

Share on your Social Media



HTML Tutorial

Published On: June 22, 2024

HyperText Markup Language is the common markup language used in texts intended for web browsers. It defines the structure and substance of web content. Beginners as well as professionals can design personalized, dynamic, and interactive websites with this **basic HTML tutorial**.

In this tutorial, you will learn the following major HTML concepts:

- HTML Basics
 - Features of HTML
 - HTML Editors
 - HTML Elements
 - HTML Attributes
 - HTML Headings
 - HTML Paragraphs
 - HTML Fonts
- HTML and CSS
- HTML Forms
- HTML Graphics
- HTML APIs
- HTML Examples

[Download HTML Tutorial PDF](#)

HTML Basics

Featured Articles

 **Want to know more about becoming an expert in IT?**

[Click Here to Get Started](#) >>

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER

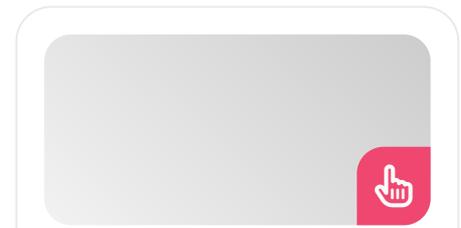
IBM

Quick Enquiry

Related Courses at SLA

- ➔ **HTML Training in OMR**
- ➔ **HTML Training in Chennai**

Related Posts



HTML stands for HyperText Markup Language. HTML is not a programming language; rather, it is a markup language. The language HTML is widely used on the internet. It is used to create web pages and web apps. Only HTML can be used to create a static website.

Features of HTML

- HTML is basic and easy. It is simple to comprehend and adjust.
- As HTML contains so many formatting tags, creating an effective presentation with it is quite simple.
- It is a markup language, it offers an adaptable means of creating web pages in addition to text.
- It makes it easier for programmers to add links to web pages (using the html anchor tag), which increases user interest in surfing.
- It is displayable on any platform, including Windows, Linux, Macintosh, and others, making it platform-independent.
- It makes it easier for programmers to enhance the appearance and interactivity of websites by adding graphics, videos, and sound.
- As HTML is a case-insensitive language, we can utilize tags un either upper- or lower-case.

HTML Editors

Page layout and editing can be done with the help of experienced HTML editors. To learn HTML, however, we suggest using a simple text editor such as Notepad (PC) or TextEdit (Mac). To make your first web page using Notepad or TextEdit, follow the instructions below.

Step 1: Open Notepad

Windows 8 or above: Click on the Start Screen icon, which is located at the lower-left corner of your screen. Type Notepad.

Step 2: Write Code

```
<!DOCTYPE html>
```

Oracle Tutorial

Published On: June 21, 2024

Oracle Tutorial The fundamentals of the Oracle database are simple to learn and will take...



Web Design Tutorial

Published On: May 29, 2024

Web Design Tutorial This tutorial page will guide you through the fundamental concepts, principles, elements...



WordPress Tutorial

Published On: May 29, 2024

Introduction This tutorial will take through all the fundamental concepts in Wordpress. Here you will...



Web Development Tutorial

Published On: May 28, 2024

Introduction In this tutorial page, you will gain all the fundamental knowledge related to Web...

```
<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>My first Heading</h1>

<p>My first paragraph.</p>

</body>

</html>
```

In this example,

<!DOCTYPE html>: This document is defined as an HTML5 document.

<html>: An HTML page's root element.

<head>: The HTML page's metadata.

<title>: The HTML page's title, which appears in the tab or title bar of the browser.

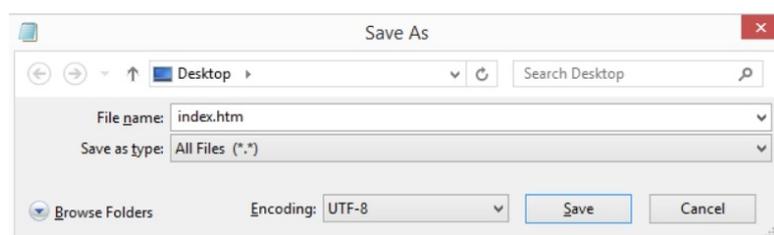
<body>: It defines the document's body, which also stores all of its visible components, including tables, lists, headings, paragraphs, images, and hyperlinks.

The <h1>: It defines a large heading.

The <p>: It defines a paragraph.

Step 3: Keep the HTML page saved.

Download and save the file on your PC. Click the File > Save as option in Notepad.



Step 4: Open your browser and view the HTML page.

Double-click the saved HTML file or use the context menu to select "Open with" to open it in your preferred browser.

The recommended encoding for HTML files is UTF-8, so give the file the name "index.htm" and set its encoding to that. This is how it shows up in the web browser:



[**Download HTML Interview Questions**](#)

HTML Elements

The foundation of a webpage is made up of HTML elements. It is employed to produce web page components.

These elements make up an HTML document tree and provide instructions on how to construct HTML documents as well as what types of material go where in an HTML document.

Differences Between Tags and Elements

HTML tags are used to build all HTML elements. However, the beginning and ending tags of an HTML

element define it. The HTML document's content, which further establishes the web page's structure and layout, is contained within these tags.

Example

A paragraph's opening tag is `<p>`, and its closing tag is `</p>`,

In contrast, `<p>`this paragraph element is called paragraph`</p>`.

Simple HTML Element

Two HTML elements will be created in this example:

- A header element
- A paragraph element.

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Smiple Element</title>

</head>

<body>

  <h1>My Header Element</h1>

  <p>My Paragrapph Element</p>

</body>

</html>
```

Nested HTML Element

We will create a parent element, two child elements inside the parent element, and a nested element in this example.

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Nested Element</title>

</head>

<body>
```

```
<p>
  This is parent element containing
  <span>Child Elelement 1</span> &
  <span>Child Elelement 2</span>.
</p>
</body>
</html>
```

Empty HTML Element

We're going to make an empty element in between two simple elements in this example.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Smiple Element</title>
  </head>
  <body>
    <h1>This is Header Element</h1>
    <hr>
    <p>This is Paragrapgh Element</p>
  </body>
```

Are elements in HTML case-sensitive?

HTML components do allow case insensitivity; therefore, tag names may contain both capital and lowercase letters.

HTML Attributes

Some attributes have specific properties and guidelines, such as how they should be used on HTML elements or tags.

- HTML offers more details about the components.
- It is usually necessary to include attributes in the opening tag.
- Every HTML element, except a handful like

<head> and <title>, can contain attributes.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Attribute Example</title>

</head>

<body>

  <p align="left">Left Aligned</p>

  <p align="center">Center Aligned</p>

  <p align="right">Right Aligned</p>

</body>

</html>
```

HTML Headings

An HTML page's hierarchy (levels) and content structure are specified using its headings.

With h1 serving as the top-level heading and denoting the most significant material or main heading, and h2, h3, h4, and so on serving as subheadings for subtopics, they fashion a visual hierarchy.

Reasons to use Headings

Headings are essential for organizing text and giving new sections or topics a distinct visual cue as to when they begin.

- To highlight important topics.
- Impact on SEO.

Example

```
<!DOCTYPE html>

<html>

<body>

  <h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

```
<h4>This is heading 4</h4>
```

```
<h5>This is heading 5</h5>
```

```
<h5>This is heading 6</h5>
```

```
</body>
```

```
</html>
```

HTML Paragraphs

HTML paragraphs are block-level components that give text on a webpage organization and formatting.

Paragraph using p tag

A paragraph element is created with the HTML p tag.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Paragraphs</title>

</head>

<body>

  <!-- HTML p Tag used -->

  <p>Welcome to the HTML tutorial by SLA</p>

</body>

</html>
```

Line Breaks with Paragraph

To manage the text layout, line breaks are inserted within paragraphs using the HTML
 tag.

Example

```
<!DOCTYPE html>

<html>
```

```
<head>

  <title>Enhancing Paragraphs </title>

</head>

<body>

  <p>This is a paragraph with a <br> line break. </p>

</body>

</html>
```

Headings with Paragraph

You can use HTML <h1> to <h6> tags with paragraphs to create a hierarchical structure.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>Enhancing Paragraphs </title>

</head>

<body>

  <h1>Main Heading</h1>

  <p> This is a paragraph beneath the main heading. </p>

</body>

</html>
```

Lists with Paragraphs

When a list of elements is required following a description, the HTML tag can be used. Lists can be used to organize text within paragraphs.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>Enhancing Paragraphs </title>

</head>
```

```
<body>

<ul>

  <li>Item 1</li>

  <li>Item 2</li>

</ul>

<p> This is a paragraph following an unordered list. </p>

</body>

</html>
```

HTML Fonts

When it comes to improving a website's usability and content readability, fonts are crucial. The HTML `` tag can be used to change the text on your website's style, size, and color.

Example

```
<!DOCTYPE html>

<html>

  <head>

    <title>Setting Font Size</title>

  </head>

  <body>

    <font size = "1">Font size = "1"</font><br />

    <font size = "2">Font size = "2"</font><br />

    <font size = "3">Font size = "3"</font><br />

    <font size = "4">Font size = "4"</font><br />

    <font size = "5">Font size = "5"</font><br />

    <font size = "6">Font size = "6"</font><br />

    <font size = "7">Font size = "7"</font>

  </body>

</html>
```

HTML Blocks

Block components are used to arrange web pages in a logical and meaningful manner. A range of

elements, such as <div>, <p>, <table>, and so on, are around the HTML blocks.

Two categories can be used to group all HTML elements.

- Block Elements
- Inline Elements

HTML Block Elements

<address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, <div>, <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1> – <h6>, <header>, <hr>, , <main>, <nav>, <noscript>, , <p>, <pre>, <section>, <table>, <tfoot>, , and <video>.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Block Level Elements</title>

</head>

<body>

  <h3>HTML Block Level Elements</h3>

  <p>

    This line will appear in the next line

    after Heading.

  </p>

  <hr>

  <p>

    This line will appear after Horizontal

    Line.

  </p>

</body>

</html>
```

HTML Inline Elements

Inline elements do not create new lines on their own; they might appear within the same line.

The following are some typical inline elements:

`<a>`, `<abbr>`, `<acronym>`, ``, `<bdo>`, `<big>`, `
`,
`<button>`, `<cite>`, `<code>`, `<dfn>`, ``, `<i>`, ``,
`<input>`, `<kbd>`, `<label>`, `<map>`, `<object>`, `<output>`,
`<q>`, `<samp>`, `<script>`, `<select>`, `<small>`, ``,
``, `<sub>`, `<sup>`, `<textarea>`, `<time>`, `<tt>`, and
`<var>`.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML inline Element</title>

</head>

<body>

  <h3>Inline Elements in HTML</h3>

  <!-- Using <b> inline element -->

  <p>This <b>paragraph</b> is bold. </p>

  <!-- Using <i> inline element -->

  <p>This is an <i>italic</i> paragraph.</p>

</body>

</html>
```

HTML Training

HTML and CSS

Cascading Style Sheets, or CSS, is a tool that controls how online pages appear in print or on displays. Different HTML element properties can be specified with ease and effectiveness thanks to Cascading Style Sheets (CSS). For a particular HTML element, you can specify several style properties

using CSS.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML CSS</title>

</head>

<body>

  <p style="color:green;font-size:24px;">

    Hello, World!

  </p>

</body>

</html>
```

Types of CSS

There are three ways to insert CSS in your HTML document; there isn't a type or variety in CSS.

External CSS: Use the HTML `<link>` tag to add a separate .css file containing the style sheet rules in your HTML text.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML External CSS</title>

  <link rel="stylesheet" type="text/css" href="/html/style.css">

</head>

<body>

  <p class="red">This is red</p>

  <p class="thick">This is thick</p>

  <p class="green">This is green</p>

  <p class="thick green">
```

This is thick and green

</p>

</body>

</html>

Internal CSS: Use the <style> tag to define style sheet rules in the HTML document's header section.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML Internal CSS</title>
```

```
<style type="text/css">
```

```
.red {
```

```
  color: red;
```

```
}
```

```
.thick {
```

```
  font-size: 20px;
```

```
}
```

```
.green {
```

```
  color: green;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="red">This is red</p>
```

```
<p class="thick">This is thick</p>
```

```
<p class="green">This is green</p>
```

```
<p class="thick green">
```

```
  This is thick and green
```

```
</p>
```

```
</body>
```

```
</html>
```

Inline CSS: Using the style property, define style sheet rules right alongside HTML components in inline CSS.

Example

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML Inline CSS</title>

</head>

<body>

  <p style="color:red;">This is red</p>

  <p style="font-size:20px;">This is thick</p>

  <p style="color:green;">This is green</p>

  <p style="color:green;font-size:20px;">

    This is thick and green

  </p>

</body>

</html>
```

HTML Forms

User data has been gathered using HTML forms, which are basic forms. HTML forms have interactive features and a variety of input forms, including checkboxes, radio buttons, text, numbers, email, and password fields.

Syntax

```
<form action = "Script URL" method = "GET|POST">
```

form controls like input, textarea etc.

```
</form>
```

Example

```
<!DOCTYPE html>

<html>
```

```
<head>

  <meta charset="utf-8">

  <title>Sample HTML Form </title>

</head>

<body>

  <!-- Start of the form element -->

  <form action = " ">

    <!-- various form controls -->

    <label for="first_name">First name:</label>

    <input type = "text" name = "first_name" />

    <br><br>

    <label for="first_name">Last name:</label>

    <input type = "text" name = "last_name" />

    <br><br>

    <input type = "submit">

  </form>

</body>

</html>
```

HTML Graphics

HTML Canvas

Graphics on a web page are created using the HTML `<canvas>` element.

`<canvas>` was used to generate the graphic on the left. A red rectangle, a gradient rectangle, a multicolored rectangle, and multicolored lettering are the four items displayed.

Syntax

```
var canvas =
document.getElementById("mycanvas");
```

Example

```
<!DOCTYPE html>
```

```
<html>

<head>

  <style>

    #mycanvas{border:1px solid red;}

  </style>

</head>

<body>

  <canvas id="mycanvas" width="100" height="100"></canvas>

</body>

</html>
```

Drawing Rectangles

Three techniques are available for drawing rectangles on a canvas.

fillRect(x,y,width,height): With this procedure, a filled rectangle is drawn.

strokeRect(x,y,width,height): This process creates a rectangle shape.

clearRect(x,y,width,height): Using this technique, the designated region is cleared and completely transparent.

Example

```
<!DOCTYPE html>

<html>

<head>

  <style>

    #test {

      width: 100px;

      height:100px;

      margin: 0px auto;

    }

  </style>
```

```
<script type="text/javascript">

function drawShape(){

var canvas = document.getElementById('mycanvas');

if (canvas.getContext){

var ctx = canvas.getContext('2d');

ctx.fillRect(25,25,100,100);

ctx.clearRect(45,45,60,60);

ctx.strokeRect(50,50,50,50);

} else {

alert('You need Safari or Firefox 1.5+ to see this demo.');

}

}

</script>

</head>

<body id="test" onload="drawShape();">

<canvas id="mycanvas"></canvas>

</body>

</html>
```

Using Images

beginPath(): This function returns the path as it is now.

moveTo(x, y): This function takes a point as input and builds a new subpath with it.

closePath(): This function creates a new subpath with a point that is the same as the start and end of the newly closed subpath and marks the current subpath as closed.

fill(): This function applies the current fill style to the subpaths.

stroke(): This function applies the current stroke style to the subpaths.

drawImage(image, dx, dy): This function creates a

canvas on which the supplied image is drawn. An image or canvas object is referenced here.

Example

```
<!DOCTYPE HTML>

<html>

<head>

  <script type = "text/javascript">

    function drawShape() {

      var canvas = document.getElementById('mycanvas');

      if (canvas.getContext) {

        var ctx = canvas.getContext('2d');

        var img = new Image();

        img.src = '/html/images/backdrop.jpg';

        img.onload = function() {

          ctx.drawImage(img,0,0);

          ctx.beginPath();

          ctx.moveTo(30,96);

          ctx.lineTo(70,66);

          ctx.lineTo(103,76);

          ctx.lineTo(170,15);

          ctx.stroke();

        }

      } else {

        alert("You can see this demo.");

      }

    }

  </script>

</head>

<body onload = "drawShape();">

  <canvas id = "mycanvas"></canvas>

</body>
```

[Download HTML Syllabus PDF](#)

HTML Graphics with SVG

Scalable Vector Visuals, or SVG for short, is a markup language based on XML that may be used to create scalable 2D visuals and graphical applications.

There are two methods for integrating SVG files into HTML:

- Using tag
- Using <svg> tag

Using tag

The src attribute of the tag allows us to directly embed SVG files within our webpage, as demonstrated below. Either the path or an online link to the SVG picture can be passed.

```
<img src = "yourfilename.svg" height = "100px" width = "200px" />
```

Using <svg> tag

HTML permits direct SVG embedding through the use of the <svg>...</svg> element, which has the following syntax:

```
<svg>
```

```
  <!-- code to create graphics -->
```

```
</svg>
```

HTML APIs

HTML Geolocation API: To find the position of a user, use the HTML Geolocation API.

```
var geolocation = navigator.geolocation;
```

Example

```
function getLocation() {

    var geolocation = navigator.geolocation;

    geolocation.getCurrentPosition(showLocation,
    errorHandler);

}
```

HTML Drag and Drop API: This enables the user to move an element to a new location by clicking and holding down the mouse button over it, and then releasing the mouse button to drop it there.

Example

```
function EnterHandler(event) {

    DataTransfer dt = event.dataTransfer;

    .....

}
```

HTML Web Workers API: The URL of a JavaScript file containing the code that the worker will run is used to initialize Web Workers.

```
var worker = new Worker('bigLoop.js');
```

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Big for loop</title>

<script>

var worker = new Worker('bigLoop.js');

worker.onmessage = function(event) {

    alert("Completed " + event.data + " iterations");

};

function sayHello() {

    alert("Hello sir....");
```

```
}  
  
</script>  
  
</head>  
  
<body>  
  
  <input type="button" onclick="sayHello();" value="Say Hello" />  
  
</body>  
  
</html>
```

Conclusion

There are many things that this tutorial can't cover in learning HTML. However, we hope this HTML tutorial will be helpful for you to grasp the basic understanding of HTML and CSS. To learn comprehensively with hands-on exposure, join our [HTML training in Chennai](#).

Share on your Social
Media



Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

Navigation

[About Us](#)

[Blog Posts](#)

[Careers](#)

[Contact](#)

[Placement Training](#)

[Corporate Training](#)

[Hire With Us](#)

[Job Seekers](#)

[SLA's Recently Placed Students](#)

[Reviews](#)

[Sitemap](#)

Important Links

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

[Disclaimer](#)

[Privacy Policy](#)

[Terms and Conditions](#)

Courses

[Python](#)

[Software Testing](#)

[Full Stack Developer](#)

[Java](#)

[Power BI](#)

[Clinical SAS](#)

[Data Science](#)

[Embedded](#)

[Cloud Computing](#)

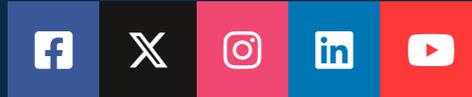
[Hardware and Networking](#)

[VBA Macros](#)

[Mobile App Development](#)

[DevOps](#)

Social Media Links



Review Sources

[Google](#)

[Trustpilot](#)

[Glassdoor](#)

[Mouthshut](#)

[Sulekha](#)

[Justdial](#)

[Ambitionbox](#)

[Indeed](#)

[Software Suggest](#)

[Sitejabber](#)