

Attributes

We have seen few HTML tags and their usage like heading tags <h1>, <h2>, paragraph tag <p> and other tags. We used them so far in their simplest form, but most of the HTML tags can also have attributes, which are extra bits of information.

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts - a name and a value

- The name is the property you want to set. For example, the paragraph element in the example carries an attribute whose name is align, which you can use to indicate the alignment of paragraph on the page.
- The value is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: left, center and right.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Example:-

```
<!DOCTYPE html>
<html>
<head>
<title>Align Attribute Example</title>
</head>
<body>
<p align = "left">This is left aligned </p>
<p align = "center">This is center aligned </p>
<p align = "right">This is right aligned </p >
</body>
</html>
```

This will produce the following result -

This is left aligned

This is center aligned

This is right aligned

Core Attributes:-

The four core attributes that can be used on the majority of HTML elements (although not all) are -

- Id
- Title
- Class
- Style

The Id Attribute:-

The id attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element -

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.
- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

We will discuss style sheet in separate tutorial. For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

Example

```
<p id = "html">This para explains what is HTML</p>  
<p id = "css">This para explains what is Cascading Style Sheet</p>
```

The Title Attribute:-

The title attribute gives a suggested title for the element. The syntax for the title attribute is similar as explained for id attribute -

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>The title Attribute Example</title>
</head>
<body>
<h3 title = "Hello HTML!">Titled Heading Tag Example</h3>
</body>
</html>
```

This will produce the following result -

Titled Heading Tag Example

The class Attribute:-

The class attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the use of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it.

The value of the attribute may also be a space-separated list of class names. For example -

```
class = "className1 className2 className3"
```

The Style Attribute:-

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

```
<!DOCTYPE html>
<html>
<head>
<title>The style Attribute</title>
</head>
<body>
<p style = "font-family:arial; color:#FF0000;">Some text...</p>
</body>
</html>
```

This will produce the following result -

Some text...

At this point of time, we are not learning CSS, so just let's proceed without bothering much about CSS. Here, you need to understand what are HTML attributes and how they can be used while formatting content.

Internationalization Attributes:-

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

- 1).dir
- 2).lang
- 3).xml:lang

1).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 -

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

Example

```
<!DOCTYPE html>
<html dir = "ltr">
<head>
<title>Display Directions</title>
</head>
<body>
This is how IE 5 renders left-to-right directed text.
</body>
</html>
```

This will produce the following result -

This is how IE 5 renders left-to-right 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 -

1).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. This attribute has been replaced by the xml:lang attribute in new XHTML documents.

The values of the lang attribute are ISO-639 standard two-character language codes. Check HTML Language Codes: ISO 639 for a complete list of language codes.

```
<!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

3).The xml:lang Attribute:-

The xml:lang attribute is the XHTML replacement for the lang attribute. The value of the xml:lang attribute should be an ISO-639 country code as mentioned in previous section.

Generic Attributes

- align - Horizontally aligns tags
- valign - Vertically aligns tags within an HTML element.
- bgcolor - Places a background color behind an element
- background - Places a background image behind an element
- id - Names an element for use with Cascading Style Sheets.
- class - Classifies an element for use with Cascading Style Sheets.
- width - Specifies the width of tables, images, or table cells.
- height - Specifies the height of tables, images, or table cells.
- title - "Pop-up" title of the elements.

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>
```

This will produce the following result -

This is heading 1

This is heading 2

This is heading 3

This is heading 4

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 -

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>
```

This will produce the following result -

Here is a first paragraph of text.
Here is a second paragraph of text.
Here is a third paragraph of text.

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.

The
 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
 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 ontime.<br/>
Thanks<br/>
Mahnaz</p>
```

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

This will produce the following result -

Hello
You delivered your assignment on time.
Thanks
Mahnaz

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>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 -

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>
```

This will produce the following result -

This is paragraph one and should be on top

This is paragraph two and should be at bottom

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>
```

This will produce the following result -

```
function testFunction( strText ){  
    alert (strText)  
}
```

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.

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>
```

This will produce the following result without displaying the content given as a part of comments -

Document content goes here.....

Valid vs Invalid Commentst:-

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:-

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

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

This will produce the following result -

Document content goes here.....

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.

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

This will produce the following result -

```
<      !--      This is invalid comment --> Document content
goes here.....
```

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:-

```
<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>
```

This will produce the following result -

Document content goes here.....

Conditional Comments:-

Conditional comments only work in Internet Explorer (IE) on Windows but they are ignored by other browsers. They are supported from Explorer 5 onwards, and you can use them to give conditional instructions to different versions of IE.

Example

```
<html>
<head>
<title>Conditional Comments</title>
<!--[if IE 6] >
Special instructions for IE 6 here
```

```
<![endif]-- >
</head>
<body>
<p>Document content goes here.....</p>
</body>
</html>
```

You will come across a situation where you will need to apply a different style sheet based on different versions of Internet Explorer, in such situation conditional comments will be helpful.

Using Comment Tag:-

There are few browsers that support

If you are using IE, then it will produce following result -

This is Internet Explorer.

But if you are not using IE, then it will produce following result -

This is not Internet Explorer.

Commenting Script Code:-

Though you will learn JavaScript with HTML, in a separate tutorial, but here you must make a note that if you are using Java Script or VB Script in your HTML code then it is recommended to put that script code inside proper HTML comments so that old browsers can work properly.

Example

```
<html>
<head>
<title>Commenting Script Code</title>
<script> <!-- document.write("Hello World!") //--> </script> </head>
<body>
<p> Hello , World! </p>
</body>
</html>
```

This will produce the following result -

This text will go left to right.

Hello World!

Hello , World!

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.

Example

```
<html>
<head>
<title>Special Terms Example</title>
</head>
<body>
<p>The following word is a <dfn>special</dfn> term.</p>
</body>
</html>
```

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.

Example

```
<html>
<head>
<title>Blockquote Example</title>
</head>
<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>
</html>
```

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.

Example

```
<html>
<head>
<title>Double Quote Example</title>
</head>
<body>
<p>Amit is in Spain, <q>I think I am wrong</q>.</p>
```



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

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.

Example

```
<html>  
<head>  
<title>Citations Example</title>  
</head>  
<body>  
<p>This HTML tutorial is derived from <cite>W3 Standard for  
HTML</cite>.</p>  
</body>  
</html>
```

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.

Example

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

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

```
<html>
<head>
<title>Keyboard Text Example Example</title>
</head>
<body>
<p>Regular text. <kbd>This is inside kbd element.</kbd> Regular
text.</p>
</body>
</html>
```

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

```
<html>
<head>
<title>Variable Text Example Example</title>
</head>
<body>
<p> <code>document.write("<var>user-name</var>")</code> </p>
</body>
</html>
```

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.

Example

```
<html>
<head>
<title>Program Output Example Example</title>
</head>
<body>
<p>Result produced by the program is <samp>Hello World!</samp> </p>
</body>
</html>
```

This will produce the following result -

Result produced by the program is `Hello World!`

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 -

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>
```

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 -

Example:-

```
<html>
<head>
<title>Bold Text Example</title>
</head>
<body>
<p>The following word uses an <i>italicized</i> typeface.</p>
</body>
</html>
```

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 -

Example

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

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 -

Example:-

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

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.

Example

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

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.

Example

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

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.

Example

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

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.

Example

```
<html>
<head>
<title>Inserted Text Example</title>
</head>
<body>
<p>I want to drink <del>cola</del> <ins>wine</ins>.</p>
</body>
</html>
```


This will produce the following result -

I want to drink ~~cola~~ wine

Deleted Text:-

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

Example

```
<html>
<head>
<title>Deleted Text Example</title>
</head>
<body>
<p>I want to drink <del>cola</del> <ins>wine</ins>.</p>
</body>
</html>
```

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 -

Example

```
<html>
<head>
<title>Larger Text Example</title>
</head>
<body>
<p>The following word uses a <big>big </big> typeface.</p>
</body>
</html>
```

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 -

Example

```
<html>
<head>
<title>Smaller Text Example</title>
</head>
<body>
<p>The following word uses a <small>small </small> typeface.</p>
</body>
</html>
```

This will produce the following result -

The following word uses a small typeface.

OVERVIEW

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 carrer 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 understands the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

Hello World using HTML.

Just to give you a little excitement about HTML, I'm going to give you a small conventional HTML Hello World program, You can try it using Demo link.

```
<!DOCTYPE html>
<html>
<head>
<title>This is document title</title>
</head>
<body>
<h1>this is a heading</h1>
<p>Hello World!</p>
</body>
</html>
```

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.

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.

Phrase tags

The phrase tags have been designed 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.

Example:-

```
<!DOCTYPE html>
<html>
<head>
<title>Emphasized Text Example</title>
</head>
<body>
<p>The following word uses a <em>emphasized</em> typeface.</p>
</body>
</html>
```

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.

Example:-

```
<html>
<head>
```

```
<title>Marked Text Example</title>
</head>
<body>
<p>The following word has been <mark>marked</mark> with
yellow.</p>
</body>
</html>
```

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.-

Example

```
<html>
<head>
<title>Strong Text Example</title>
</head>
<body>
<p>The following word uses an <strong>strong</strong> typeface.</p>
</body>
</html>
```

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.

Example:-

```
<html>
<head>
<title>Text Abbreviation</title>
</head>
<body>
<p>The following word uses an <strike>strikethrough</strike>
typeface.</p>
</body>
</html>
```

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.

Example

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

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.

Example

```
<html>
<head>
<title>Acronym Example</title>
</head>
<body>
<p>This chapter covers marking up text in
<acronym>XHTML</acronym>.</p>
</body>
</html>
```

This will produce the following result -

This chapter covers marking up text in XHTML.

Text Direction:-

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

Example

```
<html>
<head>
<title>Text Direction Example</title>
</head>
<body>
<p> This text will go left to right. </p>
```

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

This will produce the following result -

This text will go left to right.

This text will go right to left.

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.

Example

```
<html>
<head>
<title>Special Terms Example</title>
</head>
<body>
<p>The following word is a <dfn>special</dfn> term.</p>
</body>
</html>
```

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.

Example

```
<html>
<head>
<title>Blockquote Example</title>
</head>
<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>
</html>
```

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.

Example

```
<html>
<head>
<title>Double Quote Example</title>
</head>
<body>
```

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

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.

Example

```
<html>
<head>
<title>Citations Example</title>
</head>
<body>
<p>This HTML tutorial is derived from <cite>W3 Standard for
HTML</cite>.</p>
</body>
</html>
```

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.

Example

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

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

```
<html>
<head>
<title>Keyboard Text Example Example</title>
</head>
<body>
<p>Regular text. <kbd>This is inside kbd element.</kbd> Regular
text.</p>
</body>
</html>
```

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

```
<html>
<head>
<title>Variable Text Example Example</title>
</head>
<body>
<p> <code>document.write("<var>user-name</var>")</code> </p>
</body>
</html>
```

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.

Example

```
<html>
<head>
<title>Program Output Example Example</title>
</head>
<body>
<p>Result produced by the program is <samp>Hello World!</samp> </p>
```

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

This will produce the following result -

Result produced by the program is Hello World!