# Introduction to HTML

## What Is HTML?

HTML stands for *Hypertext Markup Language* and is most widely used to write web pages.

* 'Hypertext' refers to the way in which web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
* 'Markup Language' means that you use HTML to simply "mark-up" a text document with tags that tell a web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, etc. to facilitate the sharing of scientific information between researchers.

Now HTML is being widely used to format web pages with the help of different tags available in HTML language.

## HTML Document Structure

A typical HTML document has the following structure:

<!-- This is document declaration tag -->

<!DOCTYPE html>

<html>

   <head>

      <!-- Document header related tags -->

   </head>

   <body>

      <!-- Document body related tags -->

   </body>

</html>

We will study all the header and body tags in subsequent chapters.

# HTML Components

## Tags

HTML tags are enclosed within angle braces <tag-name>. Except few tags, most of the tags have their corresponding closing tags </tag-name>. For example, <html> has its closing tag </html> and <body> tag has its closing tag </body> tag, etc.

The list below shows basic tags commonly used in HTML documents:

|  |  |  |
| --- | --- | --- |
| **Groups** | **Tags** | **Descriptions** |
| **Main structure** | <!DOCTYPE ...> | Used by web browsers to understand the version of the HTML used in the document.  Current version of HTML is 5 and it makes use of the following declaration: <!DOCTYPE html> |
| <html> | Encloses the complete HTML document and mainly comprises of:   * Document header (by <head>...</head> tags) * Document body (by <body>...</body> tags) |
| <head> | Represents the document's header which can keep other HTML tags like <title>, <link>, <meta>, <base>, <style>, <script>, and <noscript>.  For more details about these tags, check [this section](https://www.tutorialspoint.com/html/html_header.htm). |
| <body> | Represents the document's body which keeps other HTML tags like <h1>, <div>, <p>, etc. |
| <!-- ... --> | Represents a comment which is ignored by all web browsers. |
| **Metadata** | <title> | * Used inside the <head> tag. * Describe the document title. |
| <link> | * Used inside the <head> tag. * Define the relationship between the current document and an external resource. |
| **Structural** | <h1> to <h6> | * Used inside the <body> tag. * Represents a heading.   Note: You can use different sizes for your headings. HTML has 6 levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. |
| <p> | * Used inside the <body> tag. * Represents a paragraph. |
| **Formatting**  **&**  **Layout** | <pre> | Forces text to follow the exact format of how it is written in the HTML document. |
| <b>, <i>, <u>, <strike>, <tt>, <sup>, <sub>, <ins>, <del>, <small>  <em>, <mark>, strong>, <abbr>, <acronym>, <q>, etc. | <https://www.tutorialspoint.com/html/html_formatting.htm>  <https://www.tutorialspoint.com/html/html_phrase_elements.htm> |
| <br /> | Breaks the current line and starts from the next line. |
| <hr /> | Creates a line from the current position in the document to the right margin and breaks the line accordingly. |
| <div> | Specifies a division or a section in a document. |
| **Image** | <img> | <https://www.tutorialspoint.com/html/html_images.htm> |
| **Table** | <table> | <https://www.tutorialspoint.com/html/html_tables.htm> |
| **List** | <ul>  <ol>  <dl> | <https://www.tutorialspoint.com/html/html_lists.htm> |
| **Link** | <a> | <https://www.tutorialspoint.com/html/html_text_links.htm>  <https://www.tutorialspoint.com/html/html_image_links.htm>  <https://www.tutorialspoint.com/html/html_email_links.htm>  <https://www.tutorialrepublic.com/html-tutorial/html-links.php> |
| **Frame** | <frame>  <iframe> | <https://www.tutorialspoint.com/html/html_frames.htm>  <https://www.tutorialspoint.com/html/html_iframes.htm> |
| [**Grouping**](#_Grouping_Elements) | <div>  <span> | Check [Block section](#_1fob9te). |
| **Font**  **(face, size and** [**color**](#_Colors)**)** | <basefont>  <font> | <https://www.tutorialspoint.com/html/html_fonts.htm>  Note: The font tag deprecated and it is supposed to be removed in a future version of HTML. It's suggested to use CSS styles to manipulate your fonts. |
| [**Forms**](#_Forms) | <form> | <https://www.tutorialspoint.com/html/html_forms.htm> |
| **Embed multimedia** | <embed> | <https://www.tutorialspoint.com/html/html_embed_multimedia.htm> |
| **Marquees** | <marquee> | <https://www.tutorialspoint.com/html/html_marquees.htm> |
| [**CSS**](#_CSS) | <style> | Check [CSS section](#_2et92p0). |
| **Scripting** | <script>  <noscript> | Check JavaScript section. |
| **…** |  |  |

**Check a** [**full list of HTML5 tags**](https://www.tutorialrepublic.com/html-reference/html5-tags.php)**.**

## Elements

An HTML element is defined within **a start tag (or opening tag) and an end tag (or closing tag), with content in between.** For examples:

|  |  |  |
| --- | --- | --- |
| **Start Tag** | **Content** | **End Tag** |
| <p> | This is paragraph content. | </p> |
| <h1> | This is heading content. | </h1> |
| <br /> |  |  |

Here <p>....</p> is an HTML element, <h1>...</h1> is another HTML element.

**Note**

Some elements don't require the end tag to be present, simply because they don't need to be closed, such as <br />, <hr /> and <img ... />. These are known as **void elements**.

## Attributes

So far, we have seen some HTML tags and their usage in the simplest form. But most HTML tags can also have *attributes*, providing extra bits of information.

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

* The *name* is the property you want to set. For example, the paragraph <p> element in the example carries an attribute whose name is align, which you can use to indicate the alignment of paragraph on the page.
* The *value* is what you want the value of the property to be set and always put within quotations.

The below example shows 3 possible values of align attribute: left, center and right.

<!DOCTYPE html>

<html>

   <head>

      <title>Align attribute example</title>

   </head>

   <body>

      <p align = "left">This is left aligned</p>

      <p align = "center">This is center aligned</p>

      <p align = "right">This is right aligned</p>

   </body>

</html>

Output:

This is left aligned

This is center aligned

This is right aligned

**Notes**:

* Attribute names and values are **case-insensitive**. However, the World Wide Web Consortium (W3C) recommends lowercase formatting.
* Both **single and double quotes** can be used to quote attribute values. However, double quotes are most common. In situations where the attribute value itself contains double quotes, it is necessary to wrap the value in single quotes, e.g., value='John "Williams" Jr.'
* **Some attributes in HTML5 don't consist of name/value pairs but consists of just name**. These are called *Boolean* attributes. For examples: checked, disabled, readonly, required, etc. Check [common Boolean attributes](https://www.tutorialrepublic.com/codelab.php?topic=html&file=boolean-attributes).

### Core Attributes

The four core attributes that can be used on the majority of HTML elements are:

#### The 'id' Attribute

It uniquely identifies any element within an HTML page. There are two primary reasons that you might want to use it on an element:

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

Example:

<p id = "html">This explains what is HTML</p>

<p id = "css">This explains what is Cascading Style Sheet</p>

#### The 'title' Attribute

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

Example:

<!DOCTYPE html>

<html>

<head>

<title>The title attribute example</title>

</head>

<body>

<h1 title = "Hello HTML!">Titled heading tag example</h3>

</body>

</html>

Output:

**Titled heading tag example**

Now try to bring your cursor over "Titled heading tag example" and you will see "Hello HTML!" is coming out as a tooltip of the cursor.

#### The 'class' Attribute

It associates an element with CSS, and specifies the class of element. You will learn more about the use of the class attribute when you will learn [Cascading Style Sheet (CSS)](#_2et92p0). So for now you can avoid it.

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

class = "class-name1 class-name2 class-name3"

#### The 'style' Attribute

It allows you to specify CSS rules within the element.

For example:

<!DOCTYPE html>

<html>

<head>

<title>The style attribute</title>

</head>

<body>

<p style = "font-family:arial; color:#FF0000;">Some text...</p>

</body>

</html>

Output:

Some text...

### Internationalization Attributes

The three internationalization attributes that can be used on the majority of HTML elements are:

#### The 'dir' Attribute

It represents the direction in which the text should flow. It can take one of two values, as you can see in the table:

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

Example:

<!DOCTYPE html>

<html dir = "rtl">

<head>

<title>Display directions</title>

</head>

<body>

This is how IE 5 renders right-to-left directed text.

</body>

</html>

Output:

This is how IE 5 renders right-to-left directed text.

#### The 'lang' Attribute

It indicates the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the xml:lang attribute in new XHTML documents.

The values of the lang attribute are ISO-639 standard two-character language codes. Check [HTML Language Codes: ISO 639](https://www.tutorialspoint.com/html/language_iso_codes.htm) for a complete list of language codes.

Example:

<!DOCTYPE html>

<html lang = "en">

<head>

<title>English Language Page</title>

</head>

<body>

This page is using English Language

</body>

</html>

Output:

This page is using English Language

# Grouping Elements

## Different Types of Elements

All HTML elements can be categorized into two categories (a) Block-Level Elements (b) Inline Elements.

### Block Elements

Block elements all **start on their own new line**, and anything that follows them appears on its own new line. For example: <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, <address>, etc.

### Inline Elements

Inline elements, on the other hand, can **appear within sentences** and do not have to appear on a new line of their own. For example: <span>, <b>, <i>, <u>, <em>, <strong>, <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, <var>, etc.

## How to Group Elements

There are two important tags which we use very frequently to group various other HTML tags: (a) <div> and (b) <span>.

### The <div> Tag

This block-level tag plays a big role in grouping various HTML tags and applying CSS on group of elements. Moreover, it's also the most common way to define [HTML layouts](#_tyjcwt).

<!DOCTYPE html>

<html>

   <head>

      <title>HTML div Tag</title>

   </head>

   <body>

      <!-- First group of tags -->

      <div style = "color:red">

         <p>Following is a list of vegetables:</p>

         <ul>

            <li>Beetroot</li>

            <li>Ginger</li>

         </ul>

      </div>

      <!-- Second group of tags -->

      <div style = "color:green">

         <p>Following is a list of fruits:</p>

         <ul>

            <li>Apple</li>

            <li>Banana</li>

         </ul>

      </div>

   </body>

</html>

Output:

|  |
| --- |
| Following is a list of vegetables:   * Beetroot * Ginger   Following is a list of fruits:   * Apple * Banana |

### The <span> Tag

This inline tag is used to group inline-elements in an HTML document. It is mostly used with CSS, like the <div> tag.

<!DOCTYPE html>

<html>

   <head>

      <title>HTML span Tag</title>

   </head>

   <body>

      <p>This is <span style = "color:red">red</span> and this is

         <span style = "color:green">green</span>

</p>

   </body>

</html>

Output:

This is red and this is green

# Table

# Colors

The <body> tag has following attributes which can be used to set different colors:

* bgcolor − sets a color for the background of the page.
* text − sets a color for the body text.
* link − sets a color for linked text.
* alink − sets a color for active links or selected links.
* vlink − sets a color for visited links − that is, for linked text that you have already clicked on.

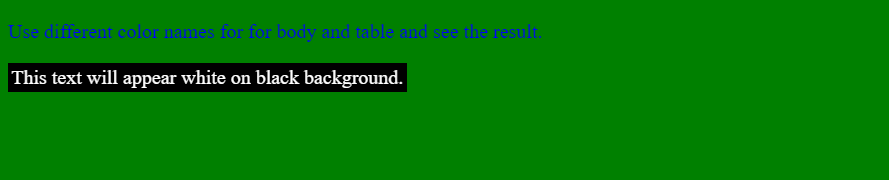
## HTML Color Coding Methods

There are following three different methods to set colors in your web page:

* Color names − You can specify color names directly like green, blue or red.
* Hex codes − A six-digit code representing the amount of red, green, and blue that makes up the color.
* Color decimal or percentage values − This value is specified using the rgb( ) property.

**Note**: Check a complete list of [HTML Color Name.](https://www.tutorialspoint.com/html/html_color_names.htm)

Below examples produce the same output as follows:



### Color Names

<!DOCTYPE html>

<html>

   <head>

      <title>HTML Colors by Name</title>

   </head>

   <body text = "blue" bgcolor = "green">

      <p>Use different color names for body and table and see the result.</p>

      <table bgcolor = "black">

         <tr>

            <td>

               <font color = "white">This text appears white on black bgr.</font>

            </td>

         </tr>

      </table>

   </body>

</html>

### Hex Codes

<!DOCTYPE html>

<html>

   <head>

      <title>HTML Colors by Hex</title>

   </head>

   <body text = "#0000FF" bgcolor = "#00FF00">

      <p>Use different color hexa for body and table and see the result.</p>

      <table bgcolor = "#000000">

         <tr>

            <td>

               <font color = "#FFFFFF">This text appears white on black bgr.</font>

            </td>

         </tr>

      </table>

   </body>

</html>

### RGB Values

<!DOCTYPE html>

<html>

   <head>

      <title>HTML Colors by RGB code</title>

   </head>

   <body text = "rgb(0,0,255)" bgcolor = "rgb(0,255,0)">

      <p>Use different color code for body and table and see the result.</p>

      <table bgcolor = "rgb(0,0,0)">

         <tr>

            <td>

               <font color = "rgb(255,255,255)">This text appears white on black bgr.</font>

            </td>

         </tr>

      </table>

   </body>

</html>

# Forms

<https://www.tutorialspoint.com/html/html_forms.htm>

# CSS

## What Is CSS?

Cascading Style Sheets (CSS) describe **how documents are presented** on screens. It provides easy and effective alternatives to specify various attributes for the HTML tags. So you can easily specify a number of style properties for a given HTML element.

Each CSS property has the form:

property-name:property-value;

## Non-CSS vs. CSS

Consider the following example which makes use of <font> tag and associated attributes to specify text color and font size:

<!DOCTYPE html>

<html>

   <head>

      <title>HTML CSS</title>

   </head>

   <body>

      <p><font color = "green" size = "5">Hello, World!</font></p>

   </body>

</html>

We can re-write above example with the help of CSS as follows:

<!DOCTYPE html>

<html>

   <head>

      <title>HTML CSS</title>

   </head>

   <body>

      <p style = "color:green; font-size:24px;" >Hello, World!</p>

   </body>

</html>

Both give the same output:

Hello, World!

## Different Ways to Use CSS in HTML

### External Style Sheet

If you need to use style sheet rules to various pages, then it's always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as .css and it will be included in HTML files using <link> tag.

Consider a style sheet file named style.css which has three CSS rules applicable to three classes defined for the HTML tags.

.red {

    color: red;

 }

 .green {

    color:green;

 }

 .thick {

    font-size:20px;

 }

Now let's make use of the above external CSS file in our HTML document:

<!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>

Output:

This is red

This is thick

This is green

This is thick and green

### Internal Style Sheet

If you want to apply style sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

**Note**: Rules defined in internal style sheet overrides the rules defined in an external CSS file.

Let's re-write above example:

<!DOCTYPE html>

<html>

   <head>

      <title>HTML Internal CSS</title>

      <style type = "text/css">

         .red {

            color: red;

         }

         .green {

            color:green;

         }

         .thick{

            font-size:20px;

         }

      </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 Style Sheet

You can apply style sheet rules directly to any HTML element using style attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

**Note**: Rules defined inlinely with the element overrides both the rules defined in an external CSS file and in an internal style sheet.

Let's re-write above 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>

## List of CSS Properties

<https://www.w3schools.com/cssref/>

<https://cssreference.io/>

# Layouts

Using the <div> elements is the most common method of creating layouts in HTML. This element is used for marking out a block of content, or set of other elements inside an HTML document. It can contain further other <div> elements if required.

For example:

<!DOCTYPE html>

<html>

   <head>

      <title>HTML Layouts Using DIV </title>

      <style>

        .header {

            font-family: sans-serif;

            font-size: 14px;

            background-color: #b5dcb3;

            width: 100%;

            padding: 5px 0px;

        }

        .nav {

            background-color: #aaa;

            height: 200px;

            width: 20%;

            float: left;

        }

        .body {

            background-color: #eee;

            height: 200px;

            width: 80%;

            float: left;

        }

        .footer {

            background-color: #b5dcb3;

            clear: both;

        }

      </style>

   </head>

   <body>

      <div style = "width:100%">

         <div class = "header">

            <h1>This is Web Page Main title</h1>

         </div>

         <div class = "nav">

            <div><b>Main Menu</b></div>

            HTML<br />

            PHP<br />

            PERL...

         </div>

         <div class = "body" >

            <p>Technical and Managerial Tutorials</p>

         </div>

         <div class = "footer">

            <center>

               Copyright © 2007 Tutorialspoint.com

            </center>

         </div>

      </div>

   </body>

</html>

Output:



**Warning**

In addition to using <div> tag for creating layouts in HTML, we can use table (which is actually the simplest way), but it's NOT recommended. Layouts created using tables are rendered very slowly. Tables should only be used to display tabular data.

<https://www.tutorialrepublic.com/html-tutorial/html-layout.php>

# JavaScript in HTML

## What Is Script in HTML?

A script is a small piece of program that can add interactivity to your website. You can write various small functions (called event handlers) using a script and then trigger those functions using HTML attributes. For example, a script could generate a pop-up alert box message when a button is clicked, or provide a dropdown menu.

Nowadays, only JavaScript and associated frameworks are being used by most of the web developers.

You can keep JavaScript code in a separate file and then include it wherever it's needed, or you can define functionality inside HTML document itself. Let's explore both ways.

## Different Ways to Use JavaScript in HTML

### External JavaScript

If you are going to define a functionality which will be used in various HTML documents, then it's always recommended to keep that functionality in a separate JavaScript file and then include that file in your HTML documents.

A JavaScript file will have extension as .js and it will be included in HTML files using <script> tag.

Consider we define a small function using JavaScript in script.js which has following code:

function ShowGreeting() {

    alert("Hello, World");

}

Now let's make use of the above external JavaScript file in the following HTML document:

<!DOCTYPE html>

<html>

   <head>

      <title>JavaScript External Script</title>

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

   </head>

   <body>

      <input type = "button" onclick = "ShowGreeting();" name="ok" value="Click Me" />

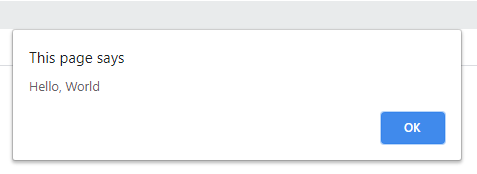
   </body>

</html>

Output:



When you click on this button, an alert message box is displayed:



### Internal Script

You can re-write the above example by inserting the JavaScript script code directly into your HTML document. There is no restriction where you can put your source code in the HTML document, but usually we keep it in header of the document using <script> tag.

<!DOCTYPE html>

<html>

   <head>

      <title>JavaScript Internal Script</title>

      <script type = "text/JavaScript">

         function ShowGreeting() {

            alert("Hello, World");

         }

      </script>

   </head>

   <body>

      <input type = "button" onclick = "ShowGreeting();" name="ok" value="Click Me" />

   </body>

</html>

## How to Choose Default Scripting Language

There may be a situation when you will include multiple script files and ultimately using multiple <script> tags. You can specify a default scripting language for all your script tags. This saves you from specifying the language every time you use a script tag within the page.

Below is the example:

<meta http-equiv = "Content-Script-Type" content = "text/JavaScript" />