HTML is the standard markup language for creating Web pages.

What is HTML?

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

An HTML element is defined by a start tag, some content, and an end tag.

HTML Attributes

* All HTML elements can have **attributes**
* Attributes provide **additional information** about elements
* Attributes are always specified in **the start tag**
* Attributes usually come in name/value pairs like: **name="value"**
* <b> - Bold text
* <strong> - Important text
* <i> - Italic text
* <em> - Emphasized text
* <mark> - Marked text
* <small> - Smaller text
* <del> - Deleted text
* <ins> - Inserted text
* <sub> - Subscript text
* <sup> - Superscript text

Using CSS

CSS can be added to HTML documents in 3 ways:

* **Inline** - by using the style attribute inside HTML elements
* **Internal** - by using a <style> element in the <head> section
* **External** - by using a <link> element to link to an external CSS file

<h1 style="color:blue;">A Blue Heading</h1>  
  
<p style="color:red;">A red paragraph.</p>

<head>  
<style>  
body {background-color: powderblue;}  
h1   {color: blue;}  
p    {color: red;}  
</style>  
</head>

<head>  
  <link rel="stylesheet" href="styles.css">  
</head>

The CSS padding property defines a padding (space) between the text and the border.

The CSS margin property defines a margin (space) outside the border.

<a href="*url*">*link text*</a>

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

HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

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>

The HTML <img> tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

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="img\_chania.jpg" alt="Flowers in Chania">

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

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

|  |  |  |
| --- | --- | --- |
| **Abbreviation** | **File Format** | **File Extension** |
| APNG | Animated Portable Network Graphics | .apng |
| GIF | Graphics Interchange Format | .gif |
| ICO | Microsoft Icon | .ico, .cur |
| JPEG | Joint Photographic Expert Group image | .jpg, .jpeg, .jfif, .pjpeg, .pjp |
| PNG | Portable Network Graphics | .png |
| SVG | Scalable Vector Graphics | .svg |

<img src="workplace.jpg" alt="Workplace" usemap="#workmap">  
  
<map name="workmap">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">  
</map>A clickable area is defined using an <area> element.

### **Shape**

You must define the shape of the clickable area, and you can choose one of these values:

* rect - defines a rectangular region
* circle - defines a circular region
* poly - defines a polygonal region
* default - defines the entire region

<map name="workmap">  
  <area shape="circle" coords="337,300,44" href="coffee.htm" onclick="myFunction()">  
</map>  
  
<script>  
function myFunction() {  
  alert("You clicked the coffee cup!");  
}  
</script>

<picture>  
  <source media="(min-width: 650px)" srcset="img\_food.jpg">  
  <source media="(min-width: 465px)" srcset="img\_car.jpg">  
  <img src="img\_girl.jpg">  
</picture>

A favicon is a small image displayed next to the page title in the browser tab.

<head>  
  <title>My Page Title</title>  
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">  
</head>

Dotted Table Borders

With the border-style property, you can set the appearance of the border.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

The following values are allowed:

* dotted
* dashed
* solid
* double
* groove
* ridge
* inset
* outset
* none
* hidden

<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>

<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>

## HTML Table - Zebra Stripes

tr:nth-child(even) {  
  background-color: #D6EEEE;  
}

vertical zebra stripes, style every other column, instead of every other row.

td:nth-child(even), th:nth-child(even) {  
  background-color: #D6EEEE;  
}

You can combine the styling from the two examples above and you will have stripes on every other row and every other column.

If you use a transparent color you will get an overlapping effect.

tr:nth-child(even) {  
  background-color: rgba(150, 212, 212, 0.4);  
}  
  
th:nth-child(even),td:nth-child(even) {  
  background-color: rgba(150, 212, 212, 0.4);  
}

tr:hover {background-color: #D6EEEE;}

## Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Two commonly used block elements are: <p> and <div>.

The <p> element defines a paragraph in an HTML document.

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

Here are the block-level elements in HTML:

[<address>](https://www.w3schools.com/tags/tag_address.asp)

[<article>](https://www.w3schools.com/tags/tag_article.asp)

[<aside>](https://www.w3schools.com/tags/tag_aside.asp)

[<blockquote>](https://www.w3schools.com/tags/tag_blockquote.asp)

[<canvas>](https://www.w3schools.com/tags/tag_canvas.asp)

[<dd>](https://www.w3schools.com/tags/tag_dd.asp)

[<div>](https://www.w3schools.com/tags/tag_div.asp)

[<dl>](https://www.w3schools.com/tags/tag_dl.asp)

[<dt>](https://www.w3schools.com/tags/tag_dt.asp)

[<fieldset>](https://www.w3schools.com/tags/tag_fieldset.asp)

[<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp)

[<figure>](https://www.w3schools.com/tags/tag_figure.asp)

[<footer>](https://www.w3schools.com/tags/tag_footer.asp)

[<form>](https://www.w3schools.com/tags/tag_form.asp)

[<h1>-<h6>](https://www.w3schools.com/tags/tag_hn.asp)

[<header>](https://www.w3schools.com/tags/tag_header.asp)

[<hr>](https://www.w3schools.com/tags/tag_hr.asp)

[<li>](https://www.w3schools.com/tags/tag_li.asp)

[<main>](https://www.w3schools.com/tags/tag_main.asp)

[<nav>](https://www.w3schools.com/tags/tag_nav.asp)

[<noscript>](https://www.w3schools.com/tags/tag_noscript.asp)

[<ol>](https://www.w3schools.com/tags/tag_ol.asp)

[<p>](https://www.w3schools.com/tags/tag_p.asp)

[<pre>](https://www.w3schools.com/tags/tag_pre.asp)

[<section>](https://www.w3schools.com/tags/tag_section.asp)

[<table>](https://www.w3schools.com/tags/tag_table.asp)

[<tfoot>](https://www.w3schools.com/tags/tag_tfoot.asp)

[<ul>](https://www.w3schools.com/tags/tag_ul.asp)

[<video>](https://www.w3schools.com/tags/tag_video.asp)

Here are the inline elements in HTML:

[<a>](https://www.w3schools.com/tags/tag_a.asp)

[<abbr>](https://www.w3schools.com/tags/tag_abbr.asp)

[<acronym>](https://www.w3schools.com/tags/tag_acronym.asp)

[<b>](https://www.w3schools.com/tags/tag_b.asp)

[<bdo>](https://www.w3schools.com/tags/tag_bdo.asp)

[<big>](https://www.w3schools.com/tags/tag_big.asp)

[<br>](https://www.w3schools.com/tags/tag_br.asp)

[<button>](https://www.w3schools.com/tags/tag_button.asp)

[<cite>](https://www.w3schools.com/tags/tag_cite.asp)

[<code>](https://www.w3schools.com/tags/tag_code.asp)

[<dfn>](https://www.w3schools.com/tags/tag_dfn.asp)

[<em>](https://www.w3schools.com/tags/tag_em.asp)

[<i>](https://www.w3schools.com/tags/tag_i.asp)

[<img>](https://www.w3schools.com/tags/tag_img.asp)

[<input>](https://www.w3schools.com/tags/tag_input.asp)

[<kbd>](https://www.w3schools.com/tags/tag_kbd.asp)

[<label>](https://www.w3schools.com/tags/tag_label.asp)

[<map>](https://www.w3schools.com/tags/tag_map.asp)

[<object>](https://www.w3schools.com/tags/tag_object.asp)

[<output>](https://www.w3schools.com/tags/tag_output.asp)

[<q>](https://www.w3schools.com/tags/tag_q.asp)

[<samp>](https://www.w3schools.com/tags/tag_samp.asp)

[<script>](https://www.w3schools.com/tags/tag_script.asp)

[<select>](https://www.w3schools.com/tags/tag_select.asp)

[<small>](https://www.w3schools.com/tags/tag_small.asp)

[<span>](https://www.w3schools.com/tags/tag_span.asp)

[<strong>](https://www.w3schools.com/tags/tag_strong.asp)

[<sub>](https://www.w3schools.com/tags/tag_sub.asp)

[<sup>](https://www.w3schools.com/tags/tag_sup.asp)

[<textarea>](https://www.w3schools.com/tags/tag_textarea.asp)

[<time>](https://www.w3schools.com/tags/tag_time.asp)

[<tt>](https://www.w3schools.com/tags/tag_tt.asp)

[<var>](https://www.w3schools.com/tags/tag_var.asp)

The <div> element is often used as a container for other HTML elements.

The <div> element has no required attributes, but style, class and id are common.

The <span> element is an inline container used to mark up a part of a text, or a part of a document.

The <span> element has no required attributes, but style, class and id are common.

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

<iframe src="demo\_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>

The HTML <script> tag is used to define a client-side script (JavaScript).

The <script> element either contains script statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript most often uses the document.getElementById() method.

The <title> element defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The <title> element is required in HTML documents!

The content of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

* defines a title in the browser toolbar
* provides a title for the page when it is added to favorites
* displays a title for the page in search engine-results

So, try to make the title as accurate and meaningful as possible!

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

HTML Layout Elements

HTML has several semantic elements that define the different parts of a web page:

|  |  |
| --- | --- |
| HTML5 Semantic Elements | * <header> - Defines a header for a document or a section * <nav> - Defines a set of navigation links * <section> - Defines a section in a document * <article> - Defines an independent, self-contained content * <aside> - Defines content aside from the content (like a sidebar) * <footer> - Defines a footer for a document or a section * <details> - Defines additional details that the user can open and close on demand * <summary> - Defines a heading for the <details> element |

There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

* CSS framework
* CSS float property
* CSS flexbox
* CSS grid

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

Media Queries

In addition to resize text and images, it is also common to use media queries in responsive web pages.

With media queries you can define completely different styles for different browser sizes.

Example: resize the browser window to see that the three div elements below will display horizontally on large screens and stack vertically on small screens:

<style>  
.left, .right {  
  float: left;  
  width: 20%; /\* The width is 20%, by default \*/  
}  
  
.main {  
  float: left;  
  width: 60%; /\* The width is 60%, by default \*/  
}  
  
/\* Use a media query to add a breakpoint at 800px: \*/  
@media screen and (max-width: 800px) {  
  .left, .main, .right {  
    width: 100%; /\* The width is 100%, when the viewport is 800px or smaller \*/  
  }  
}  
</style>

<!DOCTYPE html>  
<html>  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">  
<body>  
  
<div class="w3-container w3-green">  
  <h1>W3Schools Demo</h1>  
  <p>Resize this responsive page!</p>  
</div>  
  
<div class="w3-row-padding">  
  <div class="w3-third">  
    <h2>London</h2>  
    <p>London is the capital city of England.</p>  
    <p>It is the most populous city in the United Kingdom,  
    with a metropolitan area of over 13 million inhabitants.</p>  
  </div>  
  
  <div class="w3-third">  
    <h2>Paris</h2>  
    <p>Paris is the capital of France.</p>  
    <p>The Paris area is one of the largest population centers in Europe,  
    with more than 12 million inhabitants.</p>  
  </div>  
  
  <div class="w3-third">  
    <h2>Tokyo</h2>  
    <p>Tokyo is the capital of Japan.</p>  
    <p>It is the center of the Greater Tokyo Area,  
    and the most populous metropolitan area in the world.</p>  
  </div>  
</div>  
  
</body>  
</html>

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: <div> and <span> - Tells nothing about its content.

Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

* <article>
* <aside>
* <details>
* <figcaption>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>



HTML <section> Element

The <section> element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a <section> element can be used:

* Chapters
* Introduction
* News items
* Contact information

A web page could normally be split into sections for introduction, content, and contact information.

<html>  
<head>  
<style>  
.all-browsers {  
  margin: 0;  
  padding: 5px;  
  background-color: lightgray;  
}  
  
.all-browsers > h1, .browser {  
  margin: 10px;  
  padding: 5px;  
}  
  
.browser {  
  background: white;  
}  
  
.browser > h2, p {  
  margin: 4px;  
  font-size: 90%;  
}  
</style>  
</head>  
<body>  
  
<article class="all-browsers">  
  <h1>Most Popular Browsers</h1>  
  <article class="browser">  
    <h2>Google Chrome</h2>  
    <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!</p>  
  </article>  
  <article class="browser">  
    <h2>Mozilla Firefox</h2>  
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>  
  </article>  
  <article class="browser">  
    <h2>Microsoft Edge</h2>  
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>  
  </article>  
</article>  
  
</body>  
</html>

<html>  
<head>  
<style>  
aside {  
  width: 30%;  
  padding-left: 15px;  
  margin-left: 15px;  
  float: right;  
  font-style: italic;  
  background-color: lightgray;  
}  
</style>  
</head>  
<body>  
  
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>  
  
<aside>  
<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>  
</aside>  
  
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>  
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>  
  
</body>  
</html>

scheme://prefix.domain:port/path/filename

Explanation:

* **scheme** - defines the **type** of Internet service (most common is **http or https**)
* **prefix** - defines a domain **prefix** (default for http is **www**)
* **domain** - defines the Internet **domain name**(like w3schools.com)
* **port** - defines the **port number**at the host (default for http is **80**)
* **path** - defines a **path** at the server (If omitted: the root directory of the site)
* **filename** - defines the name of a document or resource

<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>

**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

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

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

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

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

The HTML <form> element can contain one or more of the following form elements:

* <input>
* <label>
* <select>
* <textarea>
* <button>
* <fieldset>
* <legend>
* <datalist>
* <output>
* <option>
* <optgroup>

<form action="/action\_page.php">  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>

Here are the different input types you can use in HTML:

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">

## Input Type Text

<input type="text"> defines a **single-line text input field**:

### **Example**

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

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_text)

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

First name:  
  
Last name:  


## Input Type Password

<input type="password"> defines a **password field**:

### **Example**

<form>  
  <label for="username">Username:</label><br>  
  <input type="text" id="username" name="username"><br>  
  <label for="pwd">Password:</label><br>  
  <input type="password" id="pwd" name="pwd">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_password)

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

Username:  
  
Password:  


The characters in a password field are masked (shown as asterisks or circles).

ADVERTISEMENT

## Input Type Submit

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

The form-handler is typically a server page with a script for processing input data.

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

### **Example**

<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_submit)

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

Top of Form

First name:  
  
Last name:  
  
  


Bottom of Form

If you omit the submit button's value attribute, the button will get a default text:

### **Example**

<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">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_submit_nn)

## Input Type Reset

<input type="reset"> defines a **reset button** that will reset all form values to their default values:

### **Example**

<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">  
  <input type="reset">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_reset)

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

Top of Form

First name:  
  
Last name:  
  
  
 

Bottom of Form

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

## Input Type Radio

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

Radio buttons let a user select ONLY ONE of a limited number of choices:

### **Example**

<p>Choose your favorite Web language:</p>  
  
<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_radio)

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

 HTML  
 CSS  
 JavaScript

## Input Type Checkbox

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

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

### **Example**

<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_checkbox2)

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

## Input Type Button

<input type="button"> defines a **button**:

### **Example**

<input type="button" onclick="alert('Hello World!')" value="Click Me!">

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_button)

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

## Input Type Color

The <input type="color"> is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

### **Example**

<form>  
  <label for="favcolor">Select your favorite color:</label>  
  <input type="color" id="favcolor" name="favcolor">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_color)

## Input Type Date

The <input type="date"> is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

### **Example**

<form>  
  <label for="birthday">Birthday:</label>  
  <input type="date" id="birthday" name="birthday">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_date)

You can also use the min and max attributes to add restrictions to dates:

### **Example**

<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">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_date_max_min)

## Input Type Datetime-local

The <input type="datetime-local"> specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

### **Example**

<form>  
  <label for="birthdaytime">Birthday (date and time):</label>  
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_datetime-local)

## 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.

### **Example**

<form>  
  <label for="email">Enter your email:</label>  
  <input type="email" id="email" name="email">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_email)

## 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.

### **Example**

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

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_image)

## Input Type File

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

### **Example**

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

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_file)

## 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.

**Note:** While the value is not displayed to the user in the page's content, it is visible (and can be edited) using any browser's developer tools or "View Source" functionality. Do not use hidden inputs as a form of security!

### **Example**

<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_hidden)

## Input Type Month

The <input type="month"> allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

### **Example**

<form>  
  <label for="bdaymonth">Birthday (month and year):</label>  
  <input type="month" id="bdaymonth" name="bdaymonth">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_month)

## Input Type Number

The <input type="number"> defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

### **Example**

<form>  
  <label for="quantity">Quantity (between 1 and 5):</label>  
  <input type="number" id="quantity" name="quantity" min="1" max="5">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_number)

## Input Restrictions

Here is a list of some common input restrictions:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| checked | Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio") |
| disabled | Specifies that an input field should be disabled |
| max | Specifies the maximum value for an input field |
| maxlength | Specifies the maximum number of character for an input field |
| min | Specifies the minimum value for an input field |
| pattern | Specifies a regular expression to check the input value against |
| readonly | Specifies that an input field is read only (cannot be changed) |
| required | Specifies that an input field is required (must be filled out) |
| size | Specifies the width (in characters) of an input field |
| step | Specifies the legal number intervals for an input field |
| value | Specifies the default value for an input field |

You will learn more about input restrictions in the next chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

### **Example**

<form>  
  <label for="quantity">Quantity:</label>  
  <input type="number" id="quantity" name="quantity" min="0" max="100" step="10" value="30">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_number_step)

## Input Type Range

The <input type="range"> defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

### **Example**

<form>  
  <label for="vol">Volume (between 0 and 50):</label>  
  <input type="range" id="vol" name="vol" min="0" max="50">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_range)

## Input Type Search

The <input type="search"> is used for search fields (a search field behaves like a regular text field).

### **Example**

<form>  
  <label for="gsearch">Search Google:</label>  
  <input type="search" id="gsearch" name="gsearch">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_search)

## Input Type Tel

The <input type="tel"> is used for input fields that should contain a telephone number.

### **Example**

<form>  
  <label for="phone">Enter your phone number:</label>  
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_tel)

## Input Type Time

The <input type="time"> allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

### **Example**

<form>  
  <label for="appt">Select a time:</label>  
  <input type="time" id="appt" name="appt">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_time)

## Input Type Url

The <input type="url"> is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

### **Example**

<form>  
  <label for="homepage">Add your homepage:</label>  
  <input type="url" id="homepage" name="homepage">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_url)

## Input Type Week

The <input type="week"> allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

### **Example**

<form>  
  <label for="week">Select a week:</label>  
  <input type="week" id="week" name="week">  
</form>

This chapter describes the different attributes for the HTML <input> element.

## The value Attribute

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

### **Example**

Input fields with initial (default) values:

<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_value)

## 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!

### **Example**

A read-only input field:

<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_readonly)

## 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!

### **Example**

A disabled input field:

<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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_disabled)

ADVERTISEMENT

## 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.

### **Example**

Set a width for 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" size="4">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_size)

## The maxlength Attribute

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

**Note:** When a maxlength is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

### **Example**

Set a maximum length for 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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_maxlength)

## 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.

### **Example**

Set a max date, a min date, and 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>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_max_min)

## 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.

### **Example**

A file upload field that accepts multiple values:

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

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_multiple)

## 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.

**Tip:** Use the global [title](https://www.w3schools.com/tags/att_global_title.asp) attribute to describe the pattern to help the user.

**Tip:** Learn more about [regular expressions](https://www.w3schools.com/js/js_regexp.asp) in our JavaScript tutorial.

### **Example**

An input field that can contain only three letters (no numbers or special characters):

<form>  
  <label for="country\_code">Country code:</label>  
  <input type="text" id="country\_code" name="country\_code"  
  pattern="[A-Za-z]{3}" title="Three letter country code">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_pattern)

## 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.

### **Example**

An input field with a placeholder text:

<form>  
  <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}">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_placeholder)

## The required Attribute

The input required attribute specifies that an input field must be filled out before submitting the form.

The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

### **Example**

A required input field:

<form>  
  <label for="username">Username:</label>  
  <input type="text" id="username" name="username" required>  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_required)

## The step Attribute

The input step attribute specifies the legal number intervals for an input field.

Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

**Tip:** This attribute can be used together with the max and min attributes to create a range of legal values.

The step attribute works with the following input types: number, range, date, datetime-local, month, time and week.

### **Example**

An input field with a specified legal number intervals:

<form>  
  <label for="points">Points:</label>  
  <input type="number" id="points" name="points" step="3">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_step)

**Note:** Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must also be checked by the receiver (the server)!

## The autofocus Attribute

The input autofocus attribute specifies that an input field should automatically get focus when the page loads.

### **Example**

Let the "First name" input field automatically get focus when the page loads:

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" autofocus><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_autofocus)

## The height and width Attributes

The input height and width attributes specify the height and width of an <input type="image"> element.

**Tip:** Always specify both the height and width attributes for images. If height and width are set, the space required for the image is reserved when the page is loaded. Without these attributes, the browser does not know the size of the image, and cannot reserve the appropriate space to it. The effect will be that the page layout will change during loading (while the images load).

### **Example**

Define an image as the submit button, with height and width attributes:

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

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_height_width)

## The list Attribute

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

### **Example**

An <input> element with pre-defined values in a <datalist>:

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

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_datalist)

## The autocomplete Attribute

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

Autocomplete allows the browser to predict the value. When a user starts to type in a field, the browser should display options to fill in the field, based on earlier typed values.

The autocomplete attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.

### **Example**

An HTML form with autocomplete on, and off for one input field:

<form action="/action\_page.php" autocomplete="on">  
  <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" autocomplete="off"><br><br>  
  <input type="submit" value="Submit">  
</form>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_autocomplete)

**Tip:** In some browsers you may need to activate an autocomplete function for this to work (Look under "Preferences" in the browser's menu).

## HTML Drag and Drop Example

The example below is a simple drag and drop example:

### **Example**

<!DOCTYPE HTML>  
<html>  
<head>  
<script>  
function allowDrop(ev) {  
  ev.preventDefault();  
}  
  
function drag(ev) {  
  ev.dataTransfer.setData("text", ev.target.id);  
}  
  
function drop(ev) {  
  ev.preventDefault();  
  var data = ev.dataTransfer.getData("text");  
  ev.target.appendChild(document.getElementById(data));  
}  
</script>  
</head>  
<body>  
  
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>  
  
<img id="drag1" src="img\_logo.gif" draggable="true" ondragstart="drag(event)" width="336" height="69">  
  
</body>  
</html>

HTML web storage; better than cookies.

What is HTML Web Storage?

With web storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

HTML web storage provides two objects for storing data on the client:

* window.localStorage - stores data with no expiration date
* window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)

Before using web storage, check browser support for localStorage and sessionStorage:

if (typeof(Storage) !== "undefined") {  
  // *Code for localStorage/sessionStorage.*  
} else {  
  // Sorry! No Web Storage support..  
}