

HTML

Part-1 INTRODUCTION

HTML stands for **Hyper Text Markup Language**, which is the most widely used language on Web to develop web pages. **HTML** was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Why to Learn HTML?

Originally, **HTML** was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

- **Create Web site** - You can create a website or customize an existing web template if you know HTML well.
- **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- **Learn other languages** - Once you understand the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

Applications of HTML

As mentioned before, HTML is one of the most widely used language over the web. I'm going to list few of them here:

- **Web pages development** - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.
- **Internet Navigation** - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
- **Responsive UI** - HTML pages now-a-days works well on all platform, mobile, tabs, desktop or laptops owing to responsive design strategy.
- **Offline support** HTML pages once loaded can be made available offline on the machine without any need of internet.
- **Game development**- HTML5 has native support for rich experience and is now useful in gaming development arena as well.

Audience

This **HTML tutorial** is designed for the aspiring Web Designers and Developers with a need to understand the HTML in enough detail along with its simple overview, and practical examples. This tutorial will give you enough ingredients to start with HTML from where you can take yourself at higher level of expertise.

Prerequisites

Before proceeding with this **tutorial** you should have a basic working knowledge with Windows or Linux operating system, additionally you must be familiar with –

- Experience with any text editor like notepad, notepad++, or Edit plus etc.
- How to create directories and files on your computer.
- How to navigate through different directories.
- How to type content in a file and save them on a computer.

- Understanding about images in different formats like JPEG, PNG format.

Basic HTML Document & it's Structure

Let's start from definition:

HTML stands for **H**ypertext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

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>
```

This is a heading

Document content goes here.....

HTML Tags

As told earlier, 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.

Above example of HTML document uses the following tags –

Sr.No	Tag & Description
1	<!DOCTYPE...> This tag defines the document type and HTML version.
2	<html> This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
3	<head> This tag represents the document's header which can keep other HTML tags like <title> , <link> etc.
4	<title> The <title> tag is used inside the <head> tag to mention the document title.
5	<body> This tag represents the document's body which keeps other HTML tags like <h1> , <div> , <p> etc.
6	<h1> This tag represents the heading.

This tag represents a paragraph.

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration –

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing <!DOCTYPE...> tag along with other HTML tags.

HTML - Basic Tags

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
 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.

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

```
<title>Line Break Example</title>
</head>
<body>
  <p>Hello<br />
  You delivered your assignment ontime.<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.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Centring 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>
```

This will produce following result –
This text is not in the center.

This text is in the center.

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 –

```
<!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>
```

This will produce the following result –
This is paragraph one and should be on top

This is paragraph two and should be at bottom

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.

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.

```
<html>
  <head>
    <title>Preserve Formatting Example</title>
```

```

</head>
<body>
  <pre>
    function testFunction( strText ){
      alert (strText)
    }
  </pre>
</body>
</html>

```

This will produce the following result –

```

function testFunction( strText ){
  alert (strText)
}

```

Try using the same code without keeping it inside `<pre>...</pre>` tags

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 ` `; 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 –

```

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

```

This will produce the following result –

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

HTML - Elements

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags –

Start Tag	Content	End Tag
<code><p></code>	This is paragraph content.	<code></p></code>
<code><h1></code>	This is heading content.	<code></h1></code>
<code><div></code>	This is division content.	<code></div></code>
<code>
</code>		

So here `<p>...</p>` is an HTML element, `<h1>...</h1>` is another HTML element. There are some HTML elements which don't need to be closed, such as `<img...>`, `<hr>` and `
` elements. These are known as **void elements**.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

An HTML element is defined by a *starting tag*. If the element contains other content, it ends with a *closing tag*.

For example, `<p>` is starting tag of a paragraph and `</p>` is closing tag of the same paragraph but `<p>This is paragraph</p>` is a paragraph element.

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 `...` element, is displayed in bold as shown below –

```
<body>
  <p>The following word uses a <b>bold</b> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a **bold** typeface.

Italic Text

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

```
<body>
  <p>The following word uses an <i>italicized</i> typeface.</p>
</body>
```

This will produce the following result –
The following word uses an *italicized* typeface.

Underlined Text

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

```
<body>
  <p>The following word uses an <u>underlined</u> typeface.</p>
</body>
```

This will produce the following result –
The following word uses an underlined typeface.

Strike Text

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

```
<body>
  <p>The following word uses a <strike>strikethrough</strike> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a ~~strikethrough~~ typeface.

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.

```
<body>
  <p>The following word uses a <tt>monospaced</tt> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a monospaced typeface.

Superscript Text

The content of a `^{...}` 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.

```
<body>
  <p>The following word uses a <sup>superscript</sup> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a ^{superscript} typeface.

Subscript Text

The content of a `_{...}` 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.

```
<body>
  <p>The following word uses a <sub>subscript</sub> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a _{subscript} typeface.

Inserted Text

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

```
<body>
  <p>I want to drink <del>cola</del> <ins>wine</ins></p>
</body>
```

This will produce the following result –
I want to drink ~~cola~~ wine

Deleted Text

Anything that appears within **...** element, is displayed as deleted text.

```
<body>
  <p>I want to drink <del>cola</del> <ins>wine</ins></p>
</body>
```

This will produce the following result –
I want to drink ~~cola~~ wine

Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it as shown below –

```
<body>
  <p>The following word uses a <big>big</big> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a big typeface.

Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it as shown below –

```
<body>
  <p>The following word uses a <small>small</small> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a small typeface.

Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

- dir
- lang
- xml:lang

The dir Attribute

The **dir** attribute allows you to indicate to the browser about the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows –

Value	Meaning
ltr	Left to right (the default value)
rtl	Right to left (for languages such as Hebrew or Arabic that are read right to left)

```
<!DOCTYPE html>
<html dir = "rtl">
  <head>
```

```

<title>Display Directions</title>
</head>
<body>
  This is how IE 5 renders right-to-left directed text.
</body>
</html>

```

This will produce the following result –

.This is how IE 5 renders right-to-left directed text
 When *dir* attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

The lang Attribute

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML.

```

<!DOCTYPE html>
<html lang = "en">
  <head>
    <title>English Language Page</title>
  </head>
  <body>
    This page is using English Language
  </body>
</html>

```

This will produce the following result –

This page is using English Language

Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

Attribute	Options	Function
Align	right, left, center	Horizontally aligns tags
Valign	top, middle, bottom	Vertically aligns tags within an HTML element.
Bgcolor	numeric, hexadecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
Id	User Defined	Names an element for use with Cascading Style Sheets.
Class	User Defined	Classifies an element for use with Cascading Style Sheets.
Width	Numeric Value	Specifies the width of tables, images, or table cells.
Height	Numeric Value	Specifies the height of tables, images, or table cells.
Title	User Defined	"Pop-up" title of the elements.

We will see related examples as we will proceed to study other HTML tags.

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.

```
<!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>
```

This will produce the following result without displaying the content given as a part of comments – Valid vs Invalid Comments

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. Here, the given comment is a valid comment and will be wiped off by the browser.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Valid Comment Example</title>
  </head>
  <body>
    <!-- This is valid comment -->
    <p>Document content goes here.....</p>
  </body>
</html>
```

But, following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Invalid Comment Example</title>
  </head>
  <body>
    <!-- This is not a valid comment -->
    <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.

```
<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 - Marquee

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 `<marquee>` tag.

Note – The <marquee> tag deprecated in HTML5. Do not use this element, 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.

Sr.No	Attribute & Description
1	width This specifies the width of the marquee. This can be a value like 10 or 20% etc.
2	height This specifies the height of the marquee. This can be a value like 10 or 20% etc.
3	direction This specifies the direction in which marquee should scroll. This can be a value like <i>up</i> , <i>down</i> , <i>left</i> or <i>right</i> .
4	behavior This specifies the type of scrolling of the marquee. This can have a value like <i>scroll</i> , <i>slide</i> and <i>alternate</i> .
5	scrolldelay This specifies how long to delay between each jump. This will have a value like 10 etc.
6	scrollamount This specifies the speed of marquee text. This can have a value like 10 etc.
7	loop This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
8	bgcolor This specifies background color in terms of color name or color hex value.
9	hspace This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
10	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.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML marquee Tag</title>
  </head>
  <body>
    <marquee>This is basic example of marquee</marquee>
    <marquee width = "50%">This example will take only 50% width</marquee>
    <marquee direction = "right">This text will scroll from left to right</marquee>
    <marquee direction = "up">This text will scroll from bottom to up</marquee>
  </body>
</html>
```

HTML - Phrase Tags

The phrase tags have been desicolgned for specific purposes, though they are displayed in a similar way as other basic tags like ``, `<i>`, `<pre>`, and `<tt>`, you have seen in previous chapter. This chapter will take you through all the important phrase tags, so let's start seeing them one by one.

Emphasized Text

Anything that appears within `...` element is displayed as emphasized text.

```
<body>
  <p>The following word uses an <em>emphasized</em> typeface.</p>
</body>
```

This will produce the following result –
The following word uses an *emphasized* typeface.

Marked Text

Anything that appears with-in `<mark>...</mark>` element, is displayed as marked with yellow ink.

```
<body>
  <p>The following word has been <mark>marked</mark> with yellow</p>
</body>
```

This will produce the following result –
The following word has been **marked** with yellow

Strong Text

Anything that appears within `...` element is displayed as important text.

```
<body>
  <p>The following word uses a <strong>strong</strong> typeface.</p>
</body>
```

This will produce the following result –
The following word uses a **strong** typeface.

Text Abbreviation

You can abbreviate a text by putting it inside opening `<abbr>` and closing `</abbr>` tags. If present, the title attribute must contain this full description and nothing else.

```
<body>
  <p>My best friend's name is <abbr title = "Abhishek">Abhy</abbr>.</p>
</body>
```

This will produce the following result –
My best friend's name is Abhy.

Acronym Element

The `<acronym>` element allows you to indicate that the text between `<acronym>` and `</acronym>` tags is an acronym.

At present, the major browsers do not change the appearance of the content of the `<acronym>` element.

```
<body>
  <p>This chapter covers marking up text in <acronym>HTML</acronym>.</p>
</body>
```

This will produce the following result –
This chapter covers marking up text in HTML.

Text Direction

The `<bdo>...</bdo>` element stands for Bi-Directional Override and it is used to override the current text direction.

```
<body>
  <p>This text will go left to right.</p>
  <p><bdo dir = "rtl">This text will go right to left.</bdo></p>
</body>
```

This will produce the following result –
This text will go left to right.
.tfel ot thgir og lliw txet sihT

Special Terms

The `<dfn>...</dfn>` element (or HTML Definition Element) allows you to specify that you are introducing a special term. It's usage is similar to italic words in the midst of a paragraph. Typically, you would use the `<dfn>` element the first time you introduce a key term. Most recent browsers render the content of a `<dfn>` element in an italic font.

```
<body>
  <p>The following word is a <dfn>special</dfn> term.</p>
</body>
```

This will produce the following result –
The following word is a *special* term.

Quoting Text

When you want to quote a passage from another source, you should put it in between `<blockquote>...</blockquote>` tags.

Text inside a `<blockquote>` element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

```
<body>
  <p>The following description of XHTML is taken from the W3C Web site:</p>
  <blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on
    from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>
</body>
```

This will produce the following result –

The following description of XHTML is taken from the W3C Web site:

XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.

Short Quotations

The `<q>...</q>` element is used when you want to add a double quote within a sentence.

```
<body>
  <p>Amit is in Spain, <q>I think I am wrong</q>.</p>
</body>
```

This will produce the following result –
Amit is in Spain, I think I am wrong.

Text Citations

If you are quoting a text, you can indicate the source placing it between an opening `<cite>` tag and closing `</cite>` tag

As you would expect in a print publication, the content of the `<cite>` element is rendered in italicized text by default.

```
<body>
  <p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>
</body>
```

This will produce the following result –

This HTML tutorial is derived from *W3 Standard for 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.

```
<body>
  <p>Regular text. <code>This is code.</code> Regular text.</p>
</body>
```

This will produce the following result –
Regular text. This is code. Regular text.

Keyboard Text

When you are talking about computers, if you want to tell a reader to enter some text, you can use the `<kbd>...</kbd>` element to indicate what should be typed in, as in this example.

Example

```
<body>
```

```
<p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>
</body>
```

This will produce the following result –
Regular text. This is inside kbd element Regular text.

Programming Variables

This element is usually used in conjunction with the **<pre>** and **<code>** elements to indicate that the content of that element is a variable.

Example

```
<body>
  <p><code>document.write("<var>user-name</var>")</code></p>
</body>
```

This will produce the following result –
document.write("user-name")

Program Output

The **<samp>...</samp>** element indicates sample output from a program, and script etc. Again, it is mainly used when documenting programming or coding concepts.

```
<body>
  <p>Result produced by the program is <samp>Hello World!</samp></p>
</body>
```

This will produce the following result –
Result produced by the program is Hello World!

Address Text

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

```
<body>
  <address>388A, Road No 22, Jubilee Hills - Hyderabad</address>
</body>
```

This will produce the following result –
388A, Road No 22, Jubilee Hills – Hyderabad

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</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>
    <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>
    <p>Hello<br />
    You delivered your assignment ontime.<br />
    Thanks<br />
    Mahnaz</p>
    <center>
      <p>This text is in the center.</p>
    </center>
    <p>This is paragraph one and should be on top</p>
    <hr />
```

```

<pre>
function testFunction( strText ){
    alert (strText)
}
</pre>
<p>An example of this technique appears in the movie "12 Angry Men."</p>
<p>The following word uses a <b>bold</b> typeface.</p>
<p>The following word uses an <i>italicized</i> typeface.</p>
<p>The following word uses an <u>underlined</u> typeface.</p>
<p>The following word uses a <strike>strikethrough</strike> typeface.</p>
<p>The following word uses a <tt>monospaced</tt> typeface.</p>
<p>The following word uses a <sup>superscript</sup> typeface.</p>
<p>The following word uses a <sub>subscript</sub> typeface.</p>
<p>I want to drink <del>cola</del> <ins>wine</ins></p>
<p>The following word uses a <big>big</big> typeface.</p>
<p>The following word uses a <small>small</small> typeface.</p>
<html dir = "rtl" > </html>
<html lang = "en" > </html>

<head> <!-- Document Header Starts -->
<title>This is document title</title>
</head> <!-- Document Header Ends -->
<marquee>This is basic example of marquee</marquee>
<marquee width = "50%">This example will take only 50% width</marquee>
<marquee direction = "right">This text will scroll from left to right</marquee>
<marquee direction = "up">This text will scroll from bottom to up</marquee>
<p>The following word uses an <em>emphasized</em> typeface.</p>
<p>The following word has been <mark>marked</mark> with yellow</p>
<p>The following word uses a <strong>strong</strong> typeface.</p>
<p>My best friend's name is <abbr title = "Abhishek">Abhy</abbr>.</p>
<p>This chapter covers marking up text in <acronym>HTML</acronym>.</p>
<p>This text will go left to right.</p>
<p><bdo dir = "rtl">This text will go right to left.</bdo></p>
<p>The following word is a <dfn>special</dfn> term.</p>
<p>The following description of XHTML is taken from the W3C Web site:</p>
<blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on
from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>
<p>Amit is in Spain, <q>I think I am wrong</q>.</p>
<p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>
<p>Regular text. <code>This is code.</code> Regular text.</p>
<p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>
<p><code>document.write("<var>user-name</var>")</code></p>
<p>Result produced by the program is <samp>Hello World!</samp></p>
<address>388A, Road No 22, Jubilee Hills - Hyderabad</address>
</body>
</html>

```