



HTML-5

hypertext markup language

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

About the Tutorial

HTML5 is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a markup language. In this tutorial, we will discuss the features of HTML5 and how to use it in practice.

Audience

This tutorial has been designed for beginners in *HTML5* to make them understand the basic-to-advanced concepts of the subject.

Prerequisites

Before starting this tutorial, you should have a basic understanding of HTML and its tags.

Disclaimer & Copyright

© Copyright 2016 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

Execute HTML5 Online

For most of the examples given in this tutorial you will find Try it option, so just make use of this option to execute your HTML5 programs at the spot and enjoy your learning.

Try following example using Try it option available at the top right corner of the below sample code box –

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8">
    <title>Tutorials Point</title>
  </head>

  <body>
    <header role="banner">
      <h1>HTML5 Document Structure Example</h1>
      <p>This page should be tried in safari, chrome or Mozilla.</p>
    </header>
    <footer>
      <p>Created by <a href="http://tutorialspoint.com/">Tutorials
Point</a></p>
    </footer>
  </body>

</html>
```

Table of Contents

About the Tutorial.....	1
Audience	1
Prerequisites	1
Execute HTML5 Online	2
Table of Contents	3
1. HTML5 – OVERVIEW	9
Browser Support	9
New Features	9
Backward Compatibility	10
2. HTML5 – SYNTAX.....	11
The DOCTYPE	11
Character Encoding	11
The <script> tag.....	11
The <link> tag.....	12
HTML5 Elements	12
HTML5 Attributes.....	12
HTML5 Document	13
3. HTML5 – ATTRIBUTES.....	16
Standard Attributes.....	16
Custom Attributes	17
4. HTML5 – EVENTS.....	18
5. HTML5 – WEB FORMS 2.0	21

The <input> element in HTML4	21
The <input> element in HTML5	22
HTML5 - datetime	23
HTML5 - datetime local	24
HTML5 – date	24
HTML5 – month.....	25
HTML5 - week	26
HTML5 – time.....	27
HTML5 – number.....	27
HTML5 – range	28
HTML5 - email	29
HTML5 – URL.....	30
The <output> element.....	30
The placeholder attribute.....	32
The autofocus attribute	32
The required attribute.....	33
6. HTML5 – SVG.....	35
Viewing SVG Files	35
Embedding SVG in HTML5	35
HTML5 – SVG Circle	36
HTML5 – SVG Rectangle	36
HTML5 – SVG Line	37
HTML5 – SVG Ellipse.....	38
HTML5 – SVG Polygon	39
HTML5 – SVG Polyline	40

HTML5 – SVG Gradients	40
HTML5 – SVG Star	42
7. HTML5 – MATHML	44
MathML Examples	44
Using MathML Characters	45
Matrix Presentation Examples	47
8. HTML5 – WEB STORAGE.....	49
Session Storage	49
Local Storage.....	50
Delete Web Storage	51
9. HTML5 – WEB SQL DATABASE	54
The Core Methods.....	54
Opening Database.....	54
Executing queries	55
INSERT Operation.....	55
READ Operation	55
Final Example	56
10. HTML5 – SERVER SENT EVENTS.....	58
Web Application for SSE.....	58
Server Side Script for SSE.....	59
Handle Server-Sent Events	60
11. HTML5 – WEBSOCKETS	61
WebSocket Attributes	61

WebSocket Events.....	62
WebSocket Methods.....	62
WebSocket Example.....	62
Client Side HTML & JavaScript Code	62
Install pywebsocket	64
Start the Server	64
 12. HTML5 – CANVAS.....	 65
The Rendering Context.....	66
Browser Support	66
HTML5 Canvas Examples	66
HTML5 Canvas - Drawing Rectangles.....	67
HTML5 Canvas - Drawing Paths	69
HTML5 Canvas - Drawing Lines.....	72
HTML5 Canvas - Drawing Bezier Curves.....	76
HTML5 Canvas - Drawing Quadratic Curves.....	78
HTML5 Canvas - Using Images	81
HTML5 Canvas - Create Gradients	83
HTML5 Canvas - Styles and Colors	88
HTML5 Canvas - Text and Fonts.....	92
HTML5 Canvas - Pattern and Shadow.....	94
HTML5 Canvas - Save and Restore States	98
HTML5 Canvas - Translation	101
HTML5 Canvas - Rotation	104
HTML5 Canvas - Scaling.....	106
HTML5 Canvas - Transforms	109

HTML5 Canvas - Composition	112
HTML5 Canvas - Animations	116
13. HTML5 – AUDIO & VIDEO	119
Embedding Video	119
Video Attribute Specification	120
Embedding Audio	121
Audio Attribute Specification	122
Handling Media Events.....	122
Configuring Servers for Media Type	124
14. HTML5 – GEOLOCATION.....	125
Geolocation Methods.....	125
Geolocation <code>getCurrentPosition()</code> API	126
Geolocation <code>watchPosition()</code> API	128
Geolocation <code>clearWatch()</code> API	130
Location Properties	132
Handling Errors	133
Position Options.....	134
15. HTML5 – MICRODATA	136
Example	136
Global Attributes.....	137
Properties Datatypes.....	137
Microdata API support	138
Defining Microdata Vocabulary	138
16. HTML5 – DRAG & DROP	140

Drag and Drop Events.....	140
The DataTransfer Object:	141
Drag and Drop Process	142
17. HTML5 – WEB WORKERS.....	146
What is Web Workers?.....	147
How Web Workers Work?.....	148
Stopping Web Workers	149
Handling Errors	149
Checking for Browser Support.....	150
18. HTML5 – INDEXEDDB	152
19. HTML5 – WEB MESSAGING	160
20. HTML5 – CORS	163
21. HTML5 – WEB RTC.....	167

1. HTML5 – OVERVIEW

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

Browser Support

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- **New Semantic Elements:** These are like `<header>`, `<footer>`, and `<section>`.
- **Forms 2.0:** Improvements to HTML web forms where new attributes have been introduced for `<input>` tag.
- **Persistent Local Storage:** To achieve without resorting to third-party plugins.
- **WebSocket** : A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events:** HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas:** This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video:** You can embed audio or video on your webpages without resorting to third-party plugins.

- **Geolocation:** Now visitors can choose to share their physical location with your web application.
- **Microdata:** This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop:** Drag and drop the items from one location to another location on the same webpage.

Backward Compatibility

HTML5 is designed, as much as possible, to be backward compatible with existing web browsers. Its new features have been built on existing features and allow you to provide fallback content for older browsers.

It is suggested to detect support for individual HTML5 features using a few lines of JavaScript.

If you are not familiar with any previous version of HTML, I would recommend that you go through our **HTML Tutorial** before exploring the features of HTML5.

2. HTML5 – SYNTAX

The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.

HTML 5 does not have the same syntax rules as XHTML where we needed lower case tag names, quoting our attributes, an attribute had to have a value and to close all empty elements.

HTML5 comes with a lot of flexibility and it supports the following features –

- Uppercase tag names.
- Quotes are optional for attributes.
- Attribute values are optional.
- Closing empty elements are optional.

The DOCTYPE

DOCTYPES in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.

HTML 5 authors would use simple syntax to specify DOCTYPE as follows:

```
<!DOCTYPE html>
```

The above syntax is case-insensitive.

Character Encoding

HTML 5 authors can use simple syntax to specify Character Encoding as follows –

```
<meta charset="UTF-8">
```

The above syntax is case-insensitive.

The <script> tag

It's common practice to add a type attribute with a value of "text/javascript" to script elements as follows –

```
<script type="text/javascript" src="scriptfile.js"></script>
```

HTML 5 removes extra information required and you can use simply following syntax –

```
<script src="scriptfile.js"></script>
```

The <link> tag

So far you were writing <link> as follows –

```
<link rel="stylesheet" type="text/css" href="stylefile.css">
```

HTML 5 removes extra information required and you can simply use the following syntax –

```
<link rel="stylesheet" href="stylefile.css">
```

HTML5 Elements

HTML5 elements are marked up using start tags and end tags. Tags are delimited using angle brackets with the tag name in between.

The difference between start tags and end tags is that the latter includes a slash before the tag name.

Following is the example of an HTML5 element –

```
<p>...</p>
```

HTML5 tag names are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Most of the elements contain some content like <p>...</p> contains a paragraph. Some elements, however, are forbidden from containing any content at all and these are known as void elements. For example, **br**, **hr**, **link**, **meta**, etc.

Here is a complete list of [HTML5 Elements](#).

HTML5 Attributes

Elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.

Following is the example of an HTML5 attribute which illustrates how to mark up a **div** element with an attribute named class using a value of "example" –

```
<div class="example">...</div>
```

Attributes may only be specified within start tags and must never be used in end tags.

HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Here is a complete list of [HTML5 Attributes](#).

HTML5 Document

The following tags have been introduced for better structure –

- **section:** This tag represents a generic document or application section. It can be used together with h1-h6 to indicate the document structure.
- **article:** This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.
- **aside:** This tag represents a piece of content that is only slightly related to the rest of the page.
- **header:** This tag represents the header of a section.
- **footer:** This tag represents a footer for a section and can contain information about the author, copyright information, et cetera.
- **nav:** This tag represents a section of the document intended for navigation.
- **dialog:** This tag can be used to mark up a conversation.
- **figure:** This tag can be used to associate a caption together with some embedded content, such as a graphic or video.

The markup for an HTML 5 document would look like the following –

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8">
    <title>...</title>
  </head>

  <body>
    <header>...</header>
    <nav>...</nav>

    <article>
```

```
    <section>
        ...
    </section>
</article>

<aside>...</aside>
<footer>...</footer>
</body>
</html>
```

```
<!DOCTYPE html>

<html>

    <head>
        <meta charset="utf-8">
        <title>...</title>
    </head>

    <body>

        <header role="banner">
            <h1>HTML5 Document Structure Example</h1>
            <p>This page should be tried in safari, chrome or Mozilla.</p>
        </header>

        <nav>

            <ul>
                <li><a href="http://www.tutorialspoint.com/html">HTML
Tutorial</a></li>
                <li><a href="http://www.tutorialspoint.com/css">CSS Tutorial</a></li>
                <li><a href="http://www.tutorialspoint.com/javascript">JavaScript
Tutorial</a></li>
            </ul>
```

```
</nav>

<article>
  <section>
    <p>Once article can have multiple sections</p>
  </section>
</article>

<aside>
  <p>This is  aside part of the web page</p>
</aside>

<footer>
  <p>Created by <a href="http://tutorialspoint.com/">Tutorials
Point</a></p>
</footer>

</body>
</html>
```

It will produce the following result –

HTML5 Document Structure Example

This page should be tried in safari, chrome or Mozilla.

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)

Once article can have multiple sections

This is aside part of the web page



Created by [Tutorials Point](#)

3. HTML5 – ATTRIBUTES

As explained in the previous chapter, elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.

Following is the example of an HTML5 attributes which illustrates how to mark up a div element with an attribute named class using a value of "example" –

```
<div class="example">...</div>
```

Attributes may only be specified within **start tags** and must never be used in **end tags**.

HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Standard Attributes

The attributes listed below are supported by almost all the HTML 5 tags.

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexadecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.
contextmenu	Menu id	Specifies the context menu for an element.
data-XXXX	User Defined	Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data-".
draggable	true,false, auto	Specifies whether or not a user is allowed to drag an element.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.

id	User Defined	Names an element for use with Cascading Style Sheets.
item	List of elements	Used to group elements.
itemprop	List of items	Used to group items.
spellcheck	true, false	Specifies if the element must have it's spelling or grammar checked.
style	CSS Style sheet	Specifies an inline style for an element.
subject	User define id	Specifies the element's corresponding item.
tabindex	Tab number	Specifies the tab order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

For a complete list of HTML5 Tags and related attributes, please check our reference to [HTML5 Tags](#).

Custom Attributes

A new feature being introduced in HTML 5 is the addition of custom data attributes.

A custom data attribute starts with **data-** and would be named based on your requirement. Here is a simple example –

```
<div class="example" data-subject="physics" data-level="complex">
    ...
</div>
```

The above code will be perfectly valid HTML5 with two custom attributes called *data-subject* and *data-level*. You would be able to get the values of these attributes using JavaScript APIs or CSS in similar way as you get for standard attributes.

4. HTML5 – EVENTS

When users visit your website, they perform various activities such as clicking on text and images and links, hover over defined elements, etc. These are examples of what JavaScript calls **events**.

We can write our event handlers in Javascript or VBscript and you can specify these event handlers as a value of event tag attribute. The HTML5 specification defines various event attributes as listed below –

We can use the following set of attributes to trigger any **javascript** or **vbscript** code given as value, when there is any event that takes place for any HTML5 element.

We would cover element-specific events while discussing those elements in detail in subsequent chapters.

Attribute	Value	Description
offline	script	Triggers when the document goes offline
onabort	script	Triggers on an abort event
onafterprint	script	Triggers after the document is printed
onbeforeunload	script	Triggers before the document loads
onbeforeprint	script	Triggers before the document is printed
onblur	script	Triggers when the window loses focus
oncanplay	script	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	script	Triggers when media can be played to the end, without stopping for buffering
onchange	script	Triggers when an element changes
onclick	script	Triggers on a mouse click
oncontextmenu	script	Triggers when a context menu is triggered
ondblclick	script	Triggers on a mouse double-click
ondrag	script	Triggers when an element is dragged
ondragend	script	Triggers at the end of a drag operation
ondragenter	script	Triggers when an element has been dragged to a valid drop target
ondragleave	script	Triggers when an element leaves a valid drop target
ondragover	script	Triggers when an element is being dragged over a valid drop target
ondragstart	script	Triggers at the start of a drag operation
ondrop	script	Triggers when dragged element is being dropped
ondurationchange	script	Triggers when the length of the media is changed
onemptied	script	Triggers when a media resource element suddenly becomes empty.
onended	script	Triggers when media has reach the end
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form changes

onforminput	script	Triggers when a form gets user input
onhaschange	script	Triggers when the document has change
oninput	script	Triggers when an element gets user input
oninvalid	script	Triggers when an element is invalid
onkeydown	script	Triggers when a key is pressed
onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded
onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded
onloadstart	script	Triggers when the browser starts to load the media data
onmessage	script	Triggers when the message is triggered
onmousedown	script	Triggers when a mouse button is pressed
onmousemove	script	Triggers when the mouse pointer moves
onmouseout	script	Triggers when the mouse pointer moves out of an element
onmouseover	script	Triggers when the mouse pointer moves over an element
onmouseup	script	Triggers when a mouse button is released
onmousewheel	script	Triggers when the mouse wheel is being rotated
onoffline	script	Triggers when the document goes offline
onoine	script	Triggers when the document comes online
ononline	script	Triggers when the document comes online
onpagehide	script	Triggers when the window is hidden
onpageshow	script	Triggers when the window becomes visible
onpause	script	Triggers when media data is paused
onplay	script	Triggers when media data is going to start playing
onplaying	script	Triggers when media data has start playing
onpopstate	script	Triggers when the window's history changes
onprogress	script	Triggers when the browser is fetching the media data
onratechange	script	Triggers when the media data's playing rate has changed
onreadystatechange	script	Triggers when the ready-state changes
onredo	script	Triggers when the document performs a redo
onresize	script	Triggers when the window is resized
onscroll	script	Triggers when an element's scrollbar is being scrolled
onseeked	script	Triggers when a media element's seeking attribute is no longer true, and the seeking has ended
onseeking	script	Triggers when a media element's seeking attribute is true, and the seeking has begun
onselect	script	Triggers when an element is selected
onstalled	script	Triggers when there is an error in fetching media data
onstorage	script	Triggers when a document loads
onsubmit	script	Triggers when a form is submitted
onsuspend	script	Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdate	script	Triggers when media changes its playing position
onundo	script	Triggers when a document performs an undo
onunload	script	Triggers when the user leaves the document
onvolumechange	script	Triggers when media changes the volume, also when volume is set to "mute"

onwaiting	script	Triggers when media has stopped playing, but is expected to resume
-----------	--------	--

5. HTML5 – WEB FORMS 2.0

Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and free us from a great deal of tedious scripting and styling that was required in HTML4.

The <input> element in HTML4

HTML4 input elements use the **type** attribute to specify the data type. HTML4 provides following types –

Type	Description
text	A free-form text field, nominally free of line breaks.
password	A free-form text field for sensitive information, nominally free of line breaks.
checkbox	A set of zero or more values from a predefined list.
radio	An enumerated value.
submit	A free form of button initiates form submission.
file	An arbitrary file with a MIME type and optionally a file name.
image	A coordinate, relative to a particular image's size, with the extra semantic that it must be the last value selected and initiates form submission.
hidden	An arbitrary string that is not normally displayed to the user.
select	An enumerated value, much like the radio type.
textarea	A free-form text field, nominally with no line break restrictions.
button	A free form of button which can initiates any event related to button.

Following is the simple example of using labels, radio buttons, and submit buttons –

```
...  
<form action="http://example.com/cgiscript.pl" method="post">  
  
  <p>  
  
    <label for="firstname">first name: </label>  
    <input type="text" id="firstname"><br />  
  
    <label for="lastname">last name: </label>  
    <input type="text" id="lastname"><br />  
  
    <label for="email">email: </label>  
    <input type="text" id="email"><br>
```

```

    <input type="radio" name="sex" value="male"> Male<br>
    <input type="radio" name="sex" value="female"> Female<br>
    <input type="submit" value="send"> <input type="reset">

  </p>
</form>

...

```

The <input> element in HTML5

Apart from the above-mentioned attributes, HTML5 input elements introduced several new values for the **type** attribute. These are listed below.

NOTE: Try all the following example using latest version of **Opera** browser.

Type	Description
datetime	A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.
datetime-local	A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information.
date	A date (year, month, day) encoded according to ISO 8601.
month	A date consisting of a year and a month encoded according to ISO 8601.
week	A date consisting of a year and a week number encoded according to ISO 8601.
time	A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601.
number	It accepts only numerical value. The step attribute specifies the precision, defaulting to 1.
range	The range type is used for input fields that should contain a value from a range of numbers.
email	It accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.
url	It accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in http://www.example.com format or in http://example.com format.

HTML5 - datetime

A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.

Example

```
<!DOCTYPE HTML>
```



```
<html>
  <body>

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Date and Time : <input type = "datetime" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

Output

Date and Time :

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