



# H T M L

hypertext markup language

**tutorialspoint**

S I M P L Y E A S Y L E A R N I N G

[www.tutorialspoint.com](http://www.tutorialspoint.com)



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

## About the Tutorial

---

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.

## Audience

---

This 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.

## Copyright & Disclaimer

---

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at [contact@tutorialspoint.com](mailto:contact@tutorialspoint.com)

## Table of Contents

---

About the Tutorial.....	i
Audience .....	i
Prerequisites .....	i
Copyright & Disclaimer.....	i
Table of Contents .....	ii
<b>1. HTML – OVERVIEW.....</b>	<b>1</b>
Basic HTML Document .....	1
HTML Tags.....	2
HTML Document Structure.....	3
The <!DOCTYPE> Declaration .....	3
<b>2. HTML – BASIC TAGS.....</b>	<b>4</b>
Heading Tags .....	4
Paragraph Tag .....	5
Line Break Tag.....	5
Centering Content.....	6
Horizontal Lines .....	6
Preserve Formatting.....	7
Nonbreaking Spaces.....	8
<b>3. HTML – ELEMENTS .....</b>	<b>9</b>
HTML Tag vs. Element .....	9
Nested HTML Elements .....	9
<b>4. HTML – ATTRIBUTES.....</b>	<b>11</b>
Core Attributes.....	12
Internationalization Attributes.....	13
The xml:lang Attribute .....	15

5.	HTML – FORMATTING .....	17
	<b>Bold Text</b> .....	17
	<i>Italic Text</i> .....	17
	<u>Underlined Text</u> .....	18
	<del>Strike Text</del> .....	18
	<b>Monospaced Font</b> .....	19
	<sup>Superscript Text</sup> .....	19
	<sub>Subscript Text</sub> .....	20
	<b>Inserted Text</b> .....	20
	<b>Deleted Text</b> .....	21
	<b>Larger Text</b> .....	21
	<b>Smaller Text</b> .....	22
	<b>Grouping Content</b> .....	22
6.	HTML – PHRASE TAGS .....	24
	<b>Emphasized Text</b> .....	24
	<b>Marked Text</b> .....	24
	<b>Strong Text</b> .....	25
	<b>Text Abbreviation</b> .....	25
	<b>Acronym Element</b> .....	26
	<b>Text Direction</b> .....	26
	<b>Special Terms</b> .....	27
	<b>Quoting Text</b> .....	27
	<b>Short Quotations</b> .....	28
	<b>Text Citations</b> .....	28
	<b>Computer Code</b> .....	29
	<b>Keyboard Text</b> .....	29
	<b>Programming Variables</b> .....	30

Program Output .....	30
Address Text .....	31
7. HTML – META TAGS .....	32
Adding Meta Tags to Your Documents .....	32
Specifying Keywords .....	32
Document Description .....	33
Document Revision Date .....	33
Document Refreshing.....	34
Page Redirection .....	35
Setting Cookies.....	35
Setting Author Name .....	36
Specify Character Set .....	36
8. HTML – COMMENTS.....	38
Valid vs Invalid Comments .....	38
Multiline Comments.....	39
Conditional Comments.....	40
Using Comment Tag .....	40
Commenting Script Code.....	41
Commenting Style Sheets.....	41
9. HTML – IMAGES .....	43
Insert Image .....	43
Set Image Location .....	44
Set Image Width/Height.....	44
Set Image Border.....	45
Set Image Alignment .....	46
Free Web Graphics .....	46

10. HTML – TABLES .....	47
Table Heading .....	48
Cellpadding and Cellspacing Attributes .....	49
Colspan and Rowspan Attributes .....	50
Tables Backgrounds.....	50
Table Height and Width .....	52
Table Caption .....	53
Table Header, Body, and Footer .....	54
Nested Tables.....	55
11. HTML – LISTS.....	57
HTML Unordered Lists .....	57
The type Attribute.....	58
HTML Ordered Lists.....	60
The type Attribute.....	60
The start Attribute .....	64
HTML Definition Lists .....	64
12. HTML – TEXT LINKS.....	66
Linking Documents.....	66
The target Attribute .....	66
Use of Base Path .....	67
Linking to a Page Section.....	68
Setting Link Colors.....	69
Download Links.....	69
File Download Dialog Box.....	70
13. HTML – IMAGE LINKS .....	71
Mouse-Sensitive Images.....	71

Server-Side Image Maps.....	72
Client-Side Image Maps.....	73
Coordinate System .....	74
14. HTML – EMAIL LINKS .....	75
HTML Email Tag.....	75
Default Settings.....	75
15. HTML – FRAMES.....	76
Disadvantages of Frames .....	76
Creating Frames .....	76
The <frameset> Tag Attributes .....	78
The <frame> Tag Attributes.....	79
Browser Support for Frames .....	80
Frame's name and target attributes .....	80
16. HTML – IFRAMES.....	83
The <iframe> Tag Attributes.....	84
17. HTML – BLOCKS.....	85
Block Elements.....	85
Inline Elements .....	85
Grouping HTML Elements.....	85
The <div> tag.....	85
The <span> tag .....	87
18. HTML – BACKGROUNDS .....	88
Html Background with Colors.....	88
Html Background with Images .....	89
Patterned & Transparent Backgrounds .....	90

19. HTML – COLORS .....	92
HTML Color Coding Methods.....	92
HTML Colors - Color Names .....	92
W3C Standard 16 Colors .....	93
HTML Colors - Hex Codes.....	93
HTML Colors - RGB Values .....	95
Browser Safe Colors .....	96
20. HTML – FONTS .....	99
Set Font Size.....	99
Relative Font Size .....	100
Setting Font Face.....	101
Specify alternate font faces.....	101
Setting Font Color .....	102
The <basefont> Element: .....	102
Example of the <basefont> Element.....	103
21. HTML – FORMS .....	104
Form Attributes.....	104
HTML Form Controls .....	105
Text Input Controls.....	105
Single-line text input controls .....	105
Attributes.....	106
Password Input controls.....	106
Attributes.....	107
Multiple-Line Text Input Controls.....	107
Attributes.....	108
Checkbox Control .....	109
Attributes.....	109

Radio Button Control .....	110
Select Box Control .....	111
Attributes.....	111
File Upload Box .....	112
Button Controls.....	113
Hidden Form Controls .....	114
22. HTML – EMBED MULTIMEDIA .....	115
The <embed> Tag Attributes .....	116
Supported Video Types .....	116
Background Audio.....	117
HTML Object tag .....	118
23. HTML – MARQUEES.....	120
The <marquee> Tag Attributes .....	120
24. HTML – HEADER .....	123
The HTML <title> Tag .....	123
The HTML <meta> Tag.....	124
The HTML <base> Tag.....	125
The HTML <link> Tag .....	125
The HTML <style> Tag .....	126
The HTML <script> Tag .....	127
25. HTML – STYLE SHEET .....	128
External Style Sheet .....	129
Internal Style Sheet.....	130
Inline Style Sheet .....	131
26. HTML JAVASCRIPT .....	133
External JavaScript .....	133

Internal Script .....	134
Event Handlers .....	135
Hide Scripts from Older Browsers .....	135
The <noscript> Element.....	136
Default Scripting Language.....	136
<b>27. HTML – LAYOUTS.....</b>	<b>138</b>
HTML Layout - Using Tables .....	138
Multiple Columns Layout - Using Tables.....	139
HTML Layouts - Using DIV, SPAN .....	141
<b>28. HTML – TAG REFERENCE .....</b>	<b>143</b>
HTML <comment> and <!--.....--> Tag .....	150
Browser Support .....	151
HTML <doctype> Tag.....	151
HTML <a> Tag.....	153
Global Attributes.....	153
Specific Attributes .....	153
Event Attributes .....	155
HTML <abbr> Tag .....	156
HTML <acronym> Tag.....	157
HTML <address> Tag .....	158
HTML <applet> Tag .....	159
HTML <area> Tag.....	161
HTML <article> Tag.....	164
Global Attributes.....	165
Event Attributes .....	165
Browser Support .....	165
HTML <aside> Tag .....	165

Description.....	165
HTML <audio> Tag.....	166
HTML <b> Tag .....	167
HTML <base> Tag .....	168
HTML <basefont> Tag.....	169
HTML <bdo> Tag .....	171
HTML <bdi> Tag.....	172
HTML <bgsound> Tag .....	173
HTML <big> Tag.....	174
HTML blink Tag.....	175
HTML <blockquote> Tag .....	176
HTML <body> Tag.....	177
HTML   Tag .....	178
HTML Button Tag .....	179
HTML <canvas> Tag.....	181
HTML <caption> Tag.....	182
HTML <center> Tag .....	184
HTML <cite> Tag.....	184
HTML <code> Tag .....	185
HTML <col> Tag.....	186
HTML colgroup Tag .....	188
HTML <comment> and <!--.....--> Tag .....	190
HTML <datalist> Tag.....	191
HTML <dd> Tag .....	192
HTML <del> Tag.....	193
HTML <dfn> Tag .....	194
HTML <dialog> tag .....	195

HTML <dir> Tag .....	196
HTML div Tag.....	198
HTML <dl> Tag.....	199
HTML <dt> Tag .....	201
HTML <em> Tag.....	202
HTML <embed> Tag.....	203
HTML <fieldset> Tag.....	204
HTML Figcaption Tag .....	206
HTML Figure Tag.....	207
HTML <font> Tag .....	208
HTML Footer Tag .....	209
HTML <form> Tag .....	210
HTML <frame> Tag .....	212
HTML <frameset> Tag .....	214
HTML <h1> to <h6> Tag.....	216
HTML <head> Tag.....	217
HTML Header Tag .....	218
HTML <hr> Tag .....	219
HTML <html> Tag .....	220
HTML <i> Tag.....	221
HTML <iframe> Tag .....	222
HTML <ilayer> Tag.....	224
HTML <img> Tag.....	227
HTML <input> Tag .....	229
HTML <ins> Tag .....	233
HTML <isindex> tag .....	235
HTML <kbd> Tag.....	236

HTML keygen Tag .....	236
HTML <label> Tag .....	238
HTML <layer> Tag.....	239
HTML <legend> Tag .....	242
HTML <li>Tag .....	243
HTML <link> Tag.....	245
HTML Main Tag .....	248
HTML <map> Tag.....	249
HTML Mark Tag .....	250
HTML <marquee> Tag .....	251
HTML <menu> Tag .....	253
HTML <menuitem> tag.....	254
HTML <meta> tag .....	256
HTML <meter> Tag .....	257
HTML <multicol> tag .....	259
HTML <nav> Tag.....	260
HTML <noabr> Tag .....	261
HTML <noembed> Tag .....	262
HTML <noframes> Tag.....	263
HTML <noscript> Tag.....	264
HTML <object> Tag.....	265
HTML <ol> Tag.....	267
HTML <optgroup> Tag .....	269
HTML <option> Tag .....	270
HTML <output> Tag.....	272
HTML <p> Tag .....	273
HTML <param> Tag .....	274

HTML <plaintext> Tag .....	276
HTML <pre> Tag .....	276
HTML Progress Tag.....	278
HTML <q> Tag .....	279
HTML Rp Tag .....	280
HTML Rt Tag .....	281
HTML Ruby Tag .....	282
HTML <strike> Tag.....	283
HTML Phrase Elements.....	284
HTML <script> Tag .....	286
HTML Section Tag.....	288
HTML <select> Tag .....	289
HTML <spacer> Tag .....	291
HTML <small> Tag .....	292
HTML <source> tag.....	293
HTML <span> Tag .....	294
HTML <strike> Tag.....	295
HTML <strong> tag .....	296
HTML <style> tag.....	297
HTML <sub> Tag .....	298
HTML Summary Tag .....	299
HTML <sup> Tag .....	300
HTML <table> Tag .....	301
HTML <tbody> Tag .....	303
HTML <td> Tag .....	306
HTML textarea Tag .....	309
HTML <tfoot> Tag.....	311

HTML <th> Tag .....	314
HTML <thead> Tag .....	316
HTML <time> tag .....	319
HTML <title> Tag .....	320
HTML <tr> Tag .....	321
HTML <track> tag .....	323
HTML <tt> Tag .....	324
HTML <u> Tag .....	325
HTML <ul> Tag .....	326
HTML <var> Tag .....	328
HTML <video> Tag .....	328
HTML <wbr> Tag .....	330
HTML <xmp> Tag .....	331
29. HTML – ATTRIBUTE REFERENCE .....	333
Global Attributes .....	333
Language Attributes .....	334
30. HTML EVENTS REFERENCE .....	335
Window Events Attributes .....	335
Form Events .....	336
Keyboard Events .....	337
Mouse Events .....	337
Media Events .....	338
31. HTML – FONTS REFERENCE .....	341
Fonts for Microsoft Systems .....	342
Fonts for Macintosh Systems .....	343
Fonts for Unix Systems .....	344

HTML ASCII Codes .....	344
7-BIT Printable ASCII Characters .....	344
7-BIT ASCII Device Control Characters .....	348
32. ASCII TABLE LOOKUP .....	351
7 Bit ASCII Codes .....	351
Extended ASCII Codes.....	356
33. HTML – COLOR NAMES .....	363
34. HTML – ENTITIES .....	370
Other Entities Supported by HTML Browsers .....	375
35. MIME MEDIA TYPES .....	377
36. HTML – URL ENCODING .....	393
ASCII Control Characters Encoding .....	394
Non-ASCII control characters encoding .....	395
Reserved Characters Encoding .....	402
Unsafe Characters Encoding .....	403
37. LANGUAGE ISO CODES .....	405
Language Codes: ISO 639, Microsoft .....	405
Language Codes: ISO 639, Macintosh .....	412
38. HTML – CHARACTER ENCODINGS.....	420
39. HTML – DEPRECATED TAGS.....	422
HTML Deprecated Attributes.....	422

# 1. HTML – OVERVIEW

HTML stands for **H**yper**t**ext **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.

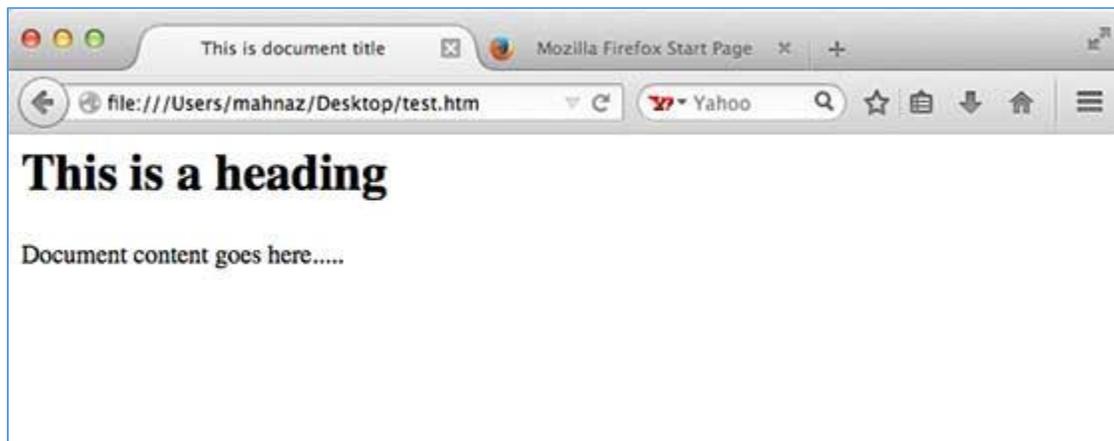
## Basic HTML Document

---

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>
```

Either you can use **Try it** option available at the top right corner of the code box to check the result of this HTML code, or let's save it in an HTML file **test.htm** using your favorite text editor. Finally open it using a web browser like Internet Explorer or Google Chrome, or Firefox etc. It must show the following output:



## 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:

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<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.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
<h1>	This tag represents the heading.

<p>	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.

## HTML Document Structure

---

A typical HTML document will have the following structure:

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

  <body>
    Document body related tags
  </body>
</html>
```

We will study all the header and body tags in subsequent chapters, but for now let's see what is document declaration tag.

## 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.

# 2. 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>
```

This will produce the following result:

# This is heading 1

## This is heading 2

### This is heading 3

#### This is heading 4

##### This is heading 5

###### This is heading 6

## 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 **<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>
```

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 the 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

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>
```

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 **&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>
```

# 3. 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
<p>	This is paragraph content.	</p>
<h1>	This is heading content.	</h1>
<div>	This is division content.	</div>

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 **<br />** 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.

## Nested HTML Elements

---

It is very much allowed to keep one HTML element inside another HTML element:

### Example

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

```
<head>
```

```
<title>Nested Elements Example</title>  
</head>  
<body>  
<h1>This is <i>italic</i> heading</h1>  
<p>This is <u>underlined</u> paragraph</p>  
</body>  
</html>
```

This will display the following result:

```
This is italic heading  
This is underlined paragraph
```

# 4. HTML – 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 **<p>** 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 display 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**

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

## 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.

- 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)

### Example

```
<!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. 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.

## Example

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

## 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

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. For a complete list of HTML Tags and related attributes please check reference to [HTML Tags List](#).

# 5. 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>
```

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

```
<!DOCTYPE html>
<html>
<head>
<title>Italic Text Example</title>
</head>
```

```
<body>
<p>The following word uses a <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

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

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

```
<title>Strike Text Example</title>
</head>
<body>
<p>The following word uses a <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

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

This will produce the following result:

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

## 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>
```

This will produce the following result:

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

## Inserted Text

---

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

### Example

```
<!DOCTYPE html>
<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 `<del>...</del>` element, is displayed as deleted text.

### Example

```
<!DOCTYPE html>
<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 ~~eola~~ 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

```
<!DOCTYPE html>
<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

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

## Grouping Content

The **<div>** and **<span>** elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a `<div>` element to indicate that all of the elements within that `<div>` element relate to the footnotes. You might then attach a style to this `<div>` element so that they appear using a special set of style rules.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Div Tag Example</title>
</head>
<body>
<div id="menu" align="middle" >
<a href="/index.htm">HOME</a> |
<a href="/about/contact_us.htm">CONTACT</a> |
<a href="/about/index.htm">ABOUT</a>
</div>

<div id="content" align="left" bgcolor="white">
<h5>Content Articles</h5>
<p>Actual content goes here.....</p>
</div>
</body>
</html>
```

This will produce the following result:

[HOME](#) | [CONTACT](#) | [ABOUT](#)

## CONTENT ARTICLES

Actual content goes here.....

The `<span>` element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the `<span>` element as follows

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Span Tag Example</title>
</head>
<body>
<p>This is the example of <span style="color:green">span tag</span> and the <span
style="color:red">div tag</span> alongwith CSS</p>
</body>
</html>
```

This will produce the following result:

This is the example of span tag and the div tag along with CSS

These tags are commonly used with CSS to allow you to attach a style to a section of a page.

# 6. 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 **<b>**, **<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 **<em>...</em>** 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

```
<!DOCTYPE html>
<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 **<strong>...</strong>** element is displayed as important text.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Strong Text Example</title>
</head>
<body>
<p>The following word uses a <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

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

```

<title>Text Abbreviation</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

```

<!DOCTYPE html>
<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

```
<!DOCTYPE html>
<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

```
<!DOCTYPE html>
<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

```
<!DOCTYPE html>
<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

```
<!DOCTYPE html>
<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

```
<!DOCTYPE html>
<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

```
<!DOCTYPE html>
<html>
<head>
<title>Computer Code 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

```
<!DOCTYPE html>
<html>
<head>
<title>Keyboard Text 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

```
<!DOCTYPE html>
<html>
<head>
<title>Variable Text 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

```
<!DOCTYPE html>
<html>
<head>
<title>Program Output 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!

## 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>
```

This will produce the following result:

*388A, Road No 22, Jubilee Hills - Hyderabad*

# 7. HTML – META TAGS

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

## Adding Meta Tags to Your Documents

---

You can add metadata to your web pages by placing `<meta>` tags inside the header of the document which is represented by `<head>` and `</head>` tags. A meta tag can have following attributes in addition to core attributes:

Attribute	Description
Name	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
content	Specifies the property's value.
scheme	Specifies a scheme to interpret the property's value (as declared in the content attribute).
http-equiv	Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

## Specifying Keywords

---

You can use `<meta>` tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

## Example

Following is an example, where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

This will produce the following result:

```
Hello HTML5!
```

## Document Description

---

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

## Example

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

## Document Revision Date

---

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Tutorialspoint, 3/7/2014" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

## Document Refreshing

---

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

### Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Tutorialspoint, 3/7/2014" />
<meta http-equiv="refresh" content="5" />
```

```
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

## Page Redirection

---

You can use `<meta>` tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

### Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Tutorialspoint, 3/7/2014" />
<meta http-equiv="refresh" content="5; url=http://www.tutorialspoint.com" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

## Setting Cookies

---

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.

You can use `<meta>` tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

## Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Tutorialspoint, 3/7/2014" />
```

End of ebook preview  
If you liked what you saw...  
Buy it from our store @ <https://store.tutorialspoint.com>