

# Introduction to HTML

## **2.1 Introduction**

## **2.2 Methods of preparing an HTML document**

## **2.3 HTML file extensions**

## **2.4 Structure of an HTML document**

## **2.5 HTML Tags**

## **2.6 Text Formatting**

## **2.7 Adding Images in HTML**

## **2.8 HTML Lists**

## **2.9 Hyperlinks, Anchors, and Links**

## **2.10 Thumbnails**

## **2.11 TABLES in HTML**

## **2.12 HTML Frame Tags**

## **2.13 Form in HTML**

## 2.1 INTRODUCTION

HTML stands for **Hypertext Markup Language**. It is used to display the document in the browser to be simple text or to be complex multimedia like images and java applets. HTML is considered to be not a programming language. HTML standards are maintained by organizations called **W3C** (World Wide Web Consortium) for web page formatting.

A tag is a format name surrounded by angle brackets. Tags can also contain a forward slash. So, Hypertext refers to the process of marking up documents with tags that indicate what type of content it is and how the documents are linked together.

HTML is short for Hypertext Markup Language.

**Hypertext** is simply a piece of text that links to other pieces of text.

**Markup Language** is a way of writing language.

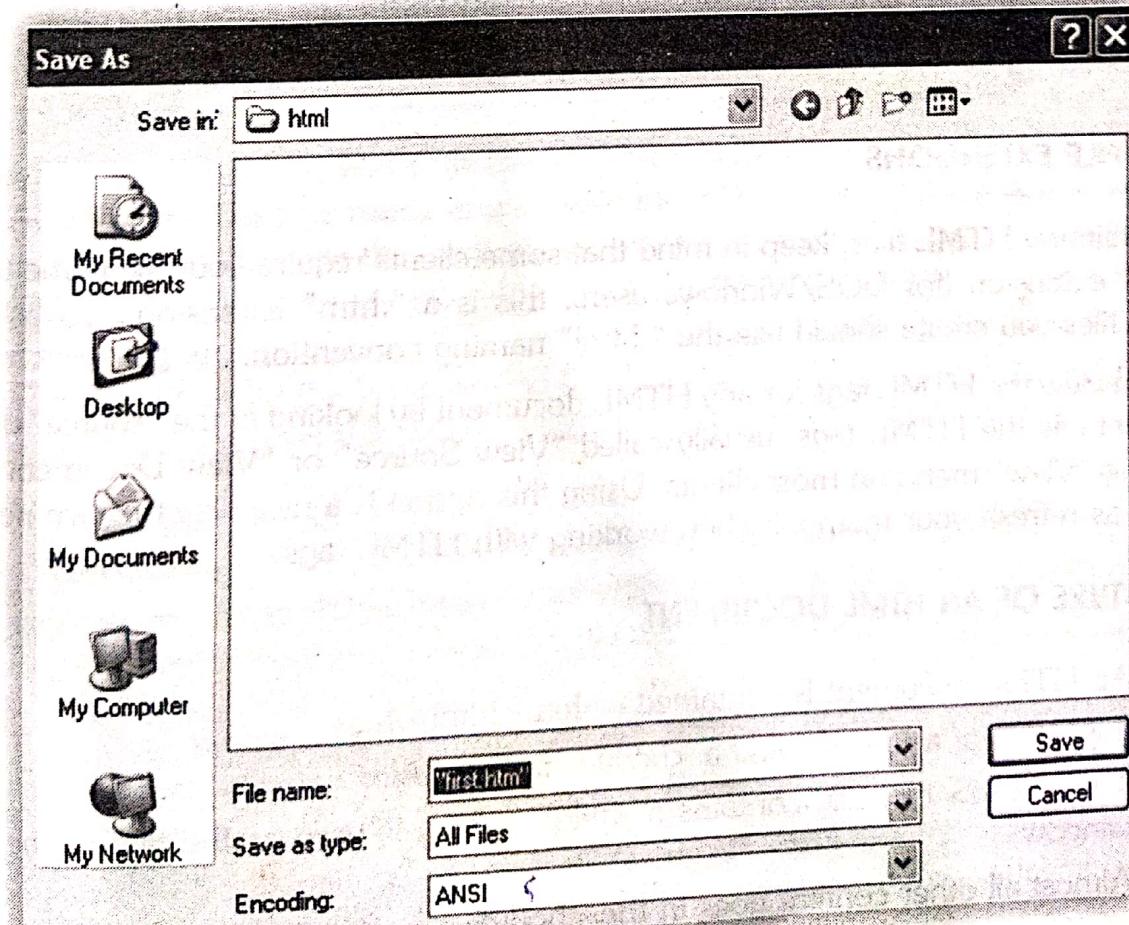
HTML is not a programming language and a computer program is a series of procedures.

Latin-1. There is no obligation to use anything but the 128 standard ASCII characters in an HTML document. In fact, sticking to straight ASCII is encouraged as it allows an HTML document to be edited by any text editor on any computer system and be transported over any network. To make this possible, HTML includes character entities for most of the commonly used non-ASCII Latin-1 characters. These character entities begin with the ampersand character (&), followed by the name or number of the character, followed by a semicolon. In the next section we describe the HTML document. HTML is **not case sensitive**.

## 2.2 METHODS OF PREPARING AN HTML DOCUMENT

To begin using HTML you will want to familiarize yourself with the software you will be using. So, to get started, open Notepad (or your text editor). You should see a completely blank page. Now, type in the text below. We will make use of it later with the web browser. Type the following:

```
<HTML>
<HEAD>
<TITLE>Test Page</TITLE>
</HEAD>
<BODY>
    Hi there, you have just written your first HTML page!
</BODY>
</HTML>
```



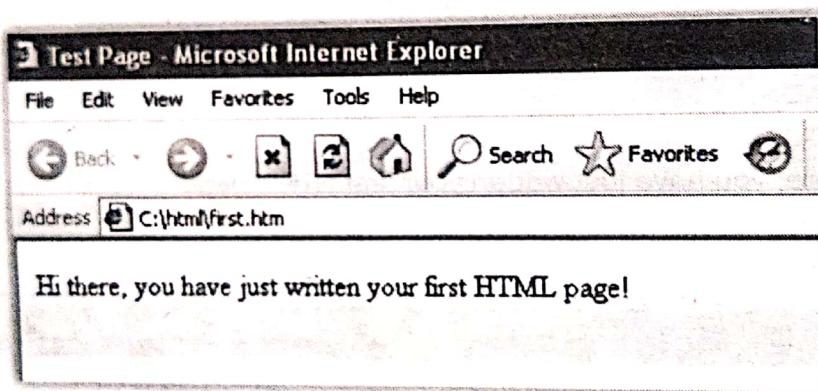
## 2.4

Once you have finished, go to the "File" menu and click on "**Save As**". This will prompt you to create a name for your file. In the box, type in **first.htm**. At the bottom of the prompt you should see a space that says "Save file as Type" or "Save as Type". The default is .txt, so you will need to change it. Click on the down arrow on the right side of the input box. You should be able to highlight **All Files (\*.\*)**. Click on this to make the change. You should see something similar to the picture previous page.

Now you may choose the drive and directory to save to, and click on "Save". You can use this routine each time you create a new HTML file.

If you do not get the option to save the file as the type **All Files (\*.\*)**, then select the plain text file type (usually shown as Text Documents, Text Files (\*.txt), or something similar in the dialogue box). Be careful not to save it just yet though if you are using Windows. Windows will save it as **first.htm.txt** by default. To get around this, you will need to place quotes around the filename.

Now you will want to use your web browser to view the file you just created. So, start your web browser.



## 2.3 HTML FILE EXTENSIONS

When naming HTML files, keep in mind that some clients require your file name to have ".html" extension (for DOS/Windows users, this is a ".htm" extension). To be safe, HTML files you create should use the ".html" naming convention.

You can view the HTML tags for any HTML document by looking at the "source" document that contains the HTML tags, usually called "View Source" or "View Document Source" from the "View" menu on most clients. Using this option is a nice way to learn new things as well as refresh your memory when working with HTML tags.

## 2.4 STRUCTURE OF AN HTML DOCUMENT

- » An HTML document is contained within <html> tags
- » It consists of a <head> and a <body>, in that order
- » The <head> typically contains a <title>, which is used as the title of the browser window
- » Almost all other content goes in the <body>

Hence, a fairly minimal HTML document looks like this:

```
<html>
  <head>
    <title>My Title</title>
  </head>
  <body>
    Hello, World!
  </body>
</html>
```

As described above an HTML program (also called an HTML script) is a sequence of three kinds of tokens ordinary text characters, tags, and special symbols.

The main commands in HTML are **Structure Tags**

**<HTML> ... </HTML>** Encloses the entire HTML document

**<HEAD> ... </HEAD>** Encloses the head of the HTML document.

**<BODY> ... </BODY>** Encloses the body (text and tags)

of the HTML document. An attribute is **BGCOLOR="..."**

**<TITLE> ... </TITLE>** Indicates the title of the document. Used with **<HEAD>**

## 2.5 HTML TAGS

HTML works as a system of tags, one word or coded commands surrounded by right angle parenthesis (<>). Most tags has a front and a back form which encases the text and instructs the browser software or computer server on how that text should appear and what functions it might activate.

All web pages begin and end with the **<HTML>** tags. After the initial **<HTML>** tag, the **<HEAD> </HEAD>** tags reside and contain the **<TITLE> </TITLE>** tags between which resides the title of the page that is displayed on the top bar of the browser screen. Next the **<BODY> </BODY>** tags are placed before and after all the text included in the page, and within body tag are specifications for text and background colors. The head tag is used for control information by the browser and the body tag contains the actual user information that is to be displayed on the screen. The basic document is shown below.

```
<html>
  <head>
    <title> Basic HTML document </title>
  </head>
  <body>
    <h1> Welcome to the world of Web Technologies </h1>
    <p> A sample HTML program </p>
  </body>
</html>
```

## 2.6

The tags (also called elements) are special instructions. Tags are identified by their names enclosed in angle brackets. The special symbols (also called entities) are code punctuations such as the ampersand & and the quotation marks ". Thus HTML markup tags are delimited by the angle brackets, < ... >. They appear either singularly, like the tag <P>, indicate a paragraph break in the text, or as a pair of starting and ending tags that wrap the content contained.

For example :

<B> Attention! </B>

is an instruction to present the text string Attention! in a bold typeface. Other examples,

<COMMENT> This is a comment </COMMENT>

which is a comment on an HTML file

and

<CENTER> ... </CENTER>

will centre the text or image.

There are tags for formatting text, tags for specifying hypertext links, tags for including sound and picture elements, tags for defining input fields for interactive pages.

Points to be remembered for HTML tags:

- » Tags are delimited by angled brackets.
- » They are not case sensitive i.e., <head>, <HEAD> and <Head> is equivalent.
- » If a browser does not understand a tag it will usually ignore it.
- » Some characters have to be replaced in the text by escape sequences.
- » White spaces, tabs and newlines are ignored by the browser.

### 2.5.1 Basic HTML tags

**1. Body tag:** Body tag contains some attributes such as bgcolor, background etc. bgcolor is used for background color, which takes background color name or hexadecimal number #FFFFFF and background attribute will take the path of the image which you can place as the background image in the browser.

<body bgcolor="#F2F3F4" background="c:\WEB\image1.gif">

**2. Paragraph tag:** Most text is part of a paragraph of information. Each paragraph can be aligned to the left, right or center of the page by using an attribute called as align.

<p align="left" | "right" | "center">

**3. Heading tag:** HTML is having six levels of heading that are commonly used. The largest heading tag is <h1>. The different levels of heading tag besides <h1> are <h2>, <h3>, <h4>, <h5> and <h6>. These heading tags also contain attribute called as

<h1 align="left" | "right" | "center"> . . . </h1>

**4. hr tag:** This tag places a horizontal line across the system. These lines are used to break the page. This tag also contains attribute i.e., width which draws the horizontal line with the screen size of the browser. This tag does not require an end tag.

<hr width="50%">>

**5. base font:** This specifies format for the basic text but not the headings.

<basefont size="10">

**6. font tag:** This sets font size, color and relative values for a particular text.

<font size="10" color="#f1f2f3">

**7. bold tag:** This tag is used for implementing bold effect on the text.

<b> ..... </b>

**8. Italic tag:** This implements italic effects on the text.

<i>.....</i>

**9. strong tag:** This tag is used to always emphasize the text

<strong>.....</strong>

**10. tt tag:** This tag is used to give typewriting effect on the text

<tt>.....</tt>

**11. sub and sup tag:** These tags are used for subscript and superscript effects on the text.

<sub> .....</sub>  
<sup> .....</sup>

**12. Break tag:** This tag is used to break the line and start from the next line.

<br />

**13. &amp &lt &gt &nbsp &quot :** These are character escape sequence which are required if you want to display characters that HTML uses as control sequences.

Example: < can be represented as &lt;

**14. Anchor tag:** This tag is used to link two HTML pages, this is represented by <a>

<a href=" path of the file"> some text </a>

href is an attribute which is used for giving the path of a file which you want to link.

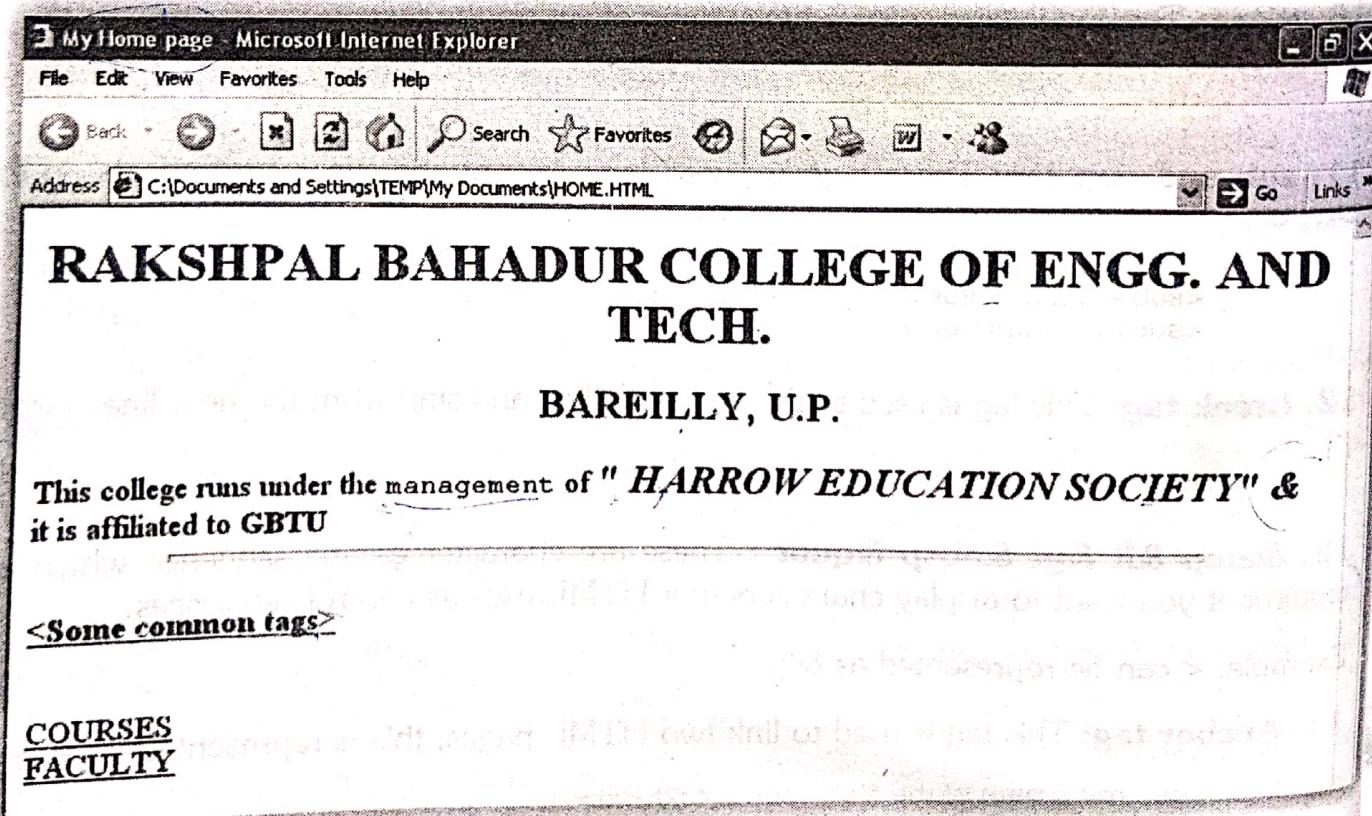
Example

```

<html>
<head> <i — This page implements common html tags —>
<title> My Home page </title>
</head>
<body>

<h1 align="center"> RAKSHPAL BAHADUR COLLEGE OF ENGG. AND TECH.</h1>
<h2 align="center"> BAREILLY, U.P.</h2>
<basefont size=4>
<p> This college runs under the <tt>management</tt> of <font size=5>
<b><i>&quot;
HARROW EDUCATION SOCIETY&quot;&amp;nbsp;</i></b> </font><br>
it is affiliated to <strong> GBTU</strong>
<hr size=5 width=80%>
<h3> <u>&lt;Some common tags&gt;</u> </h3><br>
<a href="COURSES.html"> COURSES </a><br>
<a href="FACULTY.html"> FACULTY </a><br>
</body>
</html>

```



## 2.6 TEXT FORMATTING

### <em> and <strong> tag

Let's say that you have some word or words that you think need to be emphasized. You can mark them as emphasized by surrounding them with an opening **<em>** tag and closing **</em>** tag.

You can strongly emphasize text by using an opening and closing **<strong>** tag. Put the tags around a word or words in your document, save the file, and reload it in the browser.

#### Example

```
<html>
<head>
<title> formatting </title>
</head>
<body>
```

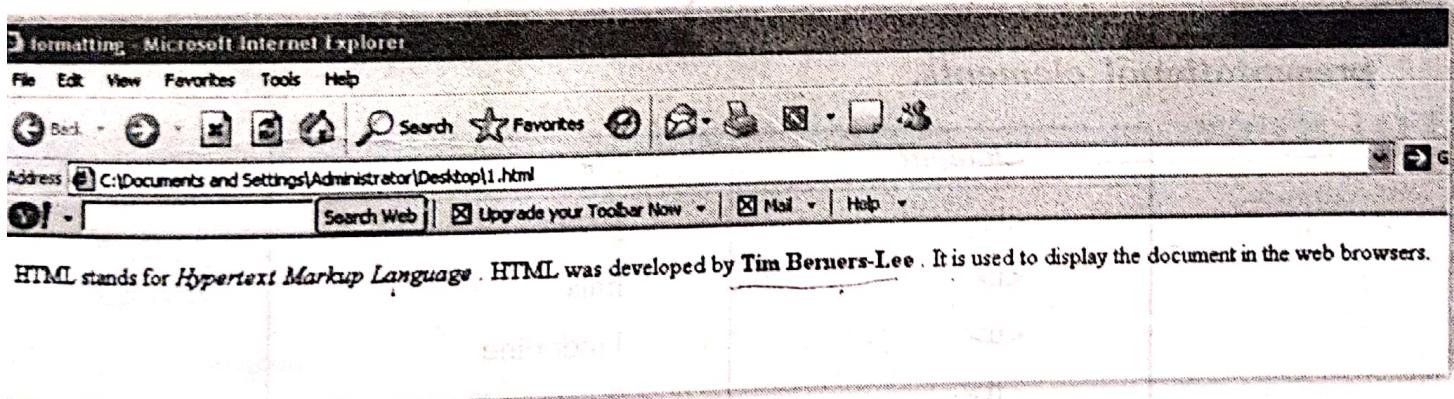
HTML stands for **<em>** Hypertext Markup Language **</em>**.

HTML was developed by **<strong>** Tim Berners-Lee **</strong>**.

It is used to display the document in the web browsers.

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

#### Output:



The **<em>** and **<strong>** elements are called *logical formatting elements*, or *logical styles*. They tell what kind of effect you want, and leave it up to the browser to decide how that effect should look. While **<em>** and **<strong>** are the most widely used, there are other logical formatting elements that you can use.

Try typing these examples into an HTML document and see what they produce.

**<sup>** - superscript

$e = mc<sup>2</sup>$

**<sub>** - subscript

Water is  $H<sub>2</sub>O$ .

2.10

**<cite>** - a citation or reference to some work (book, poem, etc.).

Joy Kilmer wrote the poem <cite>Trees</cite>.

**<dfn>** - mark the defining instance of a word

A <dfn>quadruped</dfn> has four feet.

**<samp>** - Sample output from programs, etc.

The dialog box will say <samp>Click OK</samp>.

**<kbd>** - Keyboard input from users

Press the <kbd>Enter</kbd> key.

**<code>** - Computer programming code

<code>x = y + z \* w;</code>

**<small>** - for "fine print"

<small>Not available in all colors.</small>

**<q>** - for quoting text. This element adds quote marks for you.

<q>And I am Anil of India.</q> - Sumit

**<abbr>** - represents an abbreviation or acronym

You are learning <abbr>HTML</abbr>.

**<big>** bumps the text up one size, **<small>** bumps it down one size.

The <cite> and <dfn> both appear in italics; the <samp>, <kbd>, and <code> all appear in a monospace font, where all the letters are the same width, like you'd see on a teletype machine or a typewriter.

HTML does include elements that are tied to their presentation; these are called **presentational elements**.

Element	Means
<b>	Bold
<i>	Italic
<u>	Underline
<tt>	Teletype Text
<strike>	Strike-through
<big>	larger than normal

The last four elements are **deprecated**; you will find them in HTML4 and XHTML, but have been removed from HTML5.

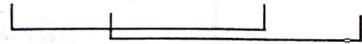
### Nesting Elements

What if you want to have two effects on a word? Say you're defining a word that's important, so you want it to be both strongly emphasized and a definition? Just put the

When you put tags inside one another, the last tag in must be the first tag out. Stated another way, you must close tags in the reverse order that you opened them.

This is an example of *nested tags*. If you are going to use tag pairs in combination (which you will probably be doing quite a bit), then to avoid confusing the browser (not to mention confusing yourself) they should be nested, not overlapping. Let me illustrate...

<this><that>cool</this></that> *<— Overlapping tags.... bad*



<this><that>cool</that></this> *<— Nested tags.... good*



## Headings

First, let's work on the title and author name. In most books, those would be a heading and a subheading. HTML has six elements, `<h1>`, `<h2>`, ... through `<h6>` which give you six levels of headings. In typography, a level one heading is a main heading, a level two heading is a subheading, etc. Let's see what happens when you add these headings. In proper HTML usage, each heading starts a new section of the document.

<H1> ... </H1> A first-level heading

<H2> ... </H2> A second-level heading

<H3> ... </H3> A third-level heading

<H4> ... </H4> A fourth-level heading

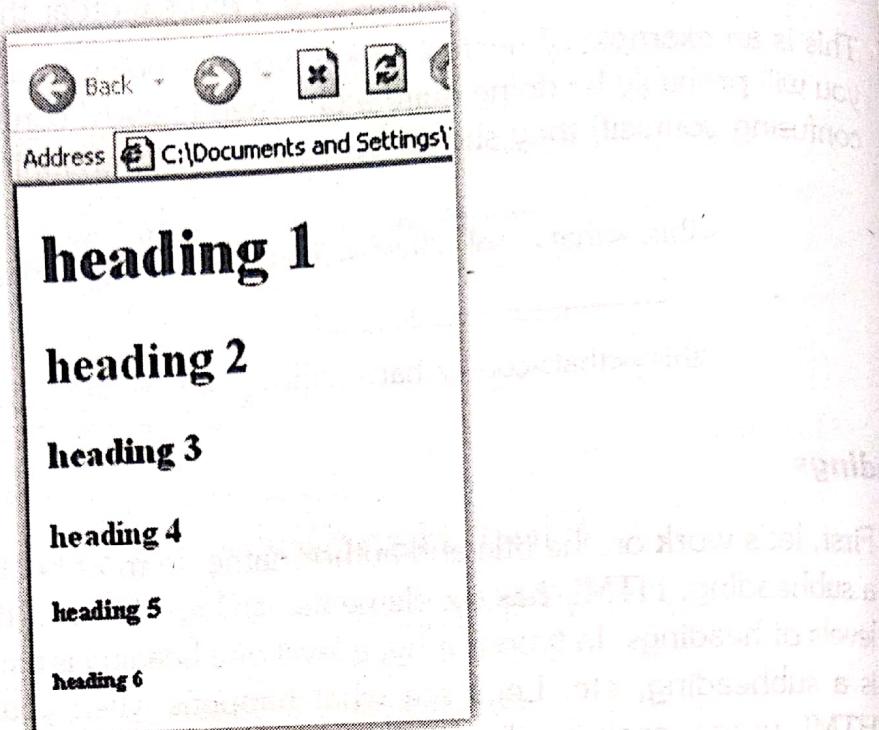
<H5> ... </H5> A fifth-level heading

<H6> ... </H6> A sixth-level heading

All heading tags accept the attribute `ALIGN="..."` Possible values are **CENTER**, **LEFT**, and **RIGHT**

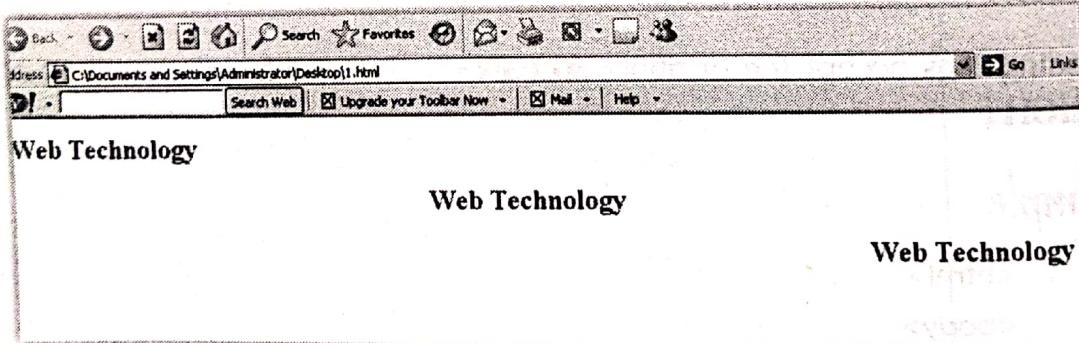
## Example

```
<html>
  <body>
    <h1>heading 1</h1>
    <h2>heading 2</h2>
    <h3>heading 3</h3>
    <h4>heading 4</h4>
    <h5>heading 5</h5>
    <h6>heading 6</h6>
  </body>
</html>
```

**Output :**

A useful heading attribute is **align**. It's fairly self-explanatory...

```
<h2 align="left">Web Technology </h2>
<h2 align="center"> Web Technology </h2>
<h2 align="right"> Web Technology </h2>
```

**Output :****Paragraphs**

You can't use headers for all the lines in the poem and the explanatory text that follows don't want everything bold, and there'd be too much space between lines. Aside from the poem and the explanation aren't headers, so it would be a misuse of HTML to structure the document that way.

Instead, we are going to use the **<p>** element, which stands for **paragraph** to mark the beginning and end of paragraphs. Note that, in accordance with the rules for writing the XHTML syntax, every opening tag must have a closing tag.

## Example

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

## Output:

# My First Heading

My first paragraph.

So, what else is this **< p >** tag good for? Well, it's great for **aligning** stuff.

## Example

```
<html>
<head>
<title> formatting </title>
</head>
<body>
<p align="left">HTML</p>
<p align="center">DHTML</p>
<p align="right">XML</p>
<p align="left">
Hypertext Markup Language</p>
<p align="center">
Something really cool<br />
Something really cool<br />
like an icecube<br />
</p>
<p align="right">
Something really cool<br />
Something really cool<br />
or a popsicle</p>
</body>
</html>
```

HTML

Hypertext Markup Language

DHTML

Something really cool  
like an icecube

XML

Something really cool  
or a popsicle

## Line Breaks

You can't use paragraphs for every line of the poem; if you do, the poem will appear to doublespaced. What you need is an HTML element that says "go to the next line", and the `<br />` element does just that.

In XHTML, you must have a closing tag for every opening tag. Even though you can't put text into a line break, you still need both tags. The technical term for an element that never has any content is an *empty element*.

Note: An element that has both opening and closing tags and content in between is called a *container element*.

Do not write a line break like this:

`<br></br>`

First, it is not valid in HTML5. Second, it may mislead you into thinking that you should put something between those tags. XHTML syntax (which we are using in this course) allows you to write empty elements with a shortcut. The slash just before the greater than sign means "this is an opening and closing tag all wrapped up in one." Notice the blank before the slash. It is necessary.

`<br />`

**Example:** Why do we insist on a blank before the slash in a shortcut form like `<br/>`?  
**Solution :** Older browsers, when confronted with `<br/>`, will not realize that the tag name ends with the letter R, but will think that the slash is part of the tag name, and ignore the tag as meaningless.

Use the shortcut form for empty elements like `<br />` only with elements that can never contain text. Let's say you had a level two heading with no text in it; this might happen if you know you want a heading but haven't figured out what to say yet. You must always write it like this: `<h2></h2>`. Do not ever write it like this: `<h2 />`. If you do, older browsers will see only an opening `<h2>` without a closing tag, and the rest of your document will be much larger than you want.

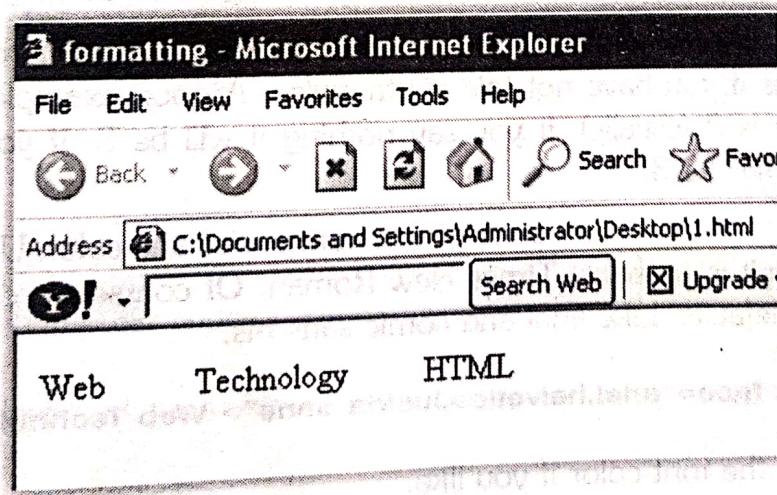
## Special character

The `&` means we are beginning a special character, the `;` means ending a special character, and the letters in between are sort of an abbreviation for what it's for. There are quite a few of these special characters. Here are five more. (Note: these should always be lower case.)

<code>&amp;nbsp;</code>	non-breaking space
<code>&amp;lt;</code>	< less-than symbol
<code>&amp;gt;</code>	> greater-than symbol
<code>&amp;amp;</code>	& ampersand
<code>&amp;quot;</code>	" quotation mark

Try this instead...

```
<body>
Web &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
Technology&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
HTML
</body>
```



## Text Formatting

A very useful type of text effect is the mono-spaced font, or Typewriter Text.

Each letter uses the same amount of horizontal space...

This is regular type ->

iiiiiiii  
oooooooooooo  
mmmmmmmmmm

This is monospaced type -> ooooooo

mmmmmmmmmm

We can change the font Size too... It's pretty easy!

First add the **<font>** tags...

```
<body>  
Something really <font>cool</font>  
</body>
```

Then specify a **size** attribute.

```
<body>  
Something really <font size="6">cool</font>  
</body>
```

Fonts come in 7 sizes:

tiny	small	regular	medium	large	big	yelling
1	2	3	4	5	6	7

Two things I want to discuss now. First, a **<tag>** tells the browser to do something. An attribute goes inside the **<tag>** and tells the browser *how to do it*.

Second point is about defaults. As you probably know, the *default value* is a value that the browser assumes if you have not told it otherwise. A good example is the font size. The default font size is 3 (usually). If you say nothing it will be 3. If you make faces at your computer it will still be 3.

Every browser has a default font setting - font name, size and color. Unless you have made changes with it, the default is probably Times New Roman. Of course we can specify font names other than the defaults. Like arial and comic sans ms.

```
<font face="arial,helvetica,lucida sans"> Web Technology</font>
```

You can change the font color if you like.

```
<body>  
Something really <font color="#ff0000">cool</font>  
</body>
```

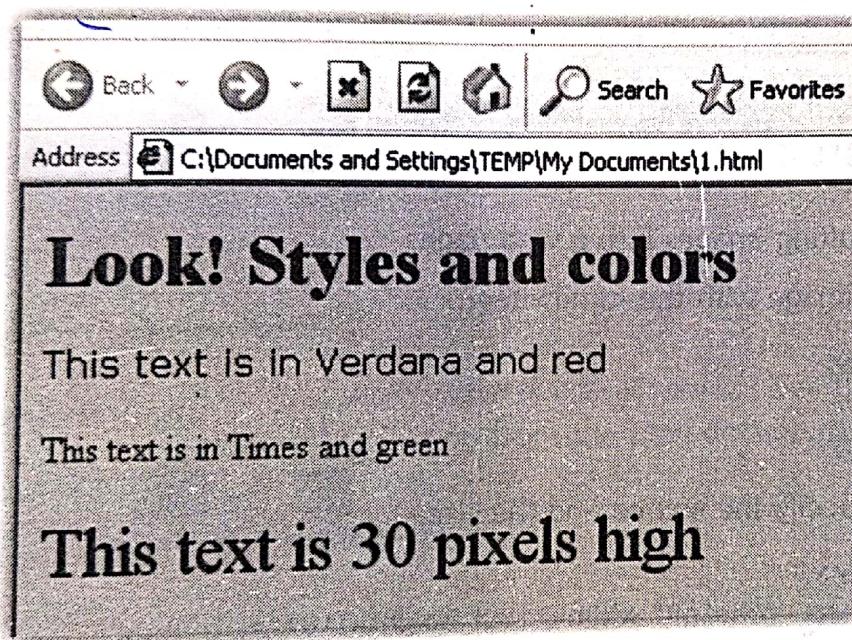
We can, of course use more than one attribute in a **<font>** tag...

```
<body>  
Something really <font color="#ff0000" face="arial" size="7">cool</font>  
</body>
```

## Example

```
<html>
<body style="background-color:PowderBlue;">
<h1>Look! Styles and colors</h1>
<p style="font-family:verdana;color:red">
This text is in Verdana and red</p>
<p style="font-family:times;color:green">
This text is in Times and green</p>
<p style="font-size:30px">This text is 30 pixels high</p>
</body>
</html>
```

## Output:



## Example

```
<html>
<body>
<code>Computer code</code>
<br />
<kbd>Keyboard input</kbd>
<br />
<tt>Teletype text</tt>
<br />
<samp>Sample text</samp>
<br />
<var>Computer variable</var>
<br />
<p><b>Note:</b> These tags are often used to display computer/programming
code.</p>
</body>
</html>
```