of the following two ways:

- **bgcolor** attribute - You can set background color for whole table or just for one cell.
- **background** attribute - You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

Example

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



Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

```
<head>
<title>HTML Table Background</title>
</head>
<body>
<table border="1" bordercolor="green" bgcolor="yellow">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>
```

Here is an example of using **background** attribute. Here we will use an image available in /images directory.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Background</title>
</head>
<body>
<table border="1" bordercolor="green" background="/images/test.png">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>
```

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Width/Height</title>
</head>
<body>
<table border="1" width="400" height="150">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Caption</title>
</head>
<body>
<table border="1" width="100%">
<caption>This is the caption</caption>
<tr>
<td>row 1, column 1</td><td>row 1, column 2</td>
</tr>
<tr>
<td>row 2, column 1</td><td>row 2, column 2</td>
</tr>
</table>
```

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

Table Header, Body, and Footer

Tables can be divided into three portions: a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

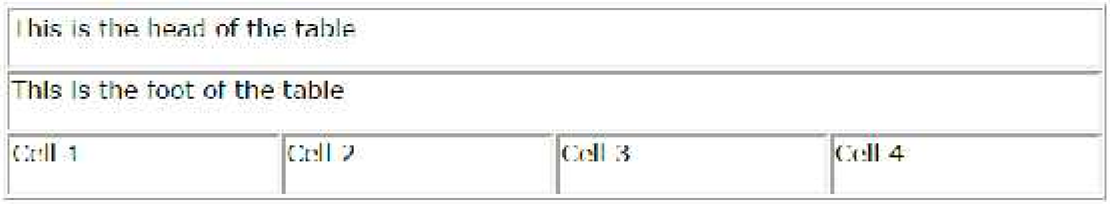
The three elements for separating the head, body, and foot of a table are:

- **<thead> -** to create a separate table header.
- **<tbody> -** to indicate the main body of the table.
- **<tfoot> -** to create a separate table footer.

A table may contain several <tbody> elements to indicate different *pages* or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<thead>
<tr>
<td colspan="4">This is the head of the table</td>
</tr>
</thead>
<tfoot>
<tr>
<td colspan="4">This is the foot of the table</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
<td>Cell 3</td>
<td>Cell 4</td>
</tr>
```



This is the head of the table			
This is the foot of the table			
Cell 1	Cell 2	Cell 3	Cell 4

```
</tbody>
</table>
</body>
</html>
```

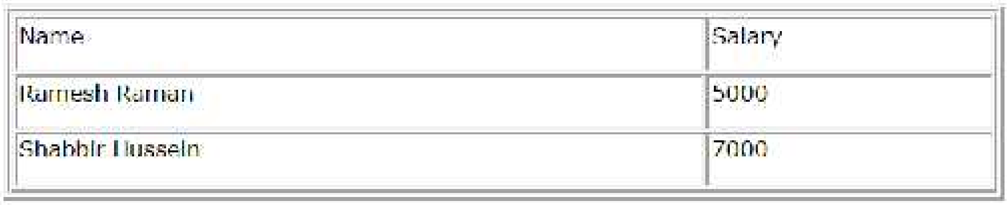
<u>Nested Tables</u>

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

Example

Following is the example of using another table and other tags inside a table cell.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<tr>
<td>
<table border="1" width="100%">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```



Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000