**What is Markup Language**

A markup language is a tag-based system that produces formatted, annotated, and human-readable results. A markup language is a set of rules that defines how the layout and presentation of text and images should appear in a digital document. It allows structuring documents, adding formatting, and specifying how different elements should be displayed (or “rendered”) on webpages.

HTML, XML, and LaTeX are among the most popular markup languages — but markup languages have existed for quite a while in many forms.

*Markup Language vs. programming language vs scripting langugage*

Markup languages are used to structure and format text and other content on a webpage. Programming languages are used to create software applications and operating systems. Markup are ***presentational*** languages and it doesn’t include any kind of logic or algorithm.

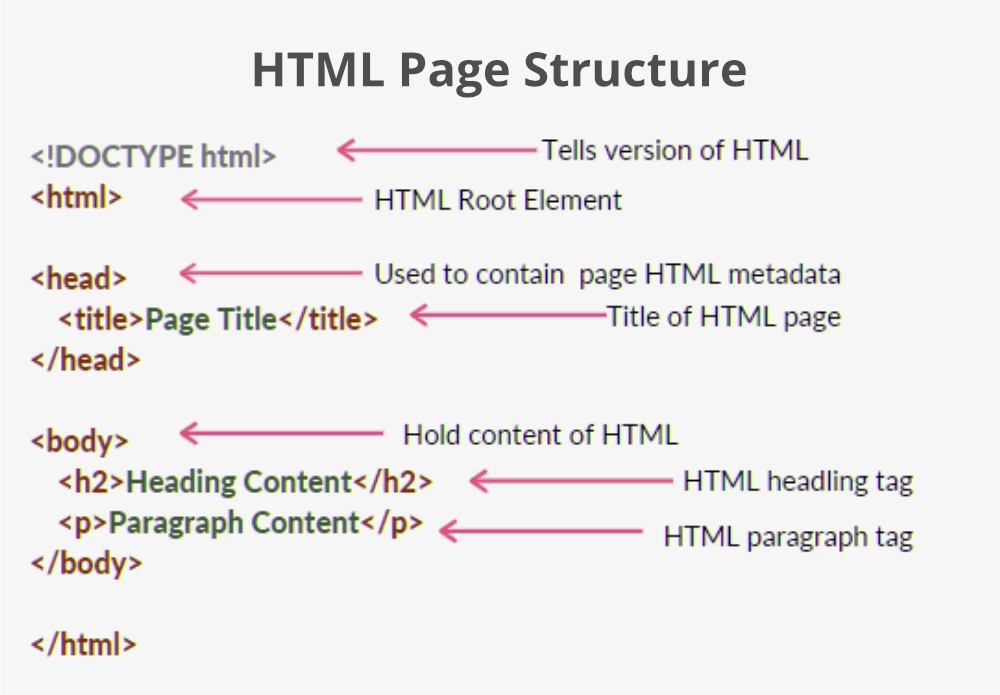
**Scripting Language:**As the name suggest, it’s all about giving the script to perform some certain task. Scripting languages are basically the subcategory of programming languages which is used to give guidance to another program or we can say to control another program, so it also involves instructions. It basically connects one language to one another languages and doesn’t work standalone. [Javascript](https://www.geeksforgeeks.org/javascript-tutorial/), [PHP](https://www.geeksforgeeks.org/php/), [Perl](https://www.geeksforgeeks.org/perl-tutorial/), [Python](https://www.geeksforgeeks.org/python-programming-language/), VBScript these all are the examples of scripting language.

HyperText Markup Language (HTML)

HTML stands for **HyperText Markup Language**. It creates a complete website structure of web pages. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between the web pages and markup language defines the text document within the tag that defines the structure of web pages. This HTML Tutorial provides basic to advanced HTML concepts for beginners and professionals.

## Basic Structure of Web Page

The basic structure of an HTML page is given below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) to create web page.



## Basic Tags

[**<DOCTYPE! html>**](https://www.geeksforgeeks.org/html-doctypes/)**–** A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. It is not an element or tag. The doctype declaration is not case-sensitive. For HTML 4: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/strict.dtd">

* [**<html>**](https://www.geeksforgeeks.org/html-html-tag/)**–** This tag is used to define the root element of an HTML document. This tag tells the browser that it is an HTML document. It is the second outer container element that contains all other elements within it.
* [**<head>**](https://www.geeksforgeeks.org/html-head-tag/)**–** This tag is used to define the head portion of the HTML document that contains information related to the document. Elements within the head tag are not visible on the front end of a webpage. It contains information such as the page <title>, links to CSS (if you choose to style your HTML content with CSS), links to custom favicons, and other metadata (data about the HTML, such as the author, and important keywords that describe the document).
* [**<body>**](https://www.geeksforgeeks.org/html-body-tag/)**–** The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.

**HTML History**

Since the early days of the World Wide Web, there have been many versions of HTML:

1989 Tim Berners-Lee invented www

### *The Timline of HTML*

**1991-** Tim Berners-Lee invents HTML 1.0

**1993-** HTML 1.0 is released. Not many developers are creating websites at this time.

**1995-** HTML 2.0 is published. This contains the features of HTML 1.0 plus new features. This remained the standard markup language for designing and creating websites until 1997.

**1997-** HTML 3.0 was invented. Here, Dave Raggett introduced a fresh draft on HTML, which improved new features of HTML and gave more powerful characteristics for webmasters in designing websites. Unfortunately, the powerful features slowed down the browser in applying further improvements.

**1999-** The widely-used HTML 4.0 comes out. It is very successful.

**2014-** HTML 5.0 is released and used worldwide. It is said to be the extended version of HTML 4.01 which was published in 2012.

**HTML Element**

An HTML element is defined by a start tag, some content, and an end tag. The HTML **element** is everything from the start tag to the end tag.

<tagname>Content goes here...</tagname>

**Start tag Element content End tag**

<h1> My First Heading </h1>

<p> My first paragraph. </p>

<br> none none

**HTML Basic Tags:**

**HTML Headings**

HTML headings are defined with the <h1> to <h6> tags.

**HTML Paragraph**

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

## *HTML Display*

You cannot be sure how HTML will be displayed.

Large or small screens, and resized windows will create different results.

With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code.

The browser will automatically remove any extra spaces and lines when the page is displayed:

**Link Break tag: <br>** Its an empty tag

**HTML Horizontal Rules**

The <hr> tag is used to break the page into various parts, creating horizontal margins with help of a horizontal line running from the left to right-hand side of the page. This is also an empty tag and doesn’t take any additional statements.

<h1>This is heading 1</h1>  
<p>This is some text.</p>  
<hr>  
<h2>This is heading 2</h2>  
<p>This is some other text.</p>  
<hr>

**The Poem Problem**

<p>

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

</p>

*Solution - The HTML <pre> Element*

The HTML <pre> element defines preformatted text.

<pre>

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

</pre>

Making text Italic or emphasize: The <i> tag is used to italicise the text. It opens with <i> and ends with </i> tag. The <em> tag is used to emphasize the text, with added semantic importance. It opens with <em> and ends with </em> tag.

Bold Text using <b> tag : <b>Here is bold text </b>

Tag for underline text: <u> Text </u>

Add images in web page: <img> tag

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The <img> tag has two required attributes:

* src - Specifies the path to the image
* alt - Specifies an alternate text for the image

<img src="*url*" alt="alternatetext">

<img src="img\_girl.jpg" alt="Girl in a jacket" width="500" height="600">

Image in other directory: <img src="/images/html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

Image on server:

<img src="https://www.w3schools.com/images/w3schools\_green.jpg" alt="W3Schools.com">

HTML link: <a href="https://nitp.ac.in/cse">Department of Computer Science & Engg</a>

**Hyperlink:**

HTML links are hyperlinks. You can click on a link and jump to another document. Syntax:

<a href="*url*">*link text*</a>

<a href="https://www.nitp.ac.in/cse.html">CSE Dept.</a>

By default, links will appear as follows in all browsers:

* An unvisited link is underlined and blue
* A visited link is underlined and purple
* An active link is underlined and red

By default, the linked page will be displayed in the current browser window.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

* \_self - Default. Opens the document in the same window/tab as it was clicked
* \_blank - Opens the document in a new window or tab
* \_parent - Opens the document in the parent frame
* \_top - Opens the document in the full body of the window

<a href="https://www.w3schools.com/" target="\_blank">Visit W3Schools!</a>

## *Absolute URLs vs. Relative URLs*

Both examples above are using an absolute URL (a full web address) in the href attribute.

A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

<h2>Absolute URLs</h2>  
<p><a href="https://www.w3.org/">W3C</a></p>  
<p><a href="https://www.google.com/">Google</a></p>  
  
<h2>Relative URLs</h2>  
<p><a href="html\_images.asp">HTML Images</a></p>  
<p><a href="/css/default.asp">CSS Tutorial</a></p>

# **HTML Styles**

The HTML style attribute is used to add styles to an element, such as color, font, size, and more.

<tagname style="property:value;">

The ***property*** is a CSS property. The ***value*** is a CSS value.

Eg. Set the background color for a page to powderblue:

<body style="background-color:powderblue;">  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>

The CSS color property defines the text color for an HTML element:

<h1 style="color:blue;">This is a heading</h1>  
<p style="color:red;">This is a paragraph.</p>

The CSS font-family property defines the font to be used for an HTML element:

<h1 style="font-family:verdana;">This is a heading</h1>  
<p style="font-family:courier;">This is a paragraph.</p>

The CSS font-size property defines the text size for an HTML element:

<h1 style="font-size:300%;">This is a heading</h1>  
<p style="font-size:160%;">This is a paragraph.</p>

The CSS text-align property defines the horizontal text alignment for an HTML element

<h1 style="text-align:center;">Centered Heading</h1>  
<p style="text-align:center;">Centered paragraph.</p>

## HTML Formatting Elements

Formatting elements were designed to display special types of text:

* <b> - Bold text
* <strong> - Important text
* <i> - Italic text
* <em> - Emphasized text
* <mark> - Marked text
* <small> - Smaller text
* <del> - Deleted text
* <ins> - Inserted text (underline the inserted text) <p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
* <sub> - Subscript text : <p>This is <sub>subscripted</sub> text.</p>
* <sup> - Superscript text

## HTML Comment Tag

<!-- Write your comments here -->

**HTML Lists:**

## Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

* Coffee
* Tea
* Milk

**Ordered List**

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

<ol>

<li>Content 1</li>

<li>Content 2</li>

<li>Content 3> </li>

</ol>

Example:

1. Content 1

2. Content 2

3. Content 3

By default, list items numbered with numbers

## HTML Description Lists

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

<dl>  
  <dt>Coffee</dt>  
  <dd>- black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>- white cold drink</dd>  
</dl>

**HTML Tables**

HTML tables allow web developers to arrange data into rows and columns. A table in HTML consists of table cells inside rows and columns.

<table>  
  <tr>  
    <th>Roll No</th>  
    <th>Name</th>  
    <th>Marks</th>  
  </tr>  
  <tr>  
    <td>100</td>  
    <td>Rajat</td>  
    <td>78</td>  
  </tr>  
  <tr>  
    <td>101</td>  
    <td>Rakesh</td>  
    <td>88</td>  
  </tr>  
</table>

The <caption> tag must be inserted immediately after the [<table>](https://www.w3schools.com/tags/tag_table.asp) tag. By default, a table caption will be center-aligned above a table. However, the CSS properties [text-align](https://www.w3schools.com/cssref/pr_text_text-align.asp) and [caption-side](https://www.w3schools.com/cssref/pr_tab_caption-side.asp) can be used to align and place the caption.

## How To Add a Border?

## To add a border, use the CSS border property on table, th, and td elements:

<head>

<style>

table, th, td {

border: 1px solid black;

}

</style>

</head>

To avoid having double borders like in the example above, set the CSS border-collapse property to collapse.

table, th, td {  
  border: 1px solid black;  
  border-collapse: collapse;  
}

set a background color of each cell, and give the border a white color

table, th, td {  
  border: 1px solid white;  
  border-collapse: collapse;  
}  
th, td {  
  background-color: #96D4D4;  
}

Dotted table border

th, td {

border-style: dotted;

}

**Table Size**

## HTML Table Width

To set the width of a table, add the style attribute to the <table> element:

<table style="width:100%">

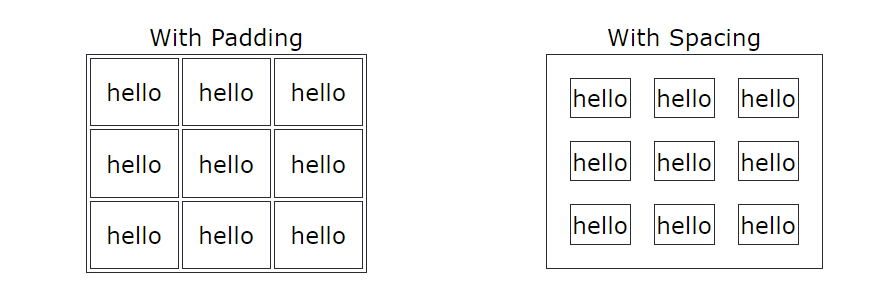
sing a percentage as the size unit for a width means how wide will this element be compared to its parent element, which in this case is the <body> element.

Set the height of the row:

<table style="width:100%">  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
    <th>Age</th>  
  </tr>  
  <tr style="height:200px">  
    <td>Jill</td>  
    <td>Smith</td>  
    <td>50</td>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>  
    <td>94</td>  
  </tr>  
</table>

# **HTML Table Padding & Spacing**

HTML tables can adjust the padding inside the cells, and also the space between the cells.



Cell padding is the space between the cell edges and the cell content.

By default the padding is set to 0.

To add padding on table cells, use the CSS padding property:

th, td {  
  padding: 15px;  
}

th, td {  
  padding-top: 10px;  
  padding-bottom: 20px;  
  padding-left: 30px;  
  padding-right: 40px;  
}

## HTML Table - Cell Spacing

Cell spacing is the space between each cell.

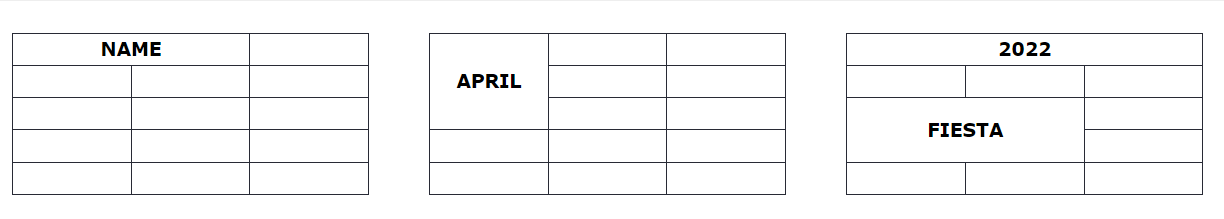
By default the space is set to 2 pixels.

To change the space between table cells, use the CSS border-spacing property on the table element:

table {  
  border-spacing: 30px;  
}

# **HTML Table Colspan & Rowspan**

HTML tables can have cells that span over multiple rows and/or columns.



## HTML Table - Colspan

To make a cell span over multiple columns, use the colspan attribute:

<table>  
  <tr>  
    <th colspan="2">Name</th>  
    <th>Age</th>  
  </tr>  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>  
    <td>43</td>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>  
    <td>57</td>  
  </tr>  
</table>

**Note:** The value of the colspan attribute represents the number of columns to span.

## HTML Table - Rowspan

To make a cell span over multiple rows, use the rowspan attribute:

<table>  
  <tr>  
    <th>Name</th>  
    <td>Jill</td>  
  </tr>  
  <tr>  
    <th rowspan="2">Phone</th>  
    <td>555-1234</td>  
  </tr>  
  <tr>  
    <td>555-8745</td>  
</tr>  
</table>

**Note:** The value of the rowspan attribute represents the number of rows to span.

**Div Tag**

The <div> tag defines a division or a section in an HTML document.

The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The <div> tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the <div> tag!

**Note:** By default, browsers always place a line break before and after the <div> element.

Example:

Lorem Ipsum <div>I am a div</div> dolor sit amet.

*In Browser:*

Lorem Ipsum

I am a div

dolor sit amet.

div {

background-color: #FFF4A3;

}

<div>

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

## London

London is the capital city of England.

London has over 13 million inhabitants.

## Center align a <div> element

## If you have a <div> element that is not 100% wide, and you want to center-align it, set the CSS margin property to auto.

## <style> div {   width:300px;   margin:auto; } </style>

## Multiple <div> elements

<div>  
  <h2>London</h2>  
  <p>London is the capital city of England.</p>  
  <p>London has over 13 million inhabitants.</p>  
</div>  
  
<div>  
  <h2>Oslo</h2>  
  <p>Oslo is the capital city of Norway.</p>  
  <p>Oslo has over 600.000 inhabitants.</p>  
</div>  
  
<div>  
  <h2>Rome</h2>  
  <p>Rome is the capital city of Italy.</p>  
  <p>Rome has almost 3 million inhabitants.</p>  
</div>

## London

London is the capital city of England.

London has over 13 million inhabitants.

## Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

## Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

## Aligning <div> elements side by side

**Using float:**

The CSS float property is used for positioning and formatting content and allow elements float next to each other instead of on top of each other.

<style>

div.mycontainer {

width:100%;

overflow:auto;

}

div.mycontainer div {

width:33%;

float:left;

}

</style>

<body>

<div class="mycontainer">

<div style="background-color:#FFF4A3;">

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

<div style="background-color:#FFC0C7;">

<h2>Oslo</h2>

<p>Oslo is the capital city of Norway.</p>

<p>Oslo has over 600.000 inhabitants.</p>

</div>

<div style="background-color:#D9EEE1;">

<h2>Rome</h2>

<p>Rome is the capital city of Italy.</p>

<p>Rome has almost 3 million inhabitants.</p>

</div>

</div>

</body>



## Inline-block

If you change the <div> element's display property from block to inline-block, the <div> elements will no longer add a line break before and after, and will be displayed side by side instead of on top of each other

<style>

div {

width:30%;

display:inline-block;

}

</style>

<body>

<div style="background-color:#FFF4A3;">

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

<div style="background-color:#FFC0C7;">

<h2>Oslo</h2>

<p>Oslo is the capital city of Norway.</p>

<p>Oslo has over 600.000 inhabitants.</p>

</div>

<div style="background-color:#D9EEE1;">

<h2>Rome</h2>

<p>Rome is the capital city of Italy.</p>

<p>Rome has almost 3 million inhabitants.</p>

</div>

</body>



## Using Flex

The CSS Flexbox Layout Module was introduced to make it easier to design flexible responsive layout structure without using float or positioning.

To make the CSS flex method work, surround the <div> elements with another <div> element and give it the status as a flex container.

<head>

<style>

div.mycontainer {

display: flex;

}

div.mycontainer div {

width:33%;

}

</style>

</head>

<body>

<h1>Flexbox Example</h1>

<p>Align three DIV elements side by side.</p>

<div class="mycontainer">

<div style="background-color:#FFF4A3;">

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

<div style="background-color:#FFC0C7;">

<h2>Oslo</h2>

<p>Oslo is the capital city of Norway.</p>

<p>Oslo has over 600.000 inhabitants.</p>

</div>

<div style="background-color:#D9EEE1;">

<h2>Rome</h2>

<p>Rome is the capital city of Italy.</p>

<p>Rome has almost 3 million.</p>

</div>

</div>

</body>

## Using Grid

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

Sounds almost the same as flex, but has the ability to define more than one row and position each row individually.

The CSS grid method requires that you surround the <div> elements with another <div> element and give the status as a grid container, and you must specify the width of each column.

<style>  
div.grid-container {  
  display: grid;  
  grid-template-columns: 33% 33% 33%;  
}  
</style>

**HTML Forms**

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

## The <form> Element

## The HTML <form> element is used to create an HTML form for user input:

<form>  
.  
*form elements*  
.  
</form>

The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

## The <input> Element

The HTML <input> element is the most used form element.

An <input> element can be displayed in many ways, depending on the type attribute.

|  |  |
| --- | --- |
| **Type** | **Description** |
| <input type="text"> | Displays a single-line text input field |
| <input type="radio"> | Displays a radio button (for selecting one of many choices) |
| <input type="checkbox"> | Displays a checkbox (for selecting zero or more of many choices) |
| <input type="submit"> | Displays a submit button (for submitting the form) |
| <input type="button"> | Displays a clickable button |

## Text Fields

The <input type="text"> defines a single-line input field for text input.

<body>

<h2>Text input fields</h2>

<form>

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe">

</form>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of text input fields is 20 characters.</p>

</body>

First name:  
  
Last name:

Top of Form



Bottom of Form

## Input Type Email

The <input type="email"> is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

## Input Type Image

The <input type="image"> defines an image as a submit button.

The path to the image is specified in the src attribute.

## <form> <input type="image" src="img\_submit.gif" alt="Submit" width="48" height="48"> </form>

## Input Type File

The <input type="file"> defines a file-select field and a "Browse" button for file uploads.

## <form>   <label for="myfile">Select a file:</label>   <input type="file" id="myfile" name="myfile"> </form>

## Input Type Hidden

The <input type="hidden"> defines a hidden input field (not visible to a user).

A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted.

A hidden field often stores what database record that needs to be updated when the form is submitted.

## <form>   <label for="fname">First name:</label>   <input type="text" id="fname" name="fname"><br><br>   <input type="hidden" id="custId" name="custId" value="3487">   <input type="submit" value="Submit"> </form>

#### INPUT type="number" and type="range"

## The number and range input types also accept parameters for min, max and step. In most cases you can leave out step as it defaults to 1.

Here you see an example including both a number input, typically displayed as a 'roller' and a range input displayed as a 'slider':

Age: <input type="number" size="6" name="age" min="18" max="99" value="21"><br>

## Satisfaction: <input type="range" size="2" name="satisfaction" min="1" max="5" value="3">

## HTML5 number and range example

## The <label> Element

The <label> tag defines a label for many form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.

The <label> element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

## Radio Buttons

The <input type="radio"> defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

<form>  
  <input type="radio" id="html" name="fav\_language" value="HTML">  
  <label for="html">HTML</label><br>  
  <input type="radio" id="css" name="fav\_language" value="CSS">  
  <label for="css">CSS</label><br>  
  <input type="radio" id="javascript" name="fav\_language" value="JavaScript">  
  <label for="javascript">JavaScript</label>  
</form>

Choose your favorite Web language:

 HTML  
 CSS  
 JavaScript

## Checkboxes

The <input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

<form>  
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">  
  <label for="vehicle1"> I have a bike</label><br>  
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">  
  <label for="vehicle2"> I have a car</label><br>  
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">  
  <label for="vehicle3"> I have a boat</label>  
</form>

This is how the HTML code above will be displayed in a browser:

 I have a bike  
 I have a car  
 I have a boat

## The <select> Element

The <select> element defines a drop-down list:

<label for="cars">Choose a car:</label>  
<select id="cars" name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

Top of Form

Choose a car:                    

Bottom of Form

The <option> element defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the selected attribute to the option:

<option value="fiat" selected>Fiat</option>

### **Visible Values:**

Use the size attribute to specify the number of visible values:

<select id="cars" name="cars" size="3">

### **Allow Multiple Selections:**

<label for="cars">Choose a car:</label>  
<select id="cars" name="cars" size="4"multiple>

## The Submit Button

The <input type="submit"> defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's action attribute.

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

First name:  
  
Last name:  
  
  


## The Name Attribute for <input>

Notice that each input field must have a name attribute to be submitted.

If the name attribute is omitted, the value of the input field will not be sent at all.

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" value="John"><br><br>  
  <input type="submit" value="Submit">  
</form>

# **HTML Form Attributes**

The action attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

In the example below, the form data is sent to a file called "action\_page.php". This file contains a server-side script that handles the form data:

<form action="/action\_page.php">

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

<input type="submit" value="Submit">

</form>

## The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

|  |  |
| --- | --- |
| \_blank | The response is displayed in a new window or tab |
| \_self | The response is displayed in the current window |
| \_parent | The response is displayed in the parent frame |
| \_top | The response is displayed in the full body of the window |
|  |  |

## The Method Attribute

The method attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

<form action="/action\_page.php" method="get">

**Notes on GET:**

* Appends the form data to the URL, in name/value pairs
* NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
* The length of a URL is limited (2048 characters)
* Useful for form submissions where a user wants to bookmark the result
* GET is good for non-secure data, like query strings in Google

**Notes on POST:**

* Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
* POST has no size limitations, and can be used to send large amounts of data.
* Form submissions with POST cannot be bookmarked

**Tip:** Always use POST if the form data contains sensitive or personal information

## The Autocomplete Attribute

The autocomplete attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

<form action="/action\_page.php" autocomplete="on">

## The Novalidate Attribute

The novalidate attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

<form action="/action\_page.php" novalidate>

# **HTML Input Attributes**

## The value Attribute

The input value attribute specifies an initial value for an input field:

## The readonly Attribute

The input readonly attribute specifies that an input field is read-only.

A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it).

The value of a read-only input field will be sent when submitting the form!

## <form>   <label for="fname">First name:</label><br>   <input type="text" id="fname" name="fname" value="John" readonly><br>   <label for="lname">Last name:</label><br>   <input type="text" id="lname" name="lname" value="Doe"> </form>

## The disabled Attribute

The input disabled attribute specifies that an input field should be disabled.

A disabled input field is unusable and un-clickable.

The value of a disabled input field will not be sent when submitting the form!

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John" disabled><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe">  
</form>

## The size Attribute

The input size attribute specifies the visible width, in characters, of an input field.

The default value for size is 20.

**Note:** The size attribute works with the following input types: text, search, tel, url, email, and password.

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" size="50"><br>  
  <label for="pin">PIN:</label><br>  
  <input type="text" id="pin" name="pin" size="4">  
</form>

## The maxlength Attribute

The input maxlength attribute specifies the maximum number of characters allowed in an input field.

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" size="50"><br>  
  <label for="pin">PIN:</label><br>  
  <input type="text" id="pin" name="pin" maxlength="4" size="4">  
</form>

## The min and max Attributes

The input min and max attributes specify the minimum and maximum values for an input field.

The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

**Tip:** Use the max and min attributes together to create a range of legal values.

<form>  
  <label for="datemax">Enter a date before 1980-01-01:</label>  
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>  
  
  <label for="datemin">Enter a date after 2000-01-01:</label>  
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>  
  
  <label for="quantity">Quantity (between 1 and 5):</label>  
  <input type="number" id="quantity" name="quantity" min="1" max="5">  
</form>

## The multiple Attribute

The input multiple attribute specifies that the user is allowed to enter more than one value in an input field.

The multiple attribute works with the following input types: email, and file.

<form>  
  <label for="files">Select files:</label>  
  <input type="file" id="files" name="files" multiple>  
</form>

## The pattern Attribute

The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.

The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

## The placeholder Attribute

The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).

The short hint is displayed in the input field before the user enters a value.

The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

<form action="/action\_page.php">

<label for="phone">Enter a phone number:</label>

<input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}"><br><br>

<input type="submit" value="Submit">

</form>

## The list Attribute

The input list attribute refers to a <datalist> element that contains pre-defined options for an <input> element.Top of FormBottom of Form

<form>  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Edge">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>

## <Fieldset> element

## Definition and Usage

The <fieldset> tag is used to group related elements in a form.

The <fieldset> tag draws a box around the related elements.

<form action="/action\_page.php">

<fieldset>

<legend>Personalia:</legend>

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

<input type="text" id="fname" name="fname"><br><br>

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

<input type="text" id="lname" name="lname"><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email"><br><br>

<label for="birthday">Birthday:</label>

<input type="date" id="birthday" name="birthday"><br><br>

<input type="submit" value="Submit">

</fieldset>

</form>

**HTML Graphics**

**SVG Graphics**

## What is SVG?

* SVG stands for Scalable Vector Graphics
* SVG is used to define graphics for the Web
* SVG is a W3C recommendation

The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

## SVG Circle

<!DOCTYPE html>  
<html>  
<body>  
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>  
  
</body>  
</html>

## SVG Rectangle

<svg width="400" height="100">  
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />  
</svg>

## SVG Rounded Rectangle

<svg width="400" height="180">  
  <rect x="50" y="20" rx="20" ry="30" width="150" height="150"  
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />  
</svg>

**x** − x-axis co-ordinate of top left of the rectangle. Default is 0.

**y** − y-axis co-ordinate of top left of the rectangle. Default is 0.

rx: The horizontal corner radius of the rect. used to round the corner of the rounded rectangle.

[ry](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/ry): The vertical corner radius of the rect.

**Ellipse**

<svg height="120" width=“400">

<ellipse cx="200" cy="80" rx=“200" ry=“100"

style="fill:yellow;stroke:purple;stroke-width:2" />

</svg>

**Line**

<svg height="200" width=“400">

<line x1="0" y1="0" x2="200" y2="200" style="stroke:red;stro

ke-width:2" />

</svg>

## Text

## <svg height=“50" width=“80">

## <text x="0" y="10" fill=“blue">NITP</text>

## </svg>

## <svg height="800" width="700">

## <text x="20" y="40" style="fill:blue;">Lines:

## <tspan x="20" y="55">Line number

## one</tspan>

## <tspan x="20" y="90">Line number two

## </tspan>

## </text>

## </svg>

## <tspan>: defines text on several lines.

## SVG Star

## <svg width="300" height="200">

## <polygon points="100,10 40,198 190,78 10,78 160,198"

## style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />

## Sorry, your browser does not support inline SVG.

## SVG Logo

## <svg height="130" width="500">

## <defs>

## <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">

## <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />

## <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />

## </linearGradient>

## </defs>

## <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />

## <text fill="#ffffff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>

## Sorry, your browser does not support inline SVG.

# HTML Multimedia

## Multimedia on the web is sound, music, videos, movies, and animations.

## Multimedia Formats

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

## The MP4, WebM, and Ogg formats are supported by HTML.

## Only MP3, WAV, and Ogg audio are supported by the HTML standard.

**HTML <video> element**

<video width="320" height="240" controls>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

## How it Works

The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.

The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

## HTML <video> Autoplay

To start a video automatically, use the autoplay attribute:

<video width="320" height="240" autoplay>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

**Mute the video:**

<video width="320" height="240" autoplay muted>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

## HTML <audio> Element

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

## HTML Audio - How It Works

The controls attribute adds audio controls, like play, pause, and volume.

The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

## HTML <audio> Autoplay

To start an audio file automatically, use the autoplay attribute:

<audio controls autoplay>

<audio controls autoplay muted>