

Prerequisites

HTML5 is a relatively young specification, and as a result, browser support is quite limited (at the time of writing). The code presented in this tutorial is built to be as cross-browser compatible as possible, but some features will not work in all browsers. Any features that are currently browser-specific will be clearly identified in the tutorial. To ensure that you can experience all of these new features, it is recommended that you install the latest versions of the following Web browsers on your system when developing HTML5 and CSS3 applications:

- [Mozilla Firefox \(version 3.5+\)](#)
- [Apple Safari \(version 4.0+\)](#)
- [Opera \(version 10.0+\)](#)
- [Google Chrome \(version 3.0+\)](#)

You do not need any specific software to write HTML and CSS code; any basic text editor will do (such as Notepad, vi, emacs, and so on.) In this tutorial, it is assumed that the [source code](#) is stored in a directory on your local computer—you do not need to use a Web server or upload the files to a Web hosting service.

New features in HTML5

In this section, you will discover some of the great new features that HTML5 has to offer. You will first learn about the new semantic elements that aim to give meaning to the various parts of a modern Web page: headers, footers, navigation bars, sidebars, and so forth. Next, you will learn about the important new `<canvas>` element and the 2D drawing JavaScript APIs that you can use to create shapes, text, animations, transitions, and more. Following this, you will see how the new `<audio>` and `<video>` elements intend on replacing the Web's current dependency on Flash as a multimedia delivery platform. Next, you will be introduced to the local storage APIs and offline applications support that will further bring Web applications in line with their desktop counterparts in terms of functionality, even when not connected to a network or the Internet. This section is wrapped up with a brief overview of the other new elements, attributes, and APIs that are proposed in the current HTML5 specification.

Develop skills on this topic

This content is part of a progressive knowledge path for advancing your skills

Semantic elements

The HTML5 specification includes a series of new semantic elements that are used to give some meaning to the various sections or parts of a Web page, such as a header, footer, navigation, and so on. In previous versions of HTML, you would typically use `<div>` elements to create these parts, using ID or class attributes to differentiate them from each other. The problem with this is that this has no semantic meaning, as there are no strict rules defined that specify what class names or IDs are to be used, making it extremely difficult for software to determine what the particular area is doing. HTML5 should help alleviate these issues, making it easier for Web browsers to parse the semantic structure of a document.

It is worth pointing out that continuing to use `<div>` elements in HTML5 is perfectly valid, but in order to future-proof your work, it is recommended that you use semantic elements where relevant. On the other side of the coin, it is also suggested that you avoid using these new elements for purposes other than their intended. For example,

the <nav> element should not be used for just any group of links; it is intended to surround the main navigation block on the page.

The main semantic elements that HTML5 introduces are:

<header>

This element is used to define a header for some part of a Web page, be it the entire page, an <article> element, or a <section> element.

<footer>

Like the <header> element, this new element defines a footer for some part of a page. A footer does not have to be included at the end of a page, article, or section, but it typically does.

<nav>

This is a container for the primary navigation links on a Web page. This element is not intended for use with all groups of links and should be used for major navigation blocks only. If you have a <footer> element that contains navigation links, you do not need to wrap these links in a <nav> element, since the <footer> element will suffice on its own.

<article>

The <article> element is used to define an independent item on the page that can be distributed on its own, such as a news item, blog post, or comment. Such items are typically syndicated using RSS feeds.

<section>

This element represents a section of a document or application, such as a chapter or a section of an article or tutorial. For example, the section you are reading now could be surrounded by a <section> element in HTML5. <section> elements typically have a header, although it is not strictly required. The header for the section you are reading now would contain the text "Semantic elements," for example.

<aside>

This new element can be used to mark up a sidebar or some other content that is considered somewhat separate to the content around it. An example of this might be advertising blocks.

<hgroup>

In some cases, a page, article, or section may require more than one heading, such as where you have a title and a subtitle. This tutorial, for example, has the title "Create modern Web sites using HTML5 and CSS3" and the subtitle "Implementing the canvas and video elements in HTML5." You could wrap these in an <hgroup> element, using an <h1> element for the main title and an <h2> element for the subtitle.

The sample Web site at the end of this tutorial includes several of these new semantic elements, and I will explain their syntax and use in more detail at that point.

The <canvas> element

The <canvas> element was originally developed by Apple® for use in Mac OS X Dashboard widgets and in Safari, but was later adopted by Mozilla® and Opera® in their Web browsers. The element has been standardized and included in the HTML5

specification, along with a series of 2D drawing APIs that can be used to create shapes, text, transitions, and animations inside the element.

Many believe that the `<canvas>` element is one of the most important aspects of HTML5 as it facilitates the production of graphs, interactive games, paint applications, and other graphics on the fly without requiring external plug-ins such as Adobe Flash.

The `<canvas>` element itself is quite basic, defining the width, height, and unique ID for the object. The developer must then use a series of JavaScript APIs to actually draw objects on the canvas, typically when the Web page has finished rendering. These APIs allow the developer to draw shapes and lines; apply color, opacity, and gradients; create text; transform canvas objects; and perform animation. The APIs also allow the `<canvas>` to be interactive and respond to user input such as mouse events and key events, facilitating the production of games and Web applications on the canvas. You will see an example of the `<canvas>` element in action in the sample HTML5/CSS3 Web site later in this tutorial.

Playing `<audio>` and `<video>`

In recent years, the popularity of video sharing sites such as YouTube and content delivery platforms like Hulu has seen a huge explosion in the use of the Web for multimedia streaming. Unfortunately, the Web was not built with such content in mind, and as a result, the provision of video and audio has by and large been facilitated by the Flash Video (.flv) file format and the Adobe Flash platform.

HTML5, however, includes support for two new elements, `<audio>` and `<video>`, which allow Web developers to include multimedia content without relying on the user to have additional browser plug-ins installed. Several browsers, including Mozilla Firefox, Apple Safari, and Google Chrome, have begun supporting these new elements and providing standard browser playback controls, should the user choose to use them. In addition, a set of standard JavaScript APIs has been provided to allow developers to create their own playback controls, should they wish to do so. A key advantage to native multimedia playback is that it theoretically requires less CPU resources, which can lead to energy savings.

A key issue with these new multimedia elements, however, is the file formats supported by each browser and the patent licensing issues that go along with the various codecs that these files can be encoded with. Mozilla and Opera want to use the open-source Theora video container and codec, which does not require patent licensing for the inclusion of the codecs in the Web browser. On the other hand, Apple and Google are not happy with the quality of Theora, particularly for the delivery of high definition (HD) content on the likes of YouTube. They prefer the H.264 codec, typically contained in MP4, MOV, or MKV files.

The issue is not just with video, however, as the same problems reside with audio codecs. The MP3 and AAC formats are restricted by patents, whereas the Vorbis format is not. The problem with Vorbis audio is that it is not in widespread use, as portable media players and many media software applications do not support it.

There are many decisions to be made about HTML5 `<video>` and `<audio>` in the near future, and it will be interesting to see what codecs and formats are facilitated in the final recommendation. In the meantime, you can try to support all browsers by making video available in a variety of formats and by providing Flash video as a fallback. Let's hope that a final decision is made and that it is not left to browser vendors to decide which formats to support, as that would essentially render these new elements useless.

Again, you will see the `<video>` element in action later in this tutorial.

Local storage and offline applications

Web developers have traditionally used cookies to store information on a visitor's local machine, allowing a Web page to read this information back at a later point. While cookies are very useful for storing basic data, they are limited by the fact that Web browsers are not required to keep more than 20 cookies per Web server or more than 4KB of data per cookie (including both name and value). In addition, they are sent to the Web server with every HTTP request, which is a waste of resources.

HTML5 provides a solution for these problems with the Local Storage APIs, which are covered in a separate specification to the main HTML5 document. This set of APIs allows developers to store information on the visitor's computer while remaining reasonably confident that they will still be there at a later date. In addition, the information is accessible at any point (even after the page has rendered) and is not loaded automatically with each HTTP request. The specification includes same-origin restrictions, which prevent Web sites from reading or changing data stored by other Web sites.

Most browsers store Web pages in local cache, allowing them to be viewed even if the user is offline. This works fine for static pages, but it is not available for dynamic content that is typically database-driven, such as Gmail, Facebook, or Twitter. HTML5 provides support for offline applications, where the browser downloads all the files necessary to use the application offline, and when the user uses the application offline, the browser can allow any changes made in the process to be uploaded to the server when they reconnect to the Internet.

Web form enhancements

If you have created Web applications before, you are more than likely familiar with HTML's set of form controls, some of which are implemented using the `<input>` element. In HTML 4, the following input types were supported:

- button
- checkbox
- file
- hidden
- image
- password
- reset
- radio
- submit
- text

In addition, there are some other elements that are used in forms such as `<select>` and `<textarea>`. These form controls provide plenty of function for basic form fields such as name, phone number, and address—like you might find on a contact form. But, the Web as a platform has grown far beyond the stage where HTML forms are used to submit contact forms—now they are used to submit application data for server-side processing. As a result, Web application developers find themselves continually in need of some more sophisticated form controls, such as spinners, sliders, date/time pickers, color pickers, and so on.

In order to tap into these types of controls, developers needed to use an external JavaScript library that provided UI components, or else use an alternative development framework such as Adobe Flex, Microsoft Silverlight, or JavaFX.

HTML5 aims to fill some of the gaps left by its predecessor in this space by providing a whole range of new form input types:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

At the moment, support for these new form fields is quite limited. The Mobile Safari browser on the iPhone makes use of some of these new types to change the type of keyboard presented to the user (for example, with the e-mail type, the @ symbol and .com shortcuts will be shown). Also, Opera provides some new widgets for many of these controls, including a spinner for the number type and a calendar date picker for the date-related types. The most widely available type of these new offerings is the range type, which is rendered as a slider by Opera, Safari, and Google Chrome.

In addition to these new input types, HTML5 also supports two major new features for form fields. The first of these is autofocus, which tells a browser to automatically give focus to a particular form field when the page has rendered, without requiring JavaScript code to do so. The second enhancement is the placeholder attribute, which allows the developer to define the text that will appear in a textbox-based control when its contents are empty. An example of this would be a search box where the developer would prefer not to use a label outside the box itself. The placeholder attribute allows the developer to specify text that will show when the value of the control is empty and the control does not have focus.

What You Can Do with HTML

There are lot more things you can do with HTML.

- You can publish documents online with text, images, lists, tables, etc.
- You can access web resources such as images, videos or other HTML document via hyperlinks.
- You can create forms to collect user inputs like name, e-mail address, comments, etc.
- You can include images, videos, sound clips, flash movies, applications and other HTML documents directly inside an HTML document.
- You can create offline version of your website that work without internet.
- You can store data in the user's web browser and access later on.
- You can find the current location of your website's visitor.

New Input Types in HTML5

HTML5 introduces several new `<input>` types like email, date, time, color, range, and so on. to improve the user experience and to make the forms more interactive. However, if a browser failed to recognize these new input types, it will treat them like a normal text box.

In this section we're going to take a brief look at each of the following new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel

- `time`
- `url`
- `week`

There was also a `datetime` input type for entering a date and time, but it is now obsolete.

Input Type Color

The `color` input type allows the user to select a color from a color picker and returns the color value in hexadecimal format (`#rrggbb`). If you don't specify a value, the default is `#000000`, which is black.

Let's try out the following example to understand how it basically works:

Example

Try this code »

```
<form>
  <label for="mycolor">Select Color:</label>
  <input type="color" value="#00ff00" id="mycolor">
</form>
```

Note: The color input (i.e. `type="color"`) is supported in all major modern web browsers such as Firefox, Chrome, Opera, Safari (12.1+), Edge (14+). Not supported by the Microsoft Internet Explorer and older version of Apple Safari browsers.

Input Type Date

The `date` input type allows the user to select a date from a drop-down calendar.

The date value includes the year, month, and day, but not the time.

Example

Try this code »

```
<form>
  <label for="mydate">Select Date:</label>
  <input type="date" value="2019-04-15" id="mydate">
</form>
```

Note: The date input (i.e. type="date") is supported by the Chrome, Firefox, Opera and Edge browsers. Not supported by the Internet Explorer and Safari browsers.

Input Type Datetime-local

The `datetime-local` input type allows the user to select both local date and time, including the year, month, and day as well as the time in hours and minutes.

Let's try out the following example to understand how it basically works:

Example

Try this code »

```
<form>
  <label for="mydatetime">Choose Date and Time:</label>
  <input type="datetime-local" id="mydatetime">
</form>
```

Warning: The input type="datetime-local" is not supported by Firefox, Safari, and Internet Explorer browsers. Currently supported by Chrome, Edge, and Opera browsers.

Input Type Email

The `email` input type allows the user to enter e-mail address. It is very similar to a standard text input type, but if it is used in combination with the `required` attribute, the browser may look for the patterns to ensure a properly-formatted e-mail address should be entered.

Let's try out this example by entering any e-mail address to see how it actually works:

Example

Try this code »

```
<form>
  <label for="myemail">Enter Email Address:</label>
  <input type="email" id="myemail" required>
</form>
```

Tip: You can style the email input field for different validation states, when an value is entered using the `:valid`, `:invalid` OR `:required` pseudo-classes.

Note: The validation for the email input (i.e. `type="email"`) is supported by all major browsers like Firefox, Chrome, Safari, Opera, Internet Explorer 10 and above.

Input Type Month

The `month` input type allows the user to select a month and year from a drop-down calendar.

The value is a string in the format "YYYY-MM", where YYYY is the four-digit year and MM is the month number. Let's try out an example to see how this basically works:

Example

Try this code »

```
<form>
  <label for="mymonth">Select Month:</label>
  <input type="month" id="mymonth">
</form>
```

Warning: The `input type="month"` is not supported by Firefox, Safari and Internet Explorer browsers. Currently supported in Chrome, Edge, and Opera browsers.

Input Type Number

The `number` input type can be used for entering a numerical value. You can also restrict the user to enter only acceptable values using the additional attributes `min`, `max`, and `step`.

The following example will allow you to enter a numeric value between 1 to 10.

Example

Try this code »

```
<form>
  <label for="mynumber">Enter a Number:</label>
  <input type="number" min="1" max="10" step="0.5"
id="mynumber">
</form>
```

Note: The number input (i.e. `type="number"`) is supported by all major web browsers such as Firefox, Chrome, Safari, Opera, Internet Explorer 10 and above. Internet Explorer however recognized the number but do not provide increment and decrement spin buttons.

Input Type Range

The `range` input type can be used for entering a numerical value within a specified range. It works very similar to `number` input, but it offers a simpler control for entering a number.

Let's try out the following example to understand how it basically works:

Example

Try this code »

```
<form>
  <label for="mynumber">Select a Number:</label>
  <input type="range" min="1" max="10" step="0.5"
id="mynumber">
</form>
```

Note: The range input (i.e. type="range") is supported by all major web browsers such as Firefox, Chrome, Safari, Opera, Internet Explorer 10 and above.

Input Type Search

The `search` input type can be used for creating search input fields.

A search field typically behaves like a regular text field, but in some browsers like Chrome and Safari as soon as you start typing in the search box a small cross appears on the right side of the field that lets you quickly clear the search field. Let's try out an example to see how it works:

Example

Try this code »

```
<form>
  <label for="mysearch">Search Website:</label>
  <input type="search" id="mysearch">
</form>
```

Note: The search input (i.e. type="search") is supported by all major web browsers such as Firefox, Chrome, Safari, Opera, Internet Explorer 10 and above.

Input Type Tel

The `tel` input type can be used for entering a telephone number.

Browsers don't support tel input validation natively. However, you can use the `placeholder` attribute to help users in entering the correct format for a phone number, or specify a [regular expression](#) to validate the user input using the `pattern` attribute. Let's check out an example:

Example

Try this code »

```
<form>
```

```
<label for="myphone">Telephone Number:</label>
<input type="tel" id="myphone" placeholder="xx-xxxx-xxxx"
required>
</form>
```

Note: The validation for tel input (i.e. type="tel") is currently not supported by any browser because format for phone numbers vary so much across countries, but it is still useful. Mobile browsers display a numeric keyboard for tel input field for entering phone numbers.

Input Type Time

The `time` input type can be used for entering a time (hours and minutes).

Browser may use 12- or 24-hour format for inputting times, based on local system's time setting.

Example

Try this code »

```
<form>
  <label for="mytime">Select Time:</label>
  <input type="time" id="mytime">
</form>
```

Warning: The input type="time" is not supported by Internet Explorer and Safari browsers. Currently supported by Chrome, Firefox, Edge, and Opera browsers.

Input Type URL

The `url` input type can be used for entering URL's or web addresses.

You can use the `multiple` attribute to enter more than one URL. Also, if `required` attribute is specified browser will automatically carry out validation to ensure that only text that matches the standard format for URLs is entered into the input box. Let's see how this works:

Example

Try this code »

```
<form>
  <label for="myurl">Enter Website URL:</label>
  <input type="url" id="myurl" required>
</form>
```

Note: The validation for the url input (i.e. type="url") is supported by all major browsers like Firefox, Chrome, Safari, Opera, Internet Explorer 10 and above.

Input Type Week

The `week` input type allows the user to select a week and year from a drop-down calendar.

Let's try out the following example to understand how this works:

Example

Try this code »

```
<form>
  <label for="myweek">Select Week:</label>
  <input type="week" id="myweek">
</form>
```

Warning: The input type="week" is not supported by Firefox, Safari and Internet Explorer browsers. Currently supported by Chrome, Edge, and Opera browsers.

HTML <!DOCTYPE> tag

On the HTML document you have often seen that there is a `<!DOCTYPE html>` declaration before the `<html>` tag. HTML `<!DOCTYPE>` tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declaration (DTD).

Technically `<!DOCTYPE >` is not a tag/element, it just an instruction to the browser about the document type. It is a null element which does not contain the closing tag, and must not include any content within it.

Actually, there are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc.

The `<!DOCTYPE>` declaration refers Document Type Declaration (DTD) in HTML 4.01; because HTML 4.01 was based on SGML. But HTML 5 is not SGML based language.

DTD defines the rules for the markup languages so that the browsers recognize the content correctly.

The doctype declaration differs between HTML versions. The HTML 5 doctype declaration is given below.

Syntax

```
<!DOCTYPE html>
```

Following are some specifications about the HTML `<!DOCTYPE>`

Display	None
Start tag/End tag	Start tag only
Usage	Structural

Let's see an example of HTML document with doctype declaration.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>`This is the title`</title>`
5. `</head>`
6. `<body>`
7. This is the content of the document.
8. `</body>`
9. `</html>`

Note: It is always a good practice to add a declaration to your HTML documents to enable web browser to recognize that what type of document to expect.

Building blocks of HTML

An HTML document consist of its basic building blocks which are:

- **Tags:** An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.
- **Attribute:** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

Syntax

1. `<tag name attribute_name= " attr_value"> content </ tag name>`
 - **Elements:** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.
 - `<!DOCTYPE html>`
 - `<html>`
 - `<head>`
 - `<title>`The basic building blocks of HTML`</title>`
 - `</head>`
 - `<body>`
 - `<h2>`The building blocks`</h2>`
 - `<p>`This is a paragraph tag`</p>`
 - `<p style="color: red">`The style is attribute of paragraph tag`</p>`
 - ``The element contains tag, attribute and content``
 - `</body>`
 - `</html>`

HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

- All HTML tags must be enclosed within < > these brackets.
- Every tag in HTML performs different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

Syntax

<tag> content </tag>

HTML Tag Examples

Note: HTML Tags are always written in lowercase letters. The basic HTML tags are given below:

<p> Paragraph Tag </p>

<h2> Heading Tag </h2>

 Bold Tag

<i> *Italic Tag* </i>

<u> Underline Tag </u>

Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

**
 Tag:** br stands for break line, it breaks the line of the code.

<hr> Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

HTML Meta Tags

DOCTYPE, title, link, meta and style

HTML Text Tags

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, , <dfn>, <kbd>, <pre>, <samp>, <var> and

HTML Link Tags

<a> and <base>

HTML Image and Object Tags

, <area>, <map>, <param> and <object>

HTML List Tags

, , , <dl>, <dt> and <dd>

HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

HTML Form Tags

form, input, textarea, select, option, optgroup, button, label, fieldset and legend


HTML Scripting Tags

script and noscript

Note: We will see examples using these tags in later chapters.

HTML Tags List

Following is the complete list of HTML tags with the description which are arranged alphabetically.







Note: Here  represents newly added Elements in HTML5.


Global Attributes

HTML global attributes are those attributes which are common for all HTML elements. The global attributes are supported by both standard and non-standard element.

The global attributes can be used with all elements, although it may not have any effect on some elements.

Following is the complete list of global attributes with their description:

Attributes	value	Description
accesskey	character	It is used to generate keyboard shortcuts for the current element.
class	classname	It is used to provide the class name for the current element. It is mainly used with the stylesheet.
Contenteditable 	true false	It determines whether the content within an element is editable or not.
contextmenu	menu_id	It defines the id for the <menu> element which is used as a context menu (a menu appear on right click) for an element.
data-* 	somevalue	It is used to store element-specific private data which can be accessed by Javascript.
dir	rtl ltr auto	It specifies the direction of the content inside the current element.
draggable 	true false auto	It specifies whether the content within an element is movable or not using Drag and Drop API.
dropzone 	copy move link	It specifies the action is taken on the dragged element when it is dropped, → such as whether it is copied, moved or linked.
hidden 		It is used to hide the element from view.
id	id	It specifies a unique id for the element. It can be used with CSS and JavaScript.
lang	language_code	It specifies the primary language for the content of an element.
style	style	It is used to apply inline CSS to the current element.
spellcheck 	true	It specifies whether the content should be checked for spelling

	false	errors or not.
tabindex	number	It determines the tabbing order of an element.
title	text	It is used to provide the title, name, or some extra information about the element.
translate 	yes no	It specifies whether the content of the element should be translated when the page is localized or not.

HTML Event Attributes

When a browser reacts on user action, then it is called as an event. For example, when you click on the submit button, then if the browser displays an information box.

In HTML5 there are lots of event attributes available which can be activated using a programming language such as JavaScript.

Following is a table of event attributes, using these attributes you can perform several events.

Windows Event Attributes

Windows events are related for the window object, and it can only be applied with `<body>` tag.

Attribute	Description
onafterprint	Executed the script after the document is printed.
onbeforeprint	Executed the script before the document is printed.
onbeforeunload	Executed the script before a document being unloaded.
onerror	Executed the script when an error occurs.

onhashchange	Executed the script when the anchor part in URL of the webpage is changed.
onload	Executed the script when the webpage is entirely loaded.
onmessage	Executed the script when a message event occurs.
onoffline	Executed the script when the network connection is disconnected, and browser started working offline.
ononline	Executed the script when the browser started working online
onpagehide	Executed the script when the current webpage is hidden such as if the user has moved away from the current webpage.
onpageshow	Executed the script when the current webpage is focused.
onpopstate	Executed the script when the window's active history is changed.
onresize	Executed the script when the window is resized.
onstorage	Executed the script when web storage is updated.
onunload	Executed the script when the current webpage is unloaded, or window is closed.

Form Event Attributes

Form event occurs when the user performs some action within the form such as submitting the form, selecting input field, etc.

The form events can be used with any element, but these are mainly used with HTML form elements.

Following is the list of all Form Event attributes:

Attribute	Description
-----------	-------------

onblur	Executed the script when form element loses the focus.
onchange	Executed the script when the value of the element is changed.
onfocus	Trigger an event when the element gets focused.
oninput	Executed the script when the user enters input to the element.
oninvalid	Executed the script when the element does not satisfy its predefined constraints.
onreset	Triggers the event when user reset the form element values.
onsearch	Triggers the event when a search field receives some input.
onselect	Triggers the event when the user has selected some text.
onsubmit	Triggers the event when a form is submitted.

Keyboard Event Attributes

Keyboard event occurs when a user interacts with the keyboard. Following is a list of the Keyboard event.

Attribute	Description
onkeydown	Triggers the event when the user presses down a key on the keyboard.
onkeypress	Trigger the event when the user presses the key which displays some character.
onkeyup	Trigger the event when the user releases the currently pressed key.

Mouse Event Attributes

Attribute	Description
-----------	-------------

onclick	Trigger the event when the mouse clicks on the element.
ondblclick	Trigger the event when mouse double-click occurs on the element.
onmousedown	Trigger the event when the mouse button is pressed on the element.
onmousemove	Trigger the event when the mouse pointer moves over the element.
onmouseout	Trigger the event when the mouse moves outside the element.
onmouseover	Trigger the event when the mouse moves onto the element.
onmouseup	Trigger the event when the mouse button is released.
onmousewheel	Deprecated. Use the onwheel attribute.
onwheel	Trigger the event when the mouse wheel rolls up or down on the element

Clipboard Event Attributes

Attribute	Description
oncopy	Trigger the event when the user copies the content to the system clipboard.
oncut	Trigger the event when the content of an element is cut and copy to the clipboard.
onpaste	Trigger the event when the user pastes some content in an element.

Media Event Attributes

Attribute	Description
onabort	Executed the script when media playback is aborted.

oncanplay	Executed the script when the media file is ready to play.
oncanplaythrough	Executed the script when the media file is ready to play without buffering or stopping.
oncuechange	Executed the script text cue of <track> element is changed.
ondurationchange	Executed the script when the media file duration is changed.
onemptied	Executed the script if media occurs some fatal error, and the file becomes unavailable.
onended	Executed the script when the media file occurs its end point.
onerror	Executed the script when some error occurred while fetching the media data.
onloadeddata	Executed the script when media data is loaded.
onloadedmetadata	Executed the script when metadata of media file is loaded.
onloadstart	Executed the script when loading of media file starts.
onpause	Executed the script when media playback is paused.
onplay	Executed the script when media file ready to play after being paused.
onplaying	Executed the script when media file is started playing.
onprogress	Executed the script when the browser is in the process of getting the media data.
onratechange	Executed the script when playback speed changed.

e	
onseeked	Executed the script when seek operation is ended and seeking attribute is set to false.
onseeking	Executed the script when seek operation is active and seeking attribute is set to true.
onstalled	Executed the script when browser unexpectedly stopped fetching the data media.
onsuspend	Executed the script if fetching of media data is intentionally stopped.
ontimeupdate	Executed the script when playback position is changed, such as if a user fasts forward the track.
onvolumechange	Executed the script when media volume is changed (muted or unmuted).
onwaiting	Executed the script if playback pause to wait for loading more data.

HTML 5 Tags

There is a list of newly included tags in HTML 5. These HTML 5 tags (elements) provide a better document structure. This list shows all HTML 5 tags in alphabetical order with description.

List of HTML 5 Tags

Tag	Description
<article>	This element is used to define an independent piece of content in a document, like a magazine or a newspaper article.
<aside>	It specifies that article is slightly related to the rest of the whole page.

<audio>	It is used to play audio file in HTML.
<bdi>	The bdi stands for bi-directional isolation. It isolates a part of text that is formatted from the outside text document.
<canvas>	It is used to draw canvas.
<data>	It provides machine readable version of its data.
<datalist>	It provides auto complete feature for textfield.
<details>	It specifies the additional information or controls required by user.
<dialog>	It defines a window or a dialog box.
<figcaption>	It is used to define a caption for a <figure> element.
<figure>	It defines a self-contained content like photos, diagrams etc.
<footer>	It defines a footer for a section.
<header>	It defines a header for a section.
<main>	It defines the main content of a document.
<mark>	It specifies the marked or highlighted content.
<menuitem>	It defines a command that the user can invoke from a popup menu.
<meter>	It is used to measure the scalar value within a given range.
<nav>	It is used to define the navigation link in the document.

<progress>	It specifies the progress of the task.
<rp>	It defines what to show in browser that don't support ruby annotation.
<rt>	It defines an explanation/pronunciation of characters.
<ruby>	It defines ruby annotation along with <rp> and <rt>.
<section>	It defines a section in the document.
<summary>	It specifies a visible heading for <detailed> element.
<svg>	It is used to display shapes.
<time>	It is used to define a date/time.
<video>	It is used to play video file in HTML.
<wbr>	It defines a possible line break.

HTML Anchor

The **HTML anchor tag** defines a *hyperlink that links one page to another page*. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL.

href attribute of HTML anchor tag

The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

```
<a href = "....."> Link Text </a>
```

Let's see an example of HTML anchor tag.

1. `Click for Second Page`

Specify a location for Link using target attribute

If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

Example:

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title></title>
5. </head>
6. <body>
7. <p>Click on this-link to go on home page of JavaTpoint.</p>
8. </body>
9. </html>

Note:

- The **target** attribute can only use with href attribute in anchor tag.
- If we will not use target attribute then link will open in same page.

Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

HTML Image

HTML img tag is used to display image on the web page. HTML img tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.

Let's see an example of HTML image.

1. <h2>HTML Image Example</h2>
2.

Attributes of HTML img tag

The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given below.

1) *src*

It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server.

The location of image may be on the same directory or another server.

2) *alt*

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words. The alt attribute is considered good for SEO prospective.

3) *width*

It is an optional attribute which is used to specify the width to display the image. It is not recommended now. You should apply CSS in place of width attribute.

4) *height*

It h3 the height of the image. The HTML height attribute also supports iframe, image and object elements. It is not recommended now. You should apply CSS in place of height attribute.

Use of height and width attribute with img tag

You have learnt about how to insert an image in your web page, now if we want to give some height and width to display image according to our requirement, then we can set it with height and width attributes of image.

Example:

1. ``

Use of alt attribute

We can use alt attribute with `` tag. It will display an alternative text in case if image cannot be displayed on browser. Following is the example for alt attribute:

1. ``

How to get image from another directory/folder?

To insert an image in your web, that image must be present in your same folder where you have put the HTML file. But if in some case image is available in some other directory then you can access the image like this:

1. ``

In above statement we have put image in local disk E----->images folder----->animal.png.

Note: If src URL will be incorrect or misspell then it will not display your image on web page, so try to put correct URL.

Use tag as a link

We can also link an image with other page or we can use an image as a link. To do this, put tag inside the <a> tag.

Example:

1. ``

HTML Table

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

We can create a table to display data in tabular form, using <table> element, with the help of <tr> , <td>, and <th> elements.

In Each table, table row is defined by <tr> tag, table header is defined by <th>, and table data is defined by <td> tags.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

HTML Table Tags

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

HTML Table Example

Let's see the example of HTML table tag. Its output is shown above.

1. **<table>**
2. **<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>**
3. **<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>**
4. **<tr><td>James</td><td>William</td><td>80</td></tr>**
5. **<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>**
6. **<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>**
7. **</table>**

HTML Table with Border

There are two ways to specify border for HTML tables.

1. By border attribute of table in HTML
2. By border property in CSS

1) HTML Border attribute

You can use border attribute of table tag in HTML to specify border. But it is not recommended now.

1. `<table border="1">`
2. `<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>`
3. `<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>`
4. `<tr><td>James</td><td>William</td><td>80</td></tr>`
5. `<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>`
6. `<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>`
7. `</table>`

2) CSS Border property

It is now recommended to use border property of CSS to specify border in table.

1. `<style>`
2. `table, th, td {`
3. `border: 1px solid black;`
4. `}`
5. `</style>`

You can collapse all the borders in one border by border-collapse property. It will collapse the border into one.

1. `<style>`
2. `table, th, td {`
3. `border: 2px solid black;`
4. `border-collapse: collapse;`
5. `}`
6. `</style>`

HTML Table with cell padding

You can specify padding for table header and table data by two ways:

1. By cellpadding attribute of table in HTML
2. By padding property in CSS

The cellpadding attribute of HTML table tag is obsolete now. It is recommended to use CSS. So let's see the code of CSS.

1. `<style>`
2. `table, th, td {`
3. `border: 1px solid pink;`
4. `border-collapse: collapse;`
5. `}`
6. `th, td {`
7. `padding: 10px;`
8. `}`
9. `</style>`

HTML Table width:

We can specify the HTML table width using the **CSS width** property. It can be specify in pixels or percentage.

We can adjust our table width as per our requirement. Following is the example to display table with width.

1. `table{`
2. `width: 100%;`
3. `}`

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>table</title>`
5. `<style>`
6. `table{`
7. `border-collapse: collapse;`
8. `width: 100%;`
9. `}`
10. `th,td{`
11. `border: 2px solid green;`


```
12.     padding: 15px;
13. }
14.
15. </style>
16. </head>
17. <body>
18. <table>
19. <tr>
20.     <th>1 header</th>
21.     <th>1 header</th>
22.     <th>1 header</th>
23. </tr>
24. <tr>
25.     <td>1data</td>
26.     <td>1data</td>
27.     <td>1data</td>
28. </tr>
29. <tr>
30.     <td>2 data</td>
31.     <td>2 data</td>
32.     <td>2 data</td>
33. </tr>
34. <tr>
35.     <td>3 data</td>
36.     <td>3 data</td>
37.     <td>3 data</td>
38. </tr>
39. </table>
40. </body>
41. </html>
```

HTML Table with colspan

If you want to make a cell span more than one column, you can use the colspan attribute.

It will divide one cell/row into multiple columns, and the number of columns depend on the value of colspan attribute.

Let's see the example that span two columns.

CSS code:

```
1. <style>
```

2. table, th, td {
3. border: 1px solid black;
4. border-collapse: collapse;
5. }
6. th, td {
7. padding: 5px;
8. }
9. `</style>`

HTML code:

1. `<table style="width:100%">`
2. `<tr>`
3. `<th>Name</th>`
4. `<th colspan="2">Mobile No.</th>`
5. `</tr>`
6. `<tr>`
7. `<td>Ajeet Maurya</td>`
8. `<td>7503520801</td>`
9. `<td>9555879135</td>`
10. `</tr>`
11. `</table>`

HTML Table with rowspan

If you want to make a cell span more than one row, you can use the rowspan attribute.

It will divide a cell into multiple rows. The number of divided rows will depend on rowspan values.

Let's see the example that span two rows.

CSS code:

1. `<style>`
2. table, th, td {
3. border: 1px solid black;
4. border-collapse: collapse;
5. }
6. th, td {
7. padding: 10px;
8. }
9. `</style>`

HTML code:

1. `<table>`
2. `<tr><th>Name</th><td>Ajeet Maurya</td></tr>`
3. `<tr><th rowspan="2">Mobile No.</th><td>7503520801</td></tr>`
4. `<tr><td>9555879135</td></tr>`
5. `</table>`

HTML table with caption

HTML caption is displayed above the table. It must be used after table tag only.

1. `<table>`
2. `<caption>Student Records</caption>`
3. `<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>`
4. `<tr><td>Vimal</td><td>Jaiswal</td><td>70</td></tr>`
5. `<tr><td>Mike</td><td>Warn</td><td>60</td></tr>`
6. `<tr><td>Shane</td><td>Warn</td><td>42</td></tr>`
7. `<tr><td>Jai</td><td>Malhotra</td><td>62</td></tr>`
8. `</table>`

Styling HTML table even and odd cells

CSS code:

1. `<style>`
2. `table, th, td {`
3. `border: 1px solid black;`
4. `border-collapse: collapse;`
5. `}`
6. `th, td {`
7. `padding: 10px;`
8. `}`
9. `table#alter tr:nth-child(even) {`
10. `background-color: #eee;`
11. `}`
12. `table#alter tr:nth-child(odd) {`
13. `background-color: #fff;`
14. `}`
15. `table#alter th {`
16. `color: white;`
17. `background-color: gray;`

18. }
19. `</style>`

HTML `<tbody>` tag

HTML `<tbody>` tag is used to group the table rows (`<tr>`) together, which indicates that this is body part of a table (`<table>`).

The `<tbody>` tag must be a child of `<table>` element.

The `<tbody>` is used along with `<thead>` and `<tfoot>` which shows the different part of the table that are table head, table body, and table footer, however, it does not affect the layout of the table.

These elements can be used for providing semantic information which can be helpful in accessibility purpose, or rendering the header at top and footer at the bottom while printing a large table.

Tips: The `<tbody>` tag must contain one or more `<tr>` elements.

Syntax

1. `<tbody>.....</tbody>`

Following are some specifications about the HTML `<tbody>` tag

Display	Inline
Start tag/End tag	Both Start and End tag
Usage	Table body

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>HTML tbody tag</title>`
5. `<style>`
6. `body{`
7. `margin-left: 195px;"`
8. `}`

```
9.     </style>
10. </head>
11. <body>
12. <h2>Example of the tbody tag</h2>
13. <table border="1" bgcolor="#98f5ff">
14.   <thead>
15.     <tr>
16.       <th>EmpId</th>
17.       <th>Name</th>
18.       <th>Email-Id</th>
19.     </tr>
20.   </thead>
21.   <tbody>
22.     <tr>
23.       <td>121</td>
24.       <td>John</td>
25.       <td>john123@gmail.com</td>
26.     </tr>
27.
28.     <tr>
29.       <td>122</td>
30.       <td>William </td>
31.       <td>william56@gmail.com</td>
32.     </tr>
33.
34.     <tr>
35.       <td>123</td>
36.       <td>Amit</td>
37.       <td>amitk98@gmail.com</td>
38.     </tr>
39.   </tbody>
40. </table>
41. </body>
42. </html>
```

Attribute:

Tag-specific attributes:

Attribute	Value	Description
align	right left center justify char	It determines the alignment of the content inside the <tbody> element. (Not Supported in HTML5)
char	character	It specifies the alignment of the <tbody> content to the character. (Not Supported in HTML5)
charoff	Number	It specifies the number of characters the content will be aligned from the character specified by the char attribute. (Not Supported in HTML5)
valign	top middle bottom baseline	It determines the vertical alignment of the content inside the <tbody> element. (Not Supported in HTML5)

Global attribute:

The <tbody> tag supports the Global attributes in HTML.

Event attribute:

The <tbody> tag supports the Event attributes in HTML.

HTML <td> tag

HTML <td> tag is used to specify the cells of an HTML table which contain data of the table. The <td> tag must be the child element of <tr> (table row) tag. Each table row can contain multiple <td> data elements.

The grouped <td> elements of a <tr> tag renders as a single row in the table. The content of the <td> elements is regular and left-aligned in the table by default.

Syntax

1. `<td>.....</td>`

Display	Inline
Start tag/End tag	Start and End tag
Usage	Table content

Example

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>HTML td tag</title>
5.   <style>
6.     th{
7.       background-color: #6495ed;
8.     }
9.     th,td{
10.      border: 1px solid black;
11.      padding: 10px;
12.    }
13.  </style>
14.</head>
15.<body>
16. <h2>Example of td Tag</h2>
17. <table style=" border-collapse: collapse; background-color:#dcdcdc;">
18.   <tr>
19.     <th>Product</th>
20.     <th>Quantity</th>
21.     <th>Price</th>
22.   </tr>
23.
24.   <tr>
25.     <td>Books</td>
26.     <td>5</td>
27.     <td>589</td>
28.   </tr>
29.
30.   <tr>
31.     <td>T-shirt</td>
32.     <td>5</td>
33.     <td>3500</td>
```

```

34. </tr>
35.
36.     <tr>
37.     <td>Jeans</td>
38.     <td>2</td>
39.     <td>5000</td>
40. </tr>
41. </table>
42. </body>
43. </html>

```

Attribute:

Tag-specific attributes:

Attribute	Value	Description
abbr	text	It defines the abbreviated version of content of the cell. (Not Supported in HTML5)
align	left right center justify char	It specifies the alignment of the content of the cell. (Not Supported in HTML5)
axis	category_name	It Categorizes Cells. . (Not Supported in HTML5)
bbgcolor	rgb(x,x,x) #xxxxxx Color_name	It sets the background color of the cell. (Not Supported in HTML5)
char	character	It specifies the alignment of the content of cell to the character. (Not Supported in HTML5)

charoff	number	It determines the number of characters the content aligned from the character specified by the char attribute. (Not Supported in HTML5)
colspan	number	It determines the number of columns a cell should span.
headers	header_id	It determines one or more header cells to which a cell is related.
height	% pixels	It determines the height of a table cell. (Not Supported in HTML5)
nowrap	nowrap	If it sets then content inside the cell should not wrap. (Not Supported in HTML5)
rowspan	number	It determines the number of rows a cell should span.
scope	col colgroup row rowgroup	It specifies the cells that the header element relates to. (Not Supported in HTML5)
valign	top middle bottom baseline	It determines the vertical alignment of the cell content. (Not Supported in HTML5)
width	% pixels	It determines the width of the cell. (Not Supported in HTML5)

Global attribute:

The <td> tag supports the Global attributes in HTML.

Event attribute:

The <td> tag supports the Event attributes in HTML.

HTML <template> tag

HTML <template> tag is used to hold the client-side content that will not render at the time of page load, but it can be instantiated during runtime using JavaScript.

The content of the template will not be displayed until it is not activated using JavaScript. The browser processes the content of the <template> element while loading the page to ensure that the content is valid, the contents are not rendered, however.

It can also be useful when you want to use same content multiple times in your HTML document without any change.

The <template> tag can be placed anywhere inside of <head>, <body>, <frameset>, or <table> elements.

The <template> tag is newly added element in HTML5.

Syntax

1. **<template>.....</template>**

Following are some specifications about the HTML <template> tag

Display	None
Start tag/End tag	Start and End tag
Usage	Formatting

Example

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>HTML Template tag</title>
5. <style>
6. body{
7. background-color: #e6e6fa;
8. }
9. </style>
10. </head>
11. <body>
12. <h2>Example of template tag</h2>

```
13. <button onclick="clickMe()">Click Me</button><br>
14.
15. <template id="mytemplate">
16.   
17.   <script>
18.     alert("Thank you for choosing template. Click OK for image.")
19.   </script>
20. </template>
21.
22. <script>
23.   function clickMe() {
24.     var x= document.getElementsByTagName("template")[0];
25.     var clon = x.content.cloneNode(true);
26.     document.body.appendChild(clon);}
27. </script>
28. </body>
29. </html>
```

Attribute:

Tag-specific attributes:

The <template> tag does not contain any specific attribute.

Global attribute:

The <template> tag supports the Global attributes in HTML.

HTML Textarea

The **HTML <textarea> tag** is used to define a multi-line text input control.

It can hold unlimited number of characters and the texts are displayed in a fixed-width font (usually courier).

The size of the HTML textarea is defined by <cols> and <rows> attribute, or it can also be defined through CSS height and width properties.

HTML Textarea Example

1. <textarea rows="9" cols="70">
2. JavaTpoint textarea tag example with rows and columns.

3. `</textarea>`

New HTML 5 Textarea Attributes

Attribute	Description
autofocus	It specifies that a text area should be automatically get focused when the page is loaded.
form	It specifies one or more forms the textarea belongs to.
maxlength	It specifies the maximum number of characters allowed in the text area.
placeholder	It specifies a short hint that describes the expected value of a textarea.
required	It specifies that textarea must be filled out.
wrap	It specifies that how the texts in the textarea are wrapped at the time of the submission of the form.

HTML Textarea form attribute

The form attribute specifies one or more forms the text area belongs to.

1. `<form action="updates.jsp" id="usrform">`
2. Name: `<input type="text" name="username">`
3. `<input type="submit">`
4. `</form>`
5. `
`
6. `<textarea rows="9" cols="70" name="comment" form="usrform">`
7. Enter text here...`</textarea>`
8. `<p>`The text area above is outside the form element, but should still be a part of the form.`</p>`
9. `<p>`Note:`` The form attribute is not supported in Internet Explorer.`</p>`

The textarea element above is outside the form , but it is still the part of the form.

Note: The form attribute is not supported in Internet Explorer.

HTML <tfoot> tag

HTML <tfoot> tag is used to define the set of rows which represents footer of an HTML table. The <tfoot> tag must contain one or more <tr> element.

The <tfoot> tag is used as a child element of HTML table (<table>) along with <thead> and <tbody> elements, where <thead> defines table header and <tbody> defines the table body.

Tips: The <thead>, <tbody>, and <tfoot> elements do not affect the table layout, and if you want to apply the change in table layout then use CSS properties.

Syntax

1. `<tfoot>`
2. `<tr></tr>`
3. `<tr></tr>`
4. `lt;/tfoot>`

Following are some specifications about the HTML <tfoot> tag

Display	None
---------	------

Start tag/End tag	Start and End tag
Usage	HTML Tables

Example

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>HTML tfoot Tag</title>
5.     <style>
6.       table{
7.         border-collapse: collapse;
8.       }
9.       thead,tfoot{
10.        background-color: #3f87a6;
11.      }
12.      tbody{
13.        background-color: #97ffff;
14.      }
15.    </style>
16.  </head>
17. <body>
18. <h1>Example of tfoot tag</h1>
19. <table border="1" >
20.   <thead>
21.     <tr>
22.       <th>Items</th>
23.       <th>Quantity</th>
24.       <th>Expenditure</th>
25.     </tr>
26.   </thead>
27.   <tfoot>
28.     <tr>
29.       <th>Total</th>
30.       <th>90</th>
31.       <th>4175</th>
32.     </tr>
33.   </tfoot>
34.   <tbody>
35.     <tr>
36.       <td>Books</td>
37.       <td>5</td>

```

```

38.         <td>1500</td>
39.     </tr>
40.     <tr>
41.         <td>Drawing-Paper</td>
42.         <td>50</td>
43.         <td>800</td>
44.     </tr>
45.     <tr>
46.         <td>Marker</td>
47.         <td>35</td>
48.         <td>1875</td>
49.     </tr>
50. </tbody>
51. </table>
52. </body>
53. </html>

```

Attribute:

Tag-specific attributes:

Attribute	Value	Description
align	right left center justify char	It determines the alignment of the content inside the <tfoot> element. (Not Supported in HTML5)
char	Character	It specifies the alignment of the content inside the <tfoot> element to the character. (Not Supported in HTML5)
charoff	Number	It specifies the number of characters the content will be aligned from the character specified by the char attribute. (Not Supported in HTML5)
valign	top middle bottom	It determines the vertical alignment of the content inside the <tfoot> element. (Not Supported in HTML5)

	baseline	
--	----------	--

Global attribute:

The <tfoot> tag supports the Global attributes in HTML.

Event attribute:

The <tfoot> tag supports the Event attributes in HTML.

HTML <th> tag

In an HTML table there are two types of cells:

- **Header cell** - It contains the header information (created using <th> element)
- **Data Cells** - It contains the main data of the table (created using <td> element).

HTML <th> tag is used to define the header cells of an HTML table. The header cell renders as bold and centered by default on the browser, but you can change its default style using CSS properties.

The <th> tag must be used as a child element of the <tr> element within <table> element. The size of the table is auto-adjustable as per the content size.

Syntax

1. <th>Content..... </th>

Following are some specifications about the HTML <th> tag

Display	None
Start tag/End tag	Both Start and End tag
Usage	HTML Tables

Example

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>HTML th Tag</title>


```

5.  <style>
6.  table{
7.      border-collapse: collapse;
8.      width: 70%;}
9.  th,td{
10.     background-color: #528b8b;
11.     padding: 10px;
12.  }
13. </style>
14. </head>
15. <body>
16. <h2>Example of th tag</h2>
17. <table border="1">
18.     <tr>
19.         <th>Month</th>
20.         <th>Date</th>
21.     </tr>
22.     <tr>
23.         <td>January</td>
24.         <td>20.01.2018</td>
25.     </tr>
26.     <tr>
27.         <td>February</td>
28.         <td>01.02.2018</td>
29.     </tr>
30.     <tr>
31.         <td>March</td>
32.         <td>15.03.2018</td>
33.     </tr>
34. </table>
35. </body>
36. </html>

```

Attribute:

Tag-specific attributes:

Attribute	Value	Description
-----------	-------	-------------

abbr	text	It defines the abbreviated version of content of the header cell. (Not Supported in HTML5)
align	left right center justify char	It specifies the alignment of the content of the header cell. (Not Supported in HTML5)
axis	category_name	It Categorizes header Cells (Not Supported in HTML5)
bgcolor	rgb(x,x,x) #xxxxxx Color_name	It sets the background color of the header cell. (Not Supported in HTML5)
char	character	It specifies the alignment of the content of the header cell to the character. (Not Supported in HTML5)
charoff	number	It specifies the number of characters the header cell content will be aligned from the character specified by the char attribute. (Not Supported in HTML5)
colspan	number	It determines the number of columns a header cell should span.sssssss
headers	header_id	It determines a space-separated list of header cells which contains information of the cells is related.
height	% pixels	It determines the height of a table header cell. (Not Supported in HTML5)
nowrap	nowrap	If it sets then content inside the header cell should not wrap. (Not Supported in HTML5)

rowspan	number	It determines the number of rows a cell should span.
scope	col colgroup row rowgroup	It specifies the cells that the header element relates to. (Not Supported in HTML5)
valign	top middle bottom baseline	It determines the vertical alignment of the cell content. (Not Supported in HTML5)
width	% pixels	It determines the width of the header cell. (Not Supported in HTML5)

Global attribute:

The <th> tag supports the Global attributes in HTML.

Event attribute:

The <th> tag supports the Event attributes in HTML.

HTML <thead> tag

HTML <thead> elements is used to define header of an HTML table. The <thead> tag is used along with <tbody> and <tfoot> tags which defines table header, table body, and table footer in an HTML table.

The <thead> tag must be child of <table> element, and it must be used before any <tbody>, <tr>, or <tfoot> elements.

The <thead> tag should contain at least one row <tr> element inside it.

Syntax

1. `<thead>`
2. `<tr>`
3. `</tr>`
4. `lt;/thead>`

Following are some specifications about the HTML `<thead>` tag

Display	Inline
Start tag/End tag	Start and End tag
Usage	Table header

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>HTML thead Tag</title>`
5. `<style>`
6. `table{`
7. `border-collapse: collapse;`
8. `margin: 15px;}`
9. `thead{`
10. `background-color: #005cb9;}`
11. `tbody{`
12. `background-color: #d4caca;}`
13. `th,td{`
14. `padding: 12px;}`
15. `</style>`
16. `</head>`
17. `<body>`
18. `<h2>Example of thead tag</h2>`
19. `<table border="1">`
20. `<caption>Population of India</caption>`
21. `<thead>`
22. `<tr>`
23. `<th>Year</th>`
24. `<th>Population</th>`
25. `</tr>`
26. `</thead>`
27. `<tbody>`
28. `<tr>`
29. `<td>2015</td>`

```

30.         <td>1,309,053,980</td>
31.     </tr>
32. <tr>
33.     <td>2016</td>
34.     <td>1,324,171,354</td>
35. </tr>
36. <tr>
37.     <td>2017</td>
38.     <td>1,339,180,127</td>
39. </tr>
40. <tr>
41.     <td>2018</td>
42.     <td>1,354,051,854</td>
43. </tr>
44. </tbody>
45. </table>
46. </body>
47. </html>

```

Attribute:

Tag-specific attributes:

Attribute	Value	Description
align	right left center justify char	It determines the alignment of the content inside the <thead> element t HTML5)
char	Character	It specifies the alignment of the content inside the <thead> element t Supported in HTML5)
charoff	Number	It specifies the number of characters the content that will be aligned specified by the char attribute. (Not Supported in HTML5)
valign	top middle bottom	It determines the vertical alignment of the content inside the <t Supported in HTML5)

Global attribute:

The <thead> tag supports the Global attributes in HTML.

Event attribute:

The <thead> tag supports the Event attributes in HTML.

HTML Time Tag

HTML <time> tag is used to define date and time. It displays time value in a 24 hour clock or a precise date in a Gregorian calendar in HTML.

It is used to encode dates and times in a machine-readable way to make easy to mark or schedule your task.

It also helps search engines to produce smarter search results.

HTML <time> is a new tag and introduced in HTML5.

Let's see the syntax to define date and time.

1. **<time>** Define Time and Date here **</time>**

Attribute

There is only one specific attribute of HTML5 time tag.

Attribute	Description
datetime	It is used to define machine-readable date/time within the time element.

HTML time tag example

1. **<p>**We open our shop at **<time>**09:00**</time>** am.**</p>**

2. `<p>The business meeting is scheduled on <time datetime="2009-02-18">next wednesday</time>.</p>`
3. `<p>The wedding of Salman's sister was scheduled at <time datetime="2014-11-19 T0 7:00-09:00">7pm last wednesday </time>.</p>`

Output:

We open our shop at 09:00 am.

The business meeting is scheduled on next wednesday.

The wedding of Salman's sister was scheduled at 7pm last wednesday .

In this example, First line in the body tag defines basic usage of time tag.

Second line shows how to use the datetime attribute to provide contents in a machine-readable format.

Third line uses the datetime attribute to provide an even more specific date and time.

The `<time>` tag also supports global attributes and event attributes in HTML 5.

HTML Phrase tag

The HTML phrase tags are special purpose tags, which defines the structural meaning of a block of text or semantics of text. Following is the list of phrase tags, some of which we have already discussed in HTML formatting.

- Abbreviation tag : `<abbr>`
- Acronym tag: `<acronym>` (not supported in HTML5)
- Marked tag: `<mark>`
- Strong tag: ``
- Emphasized tag : ``
- Definition tag: `<dfn>`
- Quoting tag: `<blockquote>`
- Short quote tag : `<q>`
- Code tag: `<code>`
- Keyboard tag: `<kbd>`

- Address tag: <address>
-

1. Text Abbreviation tag

This tag is used to abbreviate a text. To abbreviate a text, write text between <abbr> and </abbr> tag.

Example

1. **<p>**An **<abbr title = "Hypertext Markup language">**HTML **</abbr>**language is used to create web pages. **</p>**

2. Marked tag:

The content written between <mark> and </mark> tag will show as yellow mark on browser. This tag is used to highlight a particular text.

Example

1. **<p>**This tag will **<mark>**highlight**</mark>** the text.**</p>**

3. Strong text:

This tag is used to display the important text of the content. The text written between and will be displayed as important text.

Example

1. **<p>**In HTML it is recommended to use ****lower-case****, while writing a code. **</p>**

4. Emphasized text

This tag is used to emphasize the text, and displayed the text in italic form. The text written between and tag will italicized the text.

Example

1. **<p>**HTML is an ****easy ****to learn language.**</p>**

5. Definition tag:

When you use the <dfn> and </dfn> tags, it allow to specify the keyword of the content. Following is the example to show how to definition element.

Example

1. **<p>****<dfn>**HTML **</dfn>** is a markup language. **</p>**

6. Quoting text:

The HTML `<blockquote>` element shows that the enclosed content is quoted from another source. The Source URL can be given using the `cite` attribute, and text representation of source can display using `<cite> </cite>` element.

Example

1. `<blockquote cite="https://www.keepinspiring.me/famous-quotes/"><p>?The first step toward success is taken when you refuse to be a captive of the environment in which you first find yourself.?</p></blockquote>`
2. `<cite>-Mark Caine</cite>`

7. Short Quotations:

An HTML `<q> </q>` element defines a short quotation. If you will put any content between `<q> </q>`, then it will enclose the text in double quotes.

Example:

1. `<p>Steve Jobs said: <q>If You Are Working On Something That You Really Care About, You Don't Have To Be Pushed. The Vision Pulls You.</q>?</p>`

8. Code tags

The HTML `<code> </code>` element is used to display the part of computer code. It will display the content in monospaced font.

1. `<p>First Java program</p>`
2. `<p><code>class Simple{ public static void main(String args[]){`
3. `System.out.println("Hello Java"); } } </code>`
4. `</p>`

9. Keyboard Tag

In HTML the keyboard tag, `<kbd>`, indicates that a section of content is a user input from keyboard.

1. `<p>Please press <kbd>Ctrl</kbd> + <kbd>Shift</kbd> + t<kbd></kbd> to rest ore page on chrome.</p>`

10. Address tag

An HTML `<address>` tag defines the contact information about the author of the content. The content written between `<address>` and `</address>` tag, then it will be displayed in italic font.

1. **<address>** You can ask your queries by contact us on ****example123@newdomain.com****
2. **
** You can also visit at: **
**58 S. Garfield Street. Villa Rica, GA 30187.
3. **</address>**

HTML Responsive

Responsive Web design

Responsive web design is used to make your web page look appropriate, good, and well placed on all devices (desktop, tablet, smartphone etc.)

Responsive web design uses HTML and CSS to resize, hide, shrink, enlarge, or move the content. It makes the content look good on any screen.

Set the viewport

Let's see how to set the viewport.

how to set the viewport

Responsive Images

Images which can be scaled nicely to fit any browser size are known as responsive images.

How to make Image Responsive?

By using the width property

Set the CSS width property to 100% to make the image responsive and scale up and down.

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
4. `<body>`
5. `<h2>Responsive Image</h2>`
6. `<p>When we set the CSS width property to 100%, it makes the image responsive.`
7. `Resize the browser window to see the effect.</p>`
8. `(change image)`
9. `</body>`
10. `</html>`

Note: A problem with the above method (width: 100%) is that the image can be scaled up to be larger than its original size. So, it is better to use the max-width property instead.

By using the max-width Property

This method is best and most used because it facilitates that the image will scale down if it has to, but never scale up to be larger than its original size.

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
4. `<body>`
5. `<h2>Responsive Image</h2>`
6. `<p>"max-width:100%" makes the image responsive and also ensures that the image`
7. `doesn't get bigger than its original size.</p>`
8. `<p>Resize the browser window to see the effect.</p>`
9. ` (Change the image)`
10. `</body>`
11. `</html>`

Change images according to the browser width

By using the HTML `<picture>` element, you can set two or more images according to the browser width. It will change the picture when you change the browser-size. i.e. desktop and phone.

Example

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h2>Change Images Depending on Browser Width</h2>
6. <p>Resize the browser width and the image will change at 600px and 1500px.</p>
7. <picture>
8. <source srcset="img_smallflower.jpg" media="(max-width: 600px)">(Change image)
9. <source srcset="img_flowers.jpg" media="(max-width: 1500px)">(Change image)
10. <source srcset="flowers.jpg">
11.
12. </picture>
13. </body>
14. </html>

Responsive Text-size

We can make the text size responsive by using the "uv" unit. It means viewport-width. By using this, we can make the text size to follow the browser window screen.

Example

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h1 style="font-size:10vw;">Here size is 10vw.</h1>
6. <p style="font-size:6vw;">Here size is 6vw.</p>
7. <p style="font-size:4vw;">Here size is 4vw.</p>
8. <p>Resize the browser window to see how the text size changes.</p>
9. </body>
10. </html>

Note: viewport specifies the browser window size. 1vw = 1% of viewport width. Means, if the viewport is 100cm wide, 1vw is 1.0cm.

Media Query

We can also use media query to make responsive websites.

HTML Classes

Class Attribute in HTML

The HTML class attribute is used to specify a single or multiple class names for an HTML element. The class name can be used by CSS and JavaScript to do some tasks for HTML elements. You can use this class in CSS with a specific class, write a period (.) character, followed by the name of the class for selecting elements.

A class attribute can be defined within `<style>` tag or in separate file using the (.) character.

In an HTML document, we can use the same class attribute name with different elements.

Defining an HTML class

To create an HTML class, firstly define style for HTML class using `<style>` tag within `<head>` section as following example:

Example:

```
1. <head>
2.   <style>
3.     .headings{
4.       color: lightgreen;
5.       font-family: cursive;
6.       background-color: black; }
7.   </style>
8. </head>
```

We have define style for a class name "headings", and we can use this class name with any of HTML element in which we want to provide such styling. We just need to follow the following syntax to use it.

```
1. <tag class="ghf"> content </tag>
```

Example 1:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <style>
5.     .headings{
6.       color: lightgreen;
```

```
7.         font-family: cursive;
8.         background-color: black; }
9.     </style>
10. </head>
11. <body>
12. <h1 class="headings">This is first heading</h1>
13. <h2 class="headings">This is Second heading</h2>
14. <h3 class="headings">This is third heading</h3>
15. <h4 class="headings">This is fourth heading</h4>
16. </body>
17. </html>
```

HTML <!DOCTYPE> tag

On the HTML document you have often seen that there is a <!DOCTYPE html> declaration before the <html> tag. HTML <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declaration (DTD).

Technically <!DOCTYPE > is not a tag/element, it just an instruction to the browser about the document type. It is a null element which does not contain the closing tag, and must not include any content within it.

Actually, there are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc.

The <!DOCTYPE> declaration refers Document Type Declaration (DTD) in HTML 4.01; because HTML 4.01 was based on SGML. But HTML 5 is not SGML based language.

DTD defines the rules for the markup languages so that the browsers recognize the content correctly.

The doctype declaration differs between HTML versions. The HTML 5 doctype declaration is given below.

Syntax

```
<!DOCTYPE html>
```

Following are some specifications about the HTML **<!DOCTYPE>**

Display	None
Start tag/End tag	Start tag only
Usage	Structural

Let's see an example of HTML document with doctype declaration.

1. **<!DOCTYPE html>**
2. **<html>**
3. **<head>**
4. **<title>**This is the title**</title>**
5. **</head>**
6. **<body>**
7. This is the content of the document.
8. **</body>**
9. **</html>**

Note: It is always a good practice to add a declaration to your HTML documents to enable web browser to recognize that what type of document to expect.

HTML **<track>** tag

HTML **<track>** tag is used to define time-based text tracks for a media file. The **<track>** tag must use as child element of **<audio>** and **<video>** elements.

The **<track>** tag is used to add subtitle, caption, or any other form of text which displayed when a media file plays.

HTML **<track>** is new tag in HTML5.

Syntax

1. **<track src=" " kind=" " srclang=" " label=" ">**

Following are some specifications about the HTML **<track>** tag

Display	None
Start tag/End tag	Only start tag(End tag forbidden)
Usage	HTML media

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>HTML track Tag</title>`
5. `</head>`
6. `<body>`
7. `<h2>Example of track tag</h2>`
8. `<video controls="controls">`
9. `<source src="flower.mp4" type="video/mp4">`
10. `<track src="flower.vtt" kind="subtitles" srclang="en" label="English">`
11. Sorry!Your browser does not support the track
12. `</video>`
13. `</body>`
14. `</html>`

How to create WEBVTT file:

Following are some basic steps to create WEBVTT file for `<track>` tag:

1. Open text editor in your PC such as Notepad
2. Write WEBVTT as the first line in the editor
3. Leave a blank line
4. Specify the time duration in the proper format (you can also provide numbering and add CSS).
5. Enter and write your text which you want to add a subtitle or caption, and repeat step 3 to 5 until you finish it.
6. Save it using .vtt extension.

Now your WEBVTT file is ready to use.

Attribute:

Tag-specific attributes:

Attribute	Value	Description
default	default	It specifies that the track should be enabled unless the user's preferences indicate that another track is more important.
kind	captions chapters descriptions metadata subtitles	It specifies that which type of text track you want to add.
label	text	It specifies the title of the text track.
src	URL	It defines the URL of the track file.
srclang	language_code	It defines the language of the track text content, such as English, Germany, etc.

Global attribute:

The <track> tag supports the Global attributes in HTML.

Event attribute:

The <track> tag supports the Event attributes in HTML.

HTML <tt> tag (Not supported in HTML5)

HTML <tt> tag was used to define text in monospaced font or fixed-width fonts so that it would render as teletype, text-only screen, or line printer on the browser.

Note: Do not use HTML <tt> tag, as it is not supported in HTML5, instead of you can use following tags for better use:

- `<code>`: To represent the computer programming code
- `<pre>`: To preserve line break and indentation in plain text.
- `<kbd>`: To represent keyboard input.
- `<var>`: To represent variables in an equation
- `<samp>`: To represent the text as sample output.

Syntax

Following are some specifications about the HTML `<tt>` tag

HTML `<tt>` tag (Not supported in HTML5)

HTML `<tt>` tag was used to define text in monospaced font or fixed-width fonts so that it would render on the screen, or line printer on the browser.

Note: Do not use HTML `<tt>` tag, as it is not supported in HTML5, instead of you can use `<code>` tag.

- `<code>`: To represent the computer programming code
- `<pre>`: To preserve line break and indentation in plain text.
- `<kbd>`: To represent keyboard input.
- `<var>`: To represent variables in an equation
- `<samp>`: To represent the text as sample output.

Syntax

Following are some specifications about the HTML `<tt>` tag

Display	Inline
Start tag/End tag	Start and End tag
Usage	Formatting

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>HTML tt tag</title>`
5. `</head>`
5. `<body>`
7. `<h2>Example of tt tag</h2>`
3. `<p>This is paragraph with default font</p>`
9. `<p><tt>This is teletype paragraph</tt></p>`
10. `</body>`
11. `</html>`

HTML `<u>` tag

HTML `<u>` tag is used to define a span of inline text with a non-textual annotation. It rendered as an solid underlined text, but it can be changed using CSS properties. This tag was deprecated in HTML 4.0 and redefined in HTML5.

In HTML5, `<u>` tag is used to represent the text that is stylistically different with normal text.

Tips: The use of `<u>` tag should be ignored as it may generate confusion for a hyperlinked text.

Syntax

Following are some specifications about the HTML `<u>` tag

Display	Inline
Start tag/End tag	Start and End tag
Usage	Formatting

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>HTML u tag</title>`
5. `<style>`

6. u {
7. text-decoration: red wavy underline;}
8. </style>
9. </head>
10. <body>
11. <h2>Example of u tag</h2>
12. <p>This tag can be useful to identifying <u>spelling mistakes </u>in an document.</p>
13. </body>
14. </html>

Attribute:

Tag-specific attributes:

The <u> tag does not contain any specific attribute.

Global attribute:

The <u> tag supports the Global attributes in HTML.

Event attribute:

The <u> tag supports the Event attributes in HTML.

HTML <var> tag

HTML <var> tag is a phrase tag which is used to define the variable for a mathematical equation, or in the programming context.

The content within <var> tag renders in italic font in most of the browsers, but it can be overridden using appropriate CSS.

Following are some related elements of <var> tag, which can also be used for the same context:

- <code>: To determine the computer programming code.
- <kbd>: To determine the keyboard input.
- <samp>: To determine the sample output.

Syntax

1. <var>.....</var>

Following are some specifications about the HTML <var> tag

Display	Inline
Start tag/End tag	Start and End tag
Usage	Formatting

Example

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>HTML var tag</title>
5. </head>
6. <body>
7. <h2>Example of var tag</h2>
8. <p>Following is equation for distributive law</p>
9. <p><var>a</var>(<var>b</var>+<var>c</var>)=<var>ab</var>+<var>ac</var></p>
10. </body>
11. </html>

Attribute:

Tag-specific attributes:

The <var> tag does not contain any specific attribute in HTML.

Global attribute:

The <var> tag supports the Global attributes in HTML.

Event attribute:

The <var> tag supports the Event attributes in HTML.

HTML <address> tag

HTML <address> tag is used to specify the authorship information of the article or webpage. It can contain any type of information which is needed such as, URL, physical address, phone number, email, other links, etc.

The `<address>` tag is useful for various contexts such as business contact information in the header of the page, or author related contact information, etc.

The contact information written between `<address>` tags mostly renders in the italic form on the browser.

Note: To represent a random address use `<p>` tag instead of `<address>` tag, as it should contain the main contact information.

Syntax

1. `<address>Contact Author at:
`
2. `Example@gmail.com</address>`

If you want to specify the information of the author for an article, you must place the `<address>` tag inside the `<article>` element.

Following are the specifications of `<address>` tag:

Display	Block
Usage	Semantic
Tags	Start and end tag

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Address Tag</title>`
5. `</head>`
6. `<body>`
7. `<h2>Example of Address tag</h2>`
8. `<address>The article is written by:Harshita
Contact Author at:`
9. `Example@gmail.com
You can also visit our blog:`
10. `javaTpoint blog`
11. `</address>`
12. `</body>`
13. `</html>`

Attributes

The <address> tag does not contain any specific attribute for it.

Global Attribute:

The <address> tag supports the global attributes.

Event attribute:

The <address> tag supports all Events attributes.

HTML <applet> tag (Not supported in HTML5)

HTML <applet> tag was used to embed the Java applet in an HTML document. This element has been deprecated in HTML 4.0 and instead of it we can use <object> and newly added element <embed>.

The use of Java applet is also deprecated, and most browsers do not support the use of plugins.

Note: The <applet> tag is deprecated in HTML4.0 and not supported in HTML5. So you can use <object> tag or <embed> tag instead of <applet>.

Syntax

1. <applet code="URL" height="200" width="100">.....</applet>

Display	Block
Start tag/End tag	Both Start tag and End tag
Usage	Embed Applets

Example

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Applet Tag</title>
5. </head>
6. <body>

7. `<p>Example of Applet Tag</p>`
8. `<applet code="Shapes.class" align="right" height="200" width="300">`
9. `Sorry! you need Java to see this`
10. `</applet>`
11. `</body>`
12. `</html>`

Attributes

Specific Attributes

Attribute name	Value	Description
code	URL	It specifies the URL of Java applet class file.
width	pixels	It specifies the display width of the applet panel.
height	pixels	It specifies the display height of applet panel
align	<ul style="list-style-type: none">○ left○ right○ top○ middle○ bottom	It specifies the position of applet application relative to surrounding content.
alt	text	It is used to display alternative text in case browser does not support Java.
archive	URL	This specifies the archived or compressed version of an applet application.
object	name	It specifies the URL or reference to a serialized representation of an applet.

codebase	URL	It specifies the exact or relative URL of applets .class file specified in the code attribute.
hspace	pixels	It specifies the horizontal space around the applet.
vspace	pixels	It specifies the vertical space around the applet.
name	name	It specifies the name for the applet

Global Attributes

The <applet> tag supports all Global Attributes in HTML

Event Attributes

The <applet> tag supports all Event Attributes in HTML

HTML <area> tag

Description:

The <area> tag defines the clickable areas or active areas inside the image-map which are associated with the hyperlinks. If you click on those areas then it will perform some action such as open a new image, new URL, etc. This tag is always used with <map> element.

Inside an image map different areas can be hyperlinked to various locations using multiple <area> elements in a single <map> element.

The <area> element is defined with (required) attributes **shape** and **coords**. The shape attribute specifies the shape of the area such as **rectangle**, **circle**, **square**, and **polygon**. The coords attribute defines the coordinates of areas inside the image.

What is Image-map

An image-map is defined as a graphical image with active areas so that when user click on those area, it can link to different destinations. **To define an image-map, we require the following things:**

- An HTML element with **usemap** attribute which defines a valid map name.
- HTML <map> element with **name** attribute whose value must be same as **usemap**
- One or more <area> elements inside a <map> element which create clickable areas in an image-map.

Syntax

1. `<area shape="" coords="" href="">`

Following are some specifications about the HTML <area> tag

Display	Block
Start tag/End tag	Only start tag(End tag forbidden)
Usage	Image Map




Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>HTML Area tag</title>`
5. `<style>`
6. `body{`
7. `margin-left: 250px;}`
8. `</style>`
9. `</head>`
10. `<body>`
11. `<h2>Example of HTML Area tag</h2>`
12. ``
13. `<map name="web">`
14. `<area shape="rect" coords="66,117,131,168" href="https://www.google.com">`
15. `<area shape="rect" coords="199,36,277,85" href="https://www. google.com ">`
16. `<area shape="rect" coords="330,107,406,159" href="https://www. google.com">`
17. `<area shape="rect" coords="199,185,267,236" href="https://www. google.com">`
18. `</map>`
19. `</body>`

Attribute:

Tag-specific attributes:

Attribute	Value	Description
alt	text	An alternative text String to display on the browser if it does not display the image.
coords	x1,y1,x2,y2(rect)	Defines coordinates for the upper left and lower right of a rectangle.
	x,y, radius(circle)	Defines coordinates for the circle.
	x1,y1,x2,y2,x3,y3,..(polygon)	Defines the polygon vertices.
href	href	URL It determines the hyperlink destination for the active area.
target	_blank	Open link in a new window
	_parent	Open link in the parent frame
	_self	Open link in current window
	_top	Open link with full width in the same window
	frame_name	In the frame. (Not supported in HTML5)

shape	default	It defines the default area(rectangular).
	rect	It defines the rectangular area.
	circle	Defines the circular area.
	poly	Defines the polygonal.
download	filename	Defines that hyperlink, which is used for downloading the resource.
rel 	alternate author bookmark help license next nofollow noreferrer prefetch prev search tag	It defines the relationship between current and linked document.
hreflang 	language_code	It specifies the language of the linked resource.
type 	media_type	It specifies the MIME type of linked source. (Not supported in HTML5)

Global attribute:

The <area> tag supports the global attributes in HTML

Event attribute:

The <area> tag supports the event attributes in HTML.

HTML Article Tag

The **HTML <article> tag** defines *an independent self-contained content in a document, page, application or a site.*

The article tag content makes sense on its own. It is independent and complete from other content shown on the page. This tag is generally used on Forum post, Blog post, News story, comment etc.

HTML article tag example

1. **<article>**
2. **<h2>**Narendra Modi**</h2>**
3. **<i>**(Naam to suna hi hoga) **</i>**
4. **<p>**Narendra DamodarDas Modi is the 15th and current Prime Minister of India,
5. Modi, a leader of the Bharatiya Janata Party (BJP), previously served as the Chief Minister
6. of Gujarat state from 2001 to 2014. He is currently the Member of Parliament (MP) from Varanasi. **</p>**
7. **</article>**

Narendra Modi

(Naam to suna hi hoga)

Narendra DamodarDas Modi is the 15th and current Prime Minister of India, Modi, a leader of the Bharatiya Janata Party (BJP), previously served as the Chief Minister of Gujarat state from 2001 to 2014. He is currently the Member of Parliament (MP) from Varanasi.

HTML article tag also supports global and event attributes in HTML.

HTML Aside Tag

The HTML <aside> tag provides information about the main content. According to W3C definition, the <aside> element represents content that forms the main textual flow of a document.

HTML aside is a new tag introduced in HTML5.

HTML aside tag example

1. `<p>I don't want to live in Ghaziabad, I wish; I could buy a flat in New Delhi.</p>`
2. `<aside>`
3. `<h3>New Delhi</h3>`
4. `<p>New Delhi is the capital of India.</p>`
5. `</aside>`

[next →](#) [← prev](#)

HTML Aside Tag

The HTML `<aside>` tag provides information about the main content. According to W3C definition, the `<aside>` element represents content that forms the main textual flow of a document.

HTML aside is a new tag introduced in HTML5.

HTML aside tag example

1. `<p>I don't want to live in Ghaziabad, I wish; I could buy a flat in New Delhi.</p>`
2. `<aside>`
3. `<h3>New Delhi</h3>`
4. `<p>New Delhi is the capital of India.</p>`
5. `</aside>`

Output:

I don't want to live in Ghaziabad, I wish; I could buy a flat in New Delhi.

New Delhi

New Delhi is the capital of India.

HTML aside tag also supports global and event attributes in HTML.






HTML Audio Tag

HTML audio tag is used to define sounds such as music and other audio clips. Currently there are three supported file formats for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

Browser	mp3	wav	ogg
 Internet Explorer	yes	no	no
 Google Chrome	yes	yes	yes
 Mozilla Firefox	yes*	yes	yes
 Opera	no	yes	yes
 Apple Safari	yes	yes	no

HTML Audio Tag Example

Let's see the code to play mp3 file using HTML audio tag.

1. `<audio controls>`
2. `<source src="koyal.mp3" type="audio/mpeg">`
3. Your browser does not support the html audio tag.
4. `</audio>`

Let's see the example to play ogg file using HTML audio tag.

1. `<audio controls>`
2. `<source src="koyal.ogg" type="audio/ogg">`

3. Your browser does not support the html audio tag.
4. `</audio>`

Attributes of HTML Audio Tag

There is given a list of HTML audio tag.

Attribute	Description
controls	It defines the audio controls which is displayed with play/pause buttons.
autoplay	It specifies that the audio will start playing as soon as it is ready.
loop	It specifies that the audio file will start over again, every time when it is completed.
muted	It is used to mute the audio output.
preload	It specifies the author view to upload audio file when the page loads.
src	It specifies the source URL of the audio file.

HTML Audio Tag Attribute Example

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

1. `<audio controls autoplay loop>`
2. `<source src="koyal.mp3" type="audio/mpeg"></audio>`

MIME Types for HTML Audio format

The available MIME type HTML audio tag is given below.

Audio Format	MIME Type
--------------	-----------

mp3	audio/mpeg
ogg	audio/ogg
wav	audio/wav






HTML Video Tag

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

Let's see the table that defines which web browser supports video file format.

Browser	mp4	webM
 Internet Explorer	yes	no
 Google Chrome	yes	yes
 Mozilla Firefox	yes	yes
 Opera	no	yes
 Apple Safari	yes	no

Android also supports mp4 format.

HTML Video Tag Example

Let's see the code to play mp4 file using HTML video tag.

1. `<video controls>`
2. `<source src="movie.mp4" type="video/mp4">`
3. Your browser does not support the html video tag.
4. `</video>`

Let's see the example to play ogg file using HTML video tag.

1. `<video controls>`
2. `<source src="movie.ogg" type="video/ogg">`
3. Your browser does not support the html video tag.
4. `</video>`

Attributes of HTML Video Tag

Let's see the list of HTML 5 video tag attributes.

Attribute	Description
controls	It defines the video controls which is displayed with play/pause buttons.
height	It is used to set the height of the video player.
width	It is used to set the width of the video player.
poster	It specifies the image which is displayed on the screen when the video is not played
autoplay	It specifies that the video will start playing as soon as it is ready.
loop	It specifies that the video file will start over again, every time when it is completed.

muted	It is used to mute the video output.
preload	It specifies the author view to upload video file when the page loads.
src	It specifies the source URL of the video file.

HTML Video Tag Attribute Example

Let's see the example of video tag in HTML where are using height, width, autoplay, controls and loop attributes.

1. `<video width="320" height="240" controls autoplay loop>`
2. `<source src="movie.mp4" type="video/mp4">`
3. Your browser does not support the html video tag.
4. `</video>`

MIME Types for HTML Video format

The available MIME type HTML video tag is given below.

Video Format	MIME Type
mp4	video/mp4
ogg	video/ogg
webM	video/webM

HTML <menu> tag

HTML <menu> tag specifies a list or menu of commands that a user can perform or activate. It is used for creating context menu as well as lists menu.

A <menu> element can contain one or more or <menuitem> elements within it.

Note: The `<menu>` tag was deprecated in HTML 4.01 and again included in HTML 5.1 specification. But it will better to ignore it as it is experimental and not supported by many browsers.

Syntax

1. `<menu>` `</menu>`

Following are some specifications about the HTML `<menu>` tag

Display	Inline
Start tag/End tag	Both Start and End tag
Usage	Semantic and textual

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Menu Tag</title>`
5. `</head>`
6. `<body>`
7. `<h2>Example of Menu Tag</h2>`
8. `<menu>`
9. `Home`
10. `Registration`
11. `Contact-us`
12. `About-us`
13. `</menu>`
14. `</body>`
15. `</html>`

Attribute:

Tag-specific attributes:

- popup
- toolbar
- context

Attribute	Value	Description
label	text	It specifies the label for the menu.
type	It specifies the type of the menu in an HTML document.	

Global attribute:

The <menu> tag supports the global attributes in HTML

Event attribute:

The <menu> tag supports the event attributes in HTML.

HTML <picture> tag

HTML <picture> tag is used in responsive web designing where we need to load the different images based on their viewport, height, width, orientation, and pixel density.

The <picture> tag contains one or more <source> elements and one elements.

According to the viewport, the matching image will be loaded from different <source> tag, and if no source contains the matching image, then the default image present in tag will be displayed on the browser.

This tag is a new tag in HTML5.

Syntax

1. <picture>
2. <source srcset="" media="">
3.
4. </picture>

Following are some specifications about the HTML <picture> tag

Display	Inline
Start tag/End tag	Both Start and End tag
Usage	Image

Example

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Picture Tag</title>
5.   <style>
6.     body{
7.       text-align: center;
8.     }
9.     p{
10.      font-size: bold;
11.      font-size: 20px;
12.      color: green;
13.    }
14.  </style>
15.</head>
16.<body>
17.<h2>Example of picture tag</h2>
18.<p>Resize the page to see the different versions of the images at different viewports, and as per viewport image will be automatically changed.</p>
19.<picture>
20.  <source srcset="download1.jpg" media="(min-width: 750px)">
21.  <source srcset="pic2.jpg" media="(min-width: 450px)">
22.  <img srcset="rose.jpg" alt="default image" style="width: auto;">
23.</picture>
24.</body>
25.</html>

```

Attribute:

Tag-specific attributes:

Attribute	Value	Description
media	media_query	It defines and accept any media query which can be defined in CSS.
srcset	URL	It defines the URL of the image which can be used for different situations. (Required)
type	video/ogg video/mp4 video/webM audio/ogg audio/mpeg	It determines the MIME type
src	URL	It specifies the location of the image.

Global attribute:

HTML <picture> tag supports the global attributes in HTML.

HTML Progress Tag

HTML <progress> tag is used to display the progress of a task. It provides an easy way for web developers to create progress bar on the website. It is mostly used to show the progress of a file uploading on the web page.

The HTML progress tag is new in HTML5 so you must use new browsers.

Attributes of HTML Progress Tag

HTML <progress> tag supports the global and event attributes as well as 2 specific attributes.

Tag	Description
value	It defines that how much work the task has been completed.

max	It defines that how much work the task requires in total.
-----	---

The progress tag should be used to represent progress of a task only, not for just a gauge (disk pace usage). For such purpose, <meter> element is used.

HTML Progress Tag Example

Let's see HTML progress example without attribute.

1. `<progress></progress>`

Let's see the progress example with value and max attributes.

1. Downloading progress:
2. `<progress value="43" max="100"></progress>`

Styling Progress Bar

You can apply CSS code on progress tag.

1. `progress{`
2. `width: 300px;`
3. `height: 30px;`
4. `}`

HTML Progress Tag with JavaScript

The <progress> tag should be used in conjunction with JavaScript to display the progress of a task.

1. `<script>`
2. `var gvalue=1;`
3. `function abc(){`
4. `var progressExample = document.getElementById ('progress-javascript-example');`
5. `setInterval (function ()`
6. `{`
7. `if(gvalue<100){`
8. `gvalue++;`
9. `progressExample.value =gvalue;`
10. `}`
11. `abc();`
12. `}, 1000);`

13. }
14. `</script>`
15. `<progress id="progress-javascript-example" min="1" max="100"></progress>`
16. `

`
17. `<button onclick="abc()">click me</button>`

HTML Form

An **HTML form** is a *section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc. .

Why use HTML Form

HTML forms are required if you want to collect some data from of the site visitor.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

1. `<form action="server url" method="get|post">`
 2. `//input controls e.g. textfield, textarea, radiobutton, button`
 3. `</form>`
-

HTML Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<code><form></code>	It defines an HTML form to enter inputs by the used side.
<code><input></code>	It defines an input control.

<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

HTML 5 Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<datalist>	It specifies a list of pre-defined options for input control.
<keygen>	It defines a key-pair generator field for forms.
<output>	It defines the result of a calculation.

HTML <form> element

The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

Note: The <form> element does not itself create a form but it is container to contain all required form elements, such as <input>, <label>, etc.

Syntax:

1. **<form>**

2. //Form elements
 3. `</form>`
-

HTML `<input>` element

The HTML `<input>` element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input field to gather different information from user. Following is the example to show the simple text input.

Example:

1. `<body>`
2. `<form>`
3. Enter your name `
`
4. `<input type="text" name="username">`
5. `</form>`
6. `</body>`

HTML TextField Control

The `type="text"` attribute of input tag creates textfield control also known as single line textfield control. The name attribute is optional, but it is required for the server side component such as JSP, ASP, PHP etc.

1. `<form>`
2. First Name: `<input type="text" name="firstname"/>
`
3. Last Name: `<input type="text" name="lastname"/>
`
4. `</form>`

Note: If you will omit 'name' attribute then the text field input will not be submitted to server.

HTML `<textarea>` tag in form

The `<textarea>` tag in HTML is used to insert multiple-line text in a form. The size of `<textarea>` can be specified either using "rows" or "cols" attribute or by CSS.

Example:

1. `<!DOCTYPE html>`
2. `<html>`

3. `<head>`
4. `<title>Form in HTML</title>`
5. `</head>`
6. `<body>`
7. `<form>`
8. Enter your address: `
`
9. `<textarea rows="2" cols="20"></textarea>`
10. `</form>`
11. `</body>`
12. `</html>`

Label Tag in Form

It is considered better to have label in form. As it makes the code parser/browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

NOTE: It is good to use <label> tag with form, although it is optional but if you will use it, then it will provide a focus when you tap or click on label tag. It is more worthy with touchscreens.

1. `<form>`
2. `<label for="firstname">First Name: </label>
`
3. `<input type="text" id="firstname" name="firstname"/>
`
4. `<label for="lastname">Last Name: </label>`
5. `<input type="text" id="lastname" name="lastname"/>
`
6. `</form>`

HTML Password Field Control

The password is not visible to the user in password field control.

1. `<form>`
2. `<label for="password">Password: </label>`
3. `<input type="password" id="password" name="password"/>
`
4. `</form>`

HTML 5 Email Field Control

The email field is new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

1. `<form>`
2. `<label for="email">Email: </label>`
3. `<input type="email" id="email" name="email"/>
`
4. `</form>`

Note: If we will not enter the correct email, it will display error like:

Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use one name for all the radio buttons, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

1. `<form>`
2. `<label for="gender">Gender: </label>`
3. `<input type="radio" id="gender" name="gender" value="male"/>Male`
4. `<input type="radio" id="gender" name="gender" value="female"/>Female`
 `
`
5. `</form>`

Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

1. `<form>`
2. Hobby: `
`
3. `<input type="checkbox" id="cricket" name="cricket" value="cricket"/>`
4. `<label for="cricket">Cricket</label>
`
5. `<input type="checkbox" id="football" name="football" value="football"/>`
6. `<label for="football">Football</label>
`
7. `<input type="checkbox" id="hockey" name="hockey" value="hockey"/>`
8. `<label for="hockey">Hockey</label>`
9. `</form>`

Note: These are similar to radio button except it can choose multiple options at a time and radio button can select one button at a time, and its display.

Submit button control

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

1. **<input type="submit" value="submit">**

The type = submit , specifying that it is a submit button

The value attribute can be anything which we write on button on web page.

The name attribute can be omit here.

Example:

1. **<form>**
2. **<label for="name">**Enter name**</label>
**
3. **<input type="text" id="name" name="name">
**
4. **<label for="pass">**Enter Password**</label>
**
5. **<input type="Password" id="pass" name="pass">
**
6. **<input type="submit" value="submit">**
7. **</form>**

HTML <fieldset> element:

The <fieldset> element in HTML is used to group the related information of a form. This element is used with <legend> element which provide caption for the grouped elements.

Example:

1. **<form>**
2. **<fieldset>**
3. **<legend>**User Information:**</legend>**
4. **<label for="name">**Enter name**</label>
**
5. **<input type="text" id="name" name="name">
**
6. **<label for="pass">**Enter Password**</label>
**

7. `<input type="Password" id="pass" name="pass">
`
8. `<input type="submit" value="submit">`
9. `</fieldset>`
10. `lt;/form>`

HTML Form Example

Following is the example for a simple form of registration.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Form in HTML</title>`
5. `</head>`
6. `<body>`
7. `<h2>Registration form</h2>`
8. `<form>`
9. `<fieldset>`
10. `<legend>User personal information</legend>`
11. `<label>Enter your full name</label>
`
12. `<input type="text" name="name">
`
13. `<label>Enter your email</label>
`
14. `<input type="email" name="email">
`
15. `<label>Enter your password</label>
`
16. `<input type="password" name="pass">
`
17. `<label>confirm your password</label>
`
18. `<input type="password" name="pass">
`
19. `
<label>Enter your gender</label>
`
20. `<input type="radio" id="gender" name="gender" value="male"/>Male
`
21. `<input type="radio" id="gender" name="gender" value="female"/>Female
`
22. `<input type="radio" id="gender" name="gender" value="others"/>others
`
23. `
Enter your Address:
`
24. `<textarea></textarea>
`
25. `<input type="submit" value="sign-up">`
26. `</fieldset>`
27. `</form>`
28. `</body>`
29. `</html>`

HTML Form Example

Let's see a simple example of creating HTML form.

```
1. <form action="#">
2. <table>
3. <tr>
4.   <td class="tdLabel"><label for="register_name" class="label">Enter name:</label></td>
5.   <td><input type="text" name="name" value="" id="register_name" style="width:160px"/></td>
6. </tr>
7. <tr>
8.   <td class="tdLabel"><label for="register_password" class="label">Enter password:</label></td>
9.   <td><input type="password" name="password" id="register_password" style="width:160px"/></td>
10. </tr>
11. <tr>
12.   <td class="tdLabel"><label for="register_email" class="label">Enter Email:</label></td>
13.   <td>
14.     <input type="email" name="email" value="" id="register_email" style="width:160px"/></td>
15. </tr>
16. <tr>
17.   <td class="tdLabel"><label for="register_gender" class="label">Enter Gender:</label></td>
18.   <td>
19.     <input type="radio" name="gender" id="register_gendermale" value="male"/>
20.     <label for="register_gendermale">male</label>
21.     <input type="radio" name="gender" id="register_genderfemale" value="female"/>
22.     <label for="register_genderfemale">female</label>
23.   </td>
24. </tr>
25. <tr>
26.   <td class="tdLabel"><label for="register_country" class="label">Select Country:</label></td>
27.   <td><select name="country" id="register_country" style="width:160px">
28.     <option value="india">india</option>
29.     <option value="pakistan">pakistan</option>
30.     <option value="africa">africa</option>
```



```
31. <option value="china">china</option>
32. <option value="other">other</option>
33. </select>
34. </td>
35. </tr>
36. <tr>
37. <td colspan="2"><div align="right"><input type="submit" id="register_0" value
   = "register" />
38. </div></td>
39. </tr>
40. </table>
41. </form>
```

HTML5 Semantics

In any language, it is essential to understand the meaning of words during communication. And if this is a computer communication then it becomes more critical. So HTML5 provides more semantic elements which make easy understanding of the code.

Hence Semantics defines the meaning of words and phrases, i.e.

Semantic elements= elements with a meaning. Semantic elements have a simple and clear meaning for both, the browser and the developer.

For example:

In HTML4 we have seen <div>, etc. are which are non-semantic elements. They don't tell anything about its content.

On the other hand, <form>, <table>, and <article> etc. are semantic elements because they clearly define their content.

HTML5 semantic elements are supported by all major browsers.

Why to use semantic elements?

In HTML4, developers have to use their own id/class names to style elements: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc.

This is so difficult for search engines to identify the correct web page content. Now in HTML5 elements (<header> <footer> <nav> <section> <article>), this will become easier. It now allows data to be shared and reused across applications, enterprises, and communities."

Semantic elements can increase the accessibility of your website, and also helps to create a better website structure.

Semantic Elements in HTML5

Index	Semantic Tag	Description
1.	<article>	Defines an article
2.	<aside>	Defines content aside from the page content
3.	<details>	Defines additional details that the user can view or hide
4.	<figcaption>	Defines a caption for a <figure> element
5.	<figure>	Specifies self-contained content, like illustrations, diagrams, photos, co
6.	<footer>	Defines a footer for a document or section
7.	<header>	Specifies a header for a document or section
8.	<main>	Specifies the main content of a document
9.	<mark>	Defines marked/highlighted text
10.	<nav>	Defines navigation links
11.	<section>	Defines a section in a document
12.	<summary>	Defines a visible heading for a <details> element
13.	<time>	Defines a date/time

Some important semantic elements in HTML5

HTML5 <article> Element

HTML `<article>` element defines article content within a document, page, application, or a website. It can be used to represent a forum post, a magazine, a newspaper article, or a big story.

Example:

1. `<article>`
2. `<h2>Today's highlights</h2>`
3. `<p>First story</p>`
4. `<p>Second story</p>`
5. `<p>Third story</p>`
6. `</article>`

HTML5 `<aside>` Element

The `<aside>` element represent the content which is indirectly giving information to the main content of the page. It is frequently represented as a sidebar.

Example:

1. `<body>`
2. `<h2>My last year memories</h2>`
3. `<p>I have visited Paris with my friends last month. This was the memorable journey and i wish to go there again.</p>`
4. `<aside>`
5. `<h4>Paris</h4>`
6. `<p>Paris, France's capital, is a major European city and a global center for art, fashion, gastronomy and culture</p>`
7. `</aside>`
8. `</body>`

HTML5 `<section>` Element

The `<section>` element is used to represent the standalone section within an HTML document. A page can have various sections and each section can contain any content, but headings for each section is not mandatory.

Example:

1. `<h2>Web designing Tutorial</h2>`
2. `<section>`
3. `<h3>HTML</h3>`

4. `<p>HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications.</p>`
5. `</section>`
6. `<section>`
7. `<h3>CSS</h3>`
8. `<p>CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML.</p>`
9. `</section>`

Nesting `<article>` tag in `<section>` tag or Vice Versa?

We know that the `<article>` element specifies independent, self-contained content and the `<section>` element defines section in a document.

In HTML, we can use `<section>` elements within `<article>` elements, and `<article>` elements within `<section>` elements.

We can also use `<section>` elements within `<section>` elements, and `<article>` elements within `<article>` elements.

For example:

In a newspaper, the sport `<article>` in the sport section, may have a technical section in each `<article>`.

HTML5 `<nav>` Element

The HTML `<nav>` element is used to define a set of navigation links.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<body>`
4. `<nav>`
5. `HTML |`
6. `Java |`
7. `PHP |`
8. `CSS`
9. `</nav>`
10. `</body>`
11. `</html>`

Nesting <article> tag in <section> tag or Vice Versa?

We know that the <article> element specifies independent, self-contained content and the <section> element defines section in a document.

In HTML, we can use <section> elements within <article> elements, and <article> elements within <section> elements.

We can also use <section> elements within <section> elements, and <article> elements within <article> elements.

For example:

In a newspaper, the sport <article> in the sport section, may have a technical section in each <article>.

HTML5 <nav> Element

The HTML <nav> element is used to define a set of navigation links.

Example:

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <nav>
5. HTML |
6. Java |
7. PHP |
8. CSS
9. </nav>
10. </body>
11. </html>

HTML5 <header> Element

The <header> element represent the header of the document which can contain introductory content or navigation links.

Example:

1. <header>
2. <h1>Welcome to Web123.com</h1>
3. <nav>
4.
5. Home |

6. About us |
7. Contact us
8.
9. </nav>
10. </header>

HTML5 <footer> Element

The <footer> tag defines the footer of an HTML document or page.

Example:

1. <footer>
2. <p>© Copyright 2019. All rights reserved. </p>
3. </footer>

HTML Canvas Tag

The **HTML canvas element** provides HTML a bitmapped surface to work with. It is used to draw graphics on the web page.

The **HTML 5 <canvas> tag** is used to draw graphics using scripting language like JavaScript.

The <canvas> element is only a container for graphics, you must need a scripting language to draw the graphics. The <canvas> element allows for dynamic and scriptable rendering of 2D shapes and bitmap images.

It is a low level, procedural model that updates a bitmap and does not have a built-in scene. There are several methods in canvas to draw paths, boxes, circles, text and add images.

How to create a HTML canvas?

A canvas is a rectangle like area on an HTML page. It is specified with canvas element. By default, the <canvas> element has no border and no content, it is like a container.

1. <canvas id = "mycanvas" width ="200" height ="100"> </canvas>

HTML 5 Canvas Tag Example

1. <canvas id="myCanvas1" width="300" height="100" style="border:2px solid;">
2. Your browser does not support the HTML5 canvas tag.
3. </canvas>

Note: It is always necessary to specify the id attribute and the height & width attribute to define the size of the canvas. You can have multiple canvas elements on one HTML page.

HTML Canvas Tag with JavaScript

A canvas is a two dimensional grid.

Coordinates (0,0) defines the upper left corner of the canvas. The parameters (0,0,200,100) is used for fillRect() method. This parameter will fill the rectangle start with the upper-left corner (0,0) and draw a 200 * 100 rectangle.

1. `<canvas id="myCanvas" width="250" height="150" style="border:1px solid #c3c3c3;">`
2. Your browser does not support the HTML5 canvas tag.
3. `</canvas>`
4. `<script>`
5. `var c = document.getElementById("myCanvas");`
6. `var cctx = c.getContext("2d");`
7. `cctx.fillStyle = "#FF0000";`
8. `cctx.fillRect(0,0,200,100);`
9. `</script>`

Drawing Line on Canvas

If you want to draw a straight line on the canvas, you can use the following two methods.

moveTo(x,y): It is used to define the starting point of the line.

lineTo(x,y): It is used to define the ending point of the line.

If you draw a line which starting point is (0,0) and the end point is (200,100), use the stroke method to draw the line.

1. `<canvas id="myCanvasLine" width="200" height="100" style="border:1px solid #d3d3d3;">`
2. Your browser does not support the HTML5 canvas tag. `</canvas>`
3. `<script>`
4. `var c = document.getElementById("myCanvasLine");`
5. `var cctx = c.getContext("2d");`
6. `cctx.moveTo(0,0);`
7. `cctx.lineTo(200,100);`

8. `ctx.stroke();`
9. `</script>`

Drawing Circle on Canvas

If you want to draw a circle on the canvas, you can use the `arc()` method:

1. `arc(x, y, r, start, stop)`

To sketch circle on HTML canvas, use one of the `ink()` methods, like `stroke()` or `fill()`.

1. `<canvas id="myCanvasCircle" width="200" height="100" style="border:1px solid #d3d3d3;">`
2. Your browser does not support the HTML5 canvas tag. `</canvas>`
3. `<script>`
4. `var c = document.getElementById("myCanvasCircle");`
5. `var cctx = c.getContext("2d");`
6. `ctx.beginPath();`
7. `ctx.arc(95,50,40,0,2*Math.PI);`
8. `ctx.stroke();`
9. `</script>`

Drawing text on canvas

There are property and methods used for drawing text on the canvas.

font property: It is used to define the font property for the text.

fillText(text,x,y) method: It is used to draw filled text on the canvas. It looks like bold font.

strokeText(text,x,y) method: It is also used to draw text on the canvas, but the text is unfilled.

Let's see **fillText()** method example.

1. `<canvas id="myCanvasText1" width="300" height="100" style="border:1px solid #d3d3d3;">`
2. Sorry! Your browser does not support the HTML5 canvas tag. `</canvas>`
3. `<script>`
4. `var c = document.getElementById("myCanvasText1");`
5. `var cctx = c.getContext("2d");`

6. `ctx.font = "30px Arial";`
7. `ctx.fillText("Hello JavaTpoint",10,50);`
8. `</script>`

Let's see **strokeText()** method example.

1. `<canvas id="myCanvasText2" width="300" height="100" style="border:1px solid #d3d3d3;">`
2. Sorry!Upgrade your browser. It does not support the HTML5 canvas tag.`</canvas>`
3. `<script>`
4. `var c = document.getElementById("myCanvasText2");`
5. `var cctx = c.getContext("2d");`
6. `ctx.font = "30px Arial";`
7. `ctx.strokeText("Hello JavaTpoint",10,50);`
8. `</script>`

HTML5 Migration

HTML5 migration specifies that how to migrate from HTML4 to HTML5. Let?s see how to convert HTML4 page into HTML5 page without any problem in content or structure.

Table:

In HTML4	In HTML5
<code><div id="header"></code>	<code><header></code>
<code><div id="menu"></code>	<code><nav></code>
<code><div id="content"></code>	<code><section></code>
<code><div class="article"></code>	<code><article></code>
<code><div id="footer"></code>	<code><footer></code>

Example:

```
1. <!DOCTYPE HTML PUBLIC "-
   //W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2. <html lang="en">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5. <title>HTML4</title>
6. <style>
7. body {
8.     font-family: Verdana,sans-serif;
9.     font-size: 0.9em;
10. }
11.
12. div#header, div#footer {
13.     padding: 10px;
14.     color: white;
15.     background-color: black;
16. }
17.
18. div#content {
19.     margin: 5px;
20.     padding: 10px;
21.     background-color: lightgrey;
22. }
23.
24. div.article {
25.     margin: 5px;
26.     padding: 10px;
27.     background-color: white;
28. }
29.
30. div#menu ul {
31.     padding: 0;
32. }
33.
34. div#menu ul li {
35.     display: inline;
36.     margin: 5px;
37. }
38. </style>
39. </head>
```

```
40. <body>
41.
42. <div id="header">
43. <h1>JavaTpoint Times</h1>
44. </div>
45.
46. <div id="menu">
47. <ul>
48. <li>Tutorials</li>
49. <li>Technology</li>
50. <li>Blog</li>
51. </ul>
52. </div>
53.
54. <div id="content">
55. <h2>Tutorials Section</h2>
56. <div class="article">
57. <h2>Tutorial1</h2>
58. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.

59. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
60. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.

61. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
62. </div>
63. <div class="article">
64. <h2>Tutorial2</h2>
65. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.

66. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
67. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.

68. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
69. </div>
70. </div>
71.
72. <div id="footer">
73. <p>© 2018 JavaTpoint Times. All rights reserved.</p>
74. </div>
75.
76. </body>
77. </html>
```

Change HTML4 Doctype to HTML5 Doctype

HTML4 Doctype Syntax:

1. `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`

HTML5 Doctype Syntax:

1. `<!DOCTYPE html>`

Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
5. `<title>HTML5</title>`
6. `<style>`
7. `body {`
8. `font-family: Verdana,sans-serif;`
9. `font-size: 0.9em;`
10. `}`
11. `div#header, div#footer {`
12. `padding: 10px;`
13. `color: white;`
14. `background-color: black;`
15. `}`
16.
17. `div#content {`
18. `margin: 5px;`
19. `padding: 10px;`
20. `background-color: lightgrey;`
21. `}`
22. `div.article {`
23. `margin: 5px;`
24. `padding: 10px;`
25. `background-color: white;`
26. `}`
27. `div#menu ul {`
28. `padding: 0;`
29. `}`
30. `div#menu ul li {`

```
31. display: inline;
32. margin: 5px;
33. }
34. </style>
35. </head>
36. <body>
37. <div id="header">
38. <h1>JavaTpoint Times</h1>
39. </div>
40.
41. <div id="menu">
42. <ul>
43. <li>Tutorials</li>
44. <li>Technology</li>
45. <li>Blog</li>
46. </ul>
47. </div>
48. <div id="content">
49. <h2>Tutorials Section</h2>
50. <div class="article">
51. <h2>Tutorial1</h2>
52. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
53. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
54. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
55. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
56. </div>
57. <div class="article">
58. <h2>Tutorial2</h2>
59. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
60. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
61. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
62. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
63. </div>
64. </div>
65. <div id="footer">
66. <p>© 2018 JavaTpoint Times. All rights reserved.</p>
67. </div>
68.
69. </body>
70. </html>
```

