#### **Introduction**

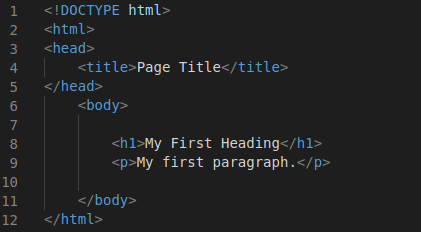
HTML is the standard markup language for creating Web pages.

## 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.

## A Simple HTML Document

### Example



### Example Explained

* The <!DOCTYPE html> declaration defines that this document is an HTML5 document
* The <html> element is the root element of an HTML page
* The <head> element contains meta information about the HTML page
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The <h1> element defines a large heading
* The <p> element defines a paragraph

## What is an HTML Element?

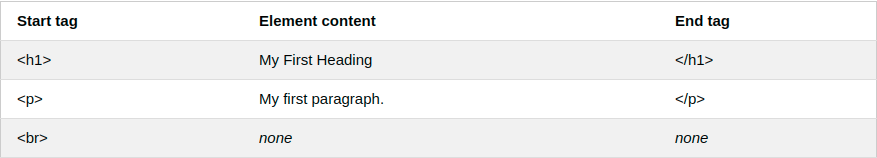
An HTML element is defined by a start tag, some content, and an end tag:

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

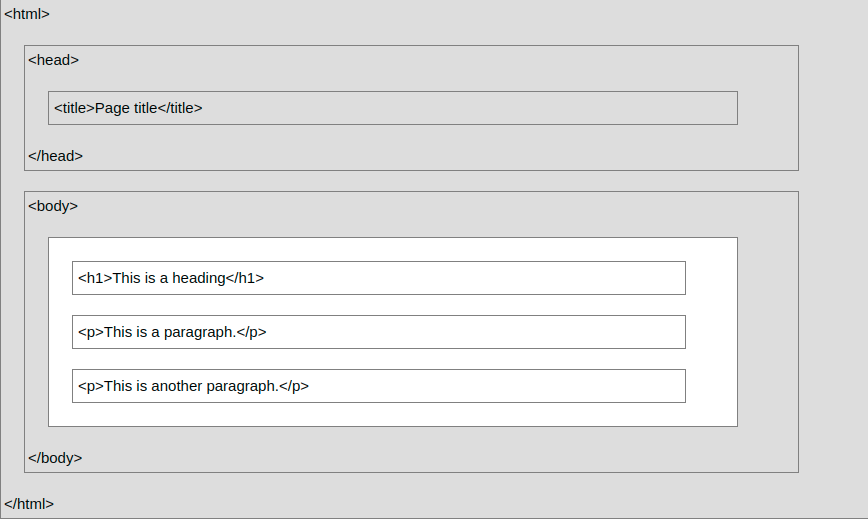
The HTML element is everything from the start tag to the end tag:

<h1>My First Heading</h1>

<p>My first paragraph.</p>



Below is a visualization of an HTML page structure:



## HTML Documents

All HTML documents must start with a document type declaration: <!DOCTYPE html>.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

## The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The <!DOCTYPE> declaration is not case sensitive.

The <!DOCTYPE> declaration for HTML5 is:

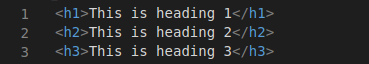


## HTML Headings

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

<h1> defines the most important heading. <h6> defines the least important heading:

### Example



## HTML Paragraphs

HTML paragraphs are defined with the <p> tag:

### Example



## HTML Links

HTML links are defined with the <a> tag:

### Example



The link's destination is specified in the href attribute.

Attributes are used to provide additional information about HTML elements.

You will learn more about attributes in a later chapter.

## HTML Images

HTML images are defined with the <img> tag.

The source file (src), alternative text (alt), width, and height are provided as attributes:



## How to View HTML Source?

Have you ever seen a Web page and wondered "Hey! How did they do that?"

### View HTML Source Code:

Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in Edge), or similar in other browsers. This will open a window containing the HTML source code of the page.

### Inspect an HTML Element:

Right-click on an element (or a blank area), and choose "Inspect" or "Inspect Element" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

## HTML Elements

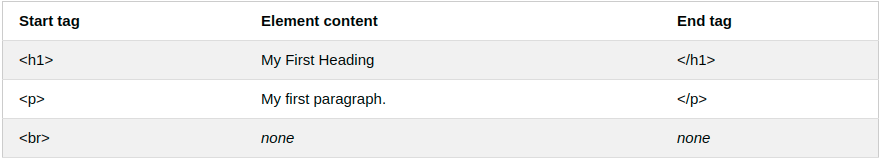
An HTML element is defined by a start tag, some content, and an end tag:

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

The HTML element is everything from the start tag to the end tag:

<h1>My First Heading</h1>

<p>My first paragraph.</p>

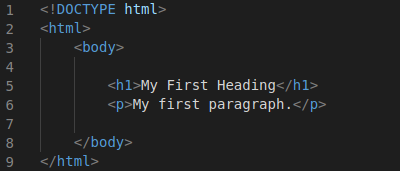


## Nested HTML Elements

HTML elements can be nested (this means that elements can contain other elements).

All HTML documents consist of nested HTML elements.

The following example contains four HTML elements (<html>, <body>, <h1> and <p>):

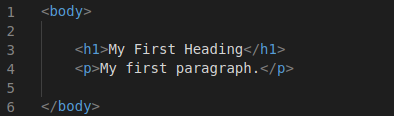


### Example Explained

The <html> element is the root element and it defines the whole HTML document.

It has a start tag <html> and an end tag </html>.

Then, inside the <html> element there is a <body> element:



The <body> element defines the document's body.

It has a start tag <body> and an end tag </body>.

Then, inside the <body> element there is two other elements: <h1> and <p>:



The <h1> element defines a heading.

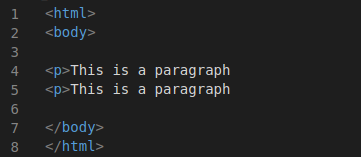
It has a start tag <h1> and an end tag </h1>:

The <p> element defines a paragraph.

It has a start tag <p> and an end tag </p>:

## Never Skip the End Tag

Some HTML elements will display correctly, even if you forget the end tag:



However, never rely on this! Unexpected results and errors may occur if you forget the end tag!

## Empty HTML Elements

HTML elements with no content are called empty elements.

The <br> tag defines a line break, and is an empty element without a closing tag:

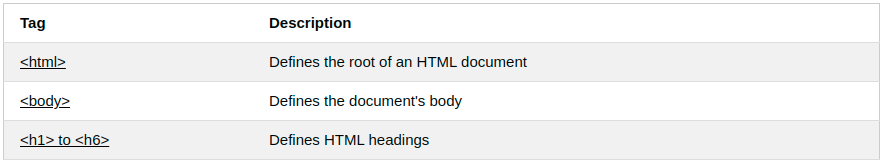


## HTML is Not Case Sensitive

HTML tags are not case sensitive: <P> means the same as <p>.

The HTML standard does not require lowercase tags, but it is recommended to use lowercase in HTML, and demands lowercase for stricter document types like XHTML.

## HTML Tag Reference



## 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"

## The href Attribute

The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to:



## The src Attribute

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed:



There are two ways to specify the URL in the src attribute:

1. Absolute URL - Links to an external image that is hosted on another website. Example: src="https://www.hackerearth.com/images/hack\_logo.jpg".

Notes: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="hack\_logo.jpg". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/hack\_logo.jpg".

Tip: It is almost always best to use relative URLs. They will not break if you change domain.

## The width and height Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):



## The alt Attribute

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.



## The style Attribute

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



## The title Attribute

The title attribute defines some extra information about an element.

The value of the title attribute will be displayed as a tooltip when you mouse over the element:



## Single or Double Quotes?

Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:



Or vice versa:



## HTML Headings

HTML headings are titles or subtitles that you want to display on a webpage.

### Example

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

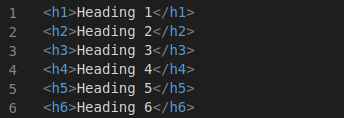
##### Heading 5

###### Heading 6

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

<h1> defines the most important heading. <h6> defines the least important heading.

### Example



## Headings Are Important

Search engines use the headings to index the structure and content of your web pages.

Users often skim a page by its headings. It is important to use headings to show the document structure.

<h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.

## Bigger Headings

Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute, using the CSS font-size property:



# HTML Styles

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

## The HTML Style Attribute

Setting the style of an HTML element, can be done with the style attribute.

The HTML style attribute has the following syntax:



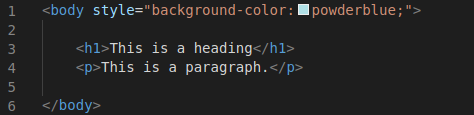
The *property* is a CSS property. The *value* is a CSS value.

## Background Color

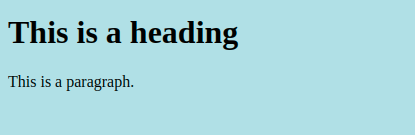
The CSS background-color property defines the background color for an HTML element.

### Example

Set the background color for a page to powderblue:



Output is going to be as follows

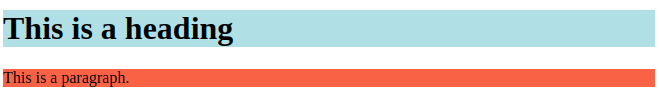


### Example

Set background color for two different elements:



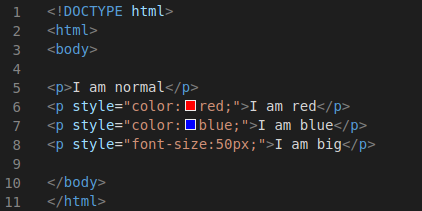
Output is going to be as follows:



## Text Color

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

### Example



## Fonts

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

### Example



## Text Alignment

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

### Example



## 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
* <sub> - Subscript text
* <sup> - Superscript text

## HTML <b> and <strong> Elements

The HTML <b> element defines bold text, without any extra importance.

The HTML <strong> element defines text with strong importance. The content inside is typically displayed in bold.

### Example



The HTML <i> element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

The HTML <em> element defines emphasized text. The content inside is typically displayed in italic.

The HTML <mark> element defines text that should be marked or highlighted:

## HTML <del> Element

The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:

### Example



Output is as follows:



## HTML <ins> Element

The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text.

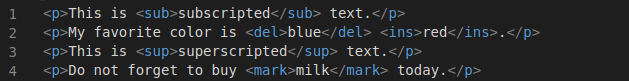
## HTML <sub> Element

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H2O.

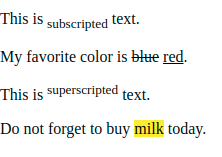
## HTML <sup> Element

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1]:

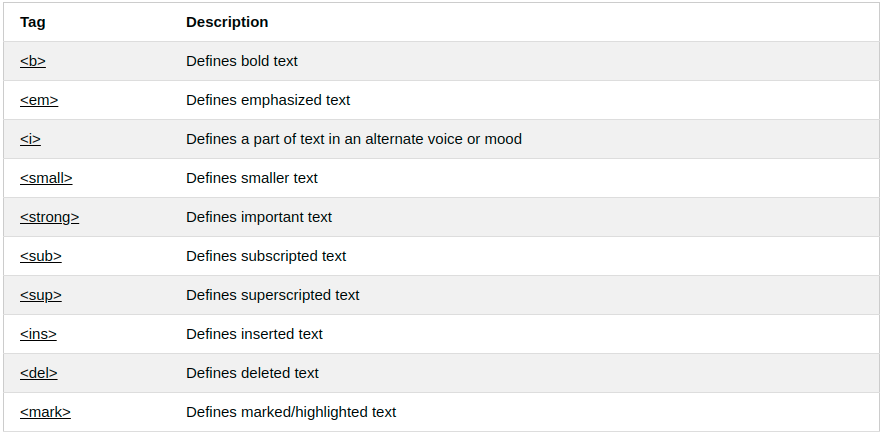
### Example



The output is as follows:



## HTML Text Formatting Elements



## HTML <abbr> for Abbreviations

The HTML <abbr> tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Tip: Use the global title attribute to show the description for the abbreviation/acronym when you mouse over the element.

### Example



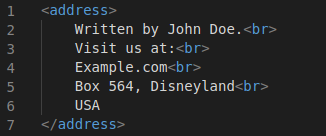
## HTML <address> for Contact Information

The HTML <address> tag defines the contact information for the author/owner of a document or an article.

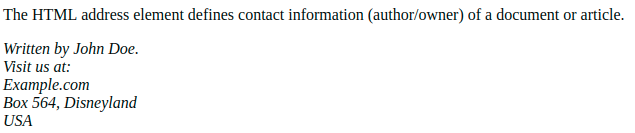
The contact information can be an email address, URL, physical address, phone number, social media handle, etc.

The text in the <address> element usually renders in *italic,* and browsers will always add a line break before and after the <address> element.

### Example



The output is as follows:



## HTML <cite> for Work Title

The HTML <cite> tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

Note: A person's name is not the title of a work.

The text in the <cite> element usually renders in *italic*.

## HTML <bdo> for Bi-Directional Override

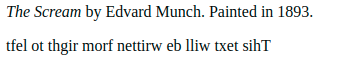
BDO stands for Bi-Directional Override.

The HTML <bdo> tag is used to override the current text direction:

### Example

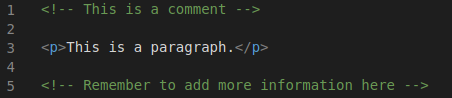


The output is as follows:



## HTML Comment Tags

You can add comments to your HTML source by using the following syntax:



Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

# HTML Background Images

A background image can be specified for almost any HTML element.

## Background Image on a HTML element

To add a background image on an HTML element, use the HTML style attribute and the CSS background-image property:

### Example

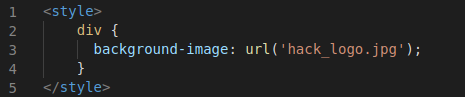
Add a background image on a HTML element:



You can also specify the background image in the <style> element, in the <head> section:

### Example

Specify the background image in the <style> element:

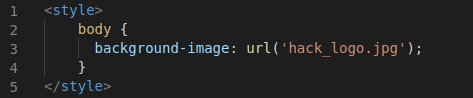


## Background Image on a Page

If you want the entire page to have a background image, you must specify the background image on the <body> element:

### Example

Add a background image for the entire page:



# HTML Tables

HTML tables allow web developers to arrange data into rows and columns.

## Define an HTML Table

The <table> tag defines an HTML table.

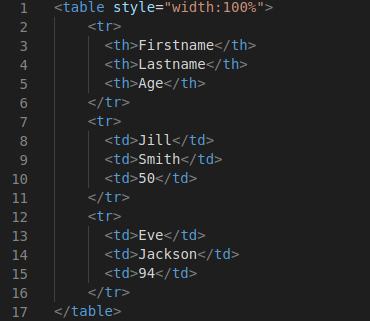
Each table row is defined with a <tr> tag. Each table header is defined with a <th> tag. Each table data/cell is defined with a <td> tag.

By default, the text in <th> elements are bold and centered.

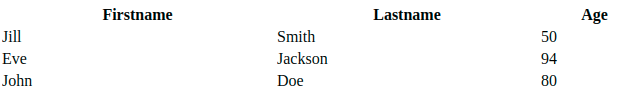
By default, the text in <td> elements are regular and left-aligned.

### Example

A simple HTML table:



Output for the above html code is as follows:

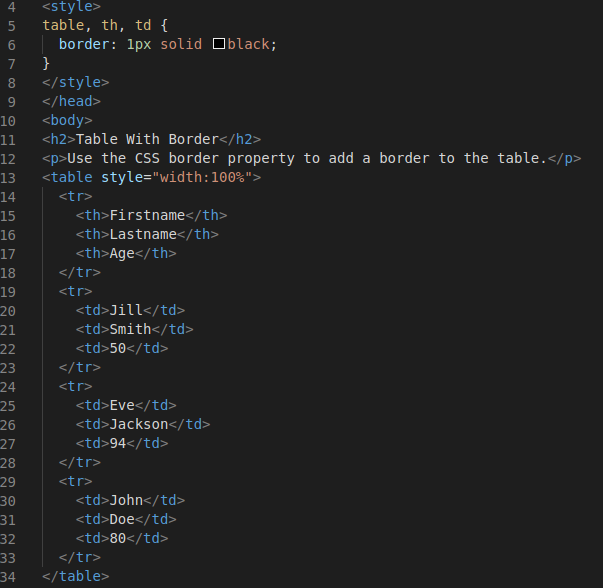


**Note**: The <td> elements are the data containers of the table.

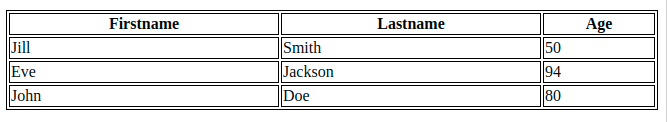
They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

## HTML Table - Add a Border

To add a border to a table, use the CSS border property:



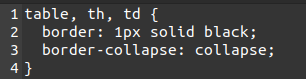
Output will be as follows:



## HTML Table - Collapsed Borders

To let the borders collapse into one border, add the CSS border-collapse property:

### Example



## HTML Table - Add Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the CSS padding property:

### Example



## HTML Table - Left-align Headings

By default, table headings are bold and centered.

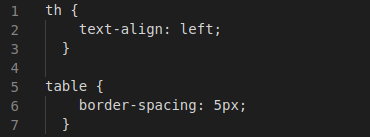
To left-align the table headings, use the CSS text-align property:

## HTML Table - Add Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the CSS border-spacing property:

### Example



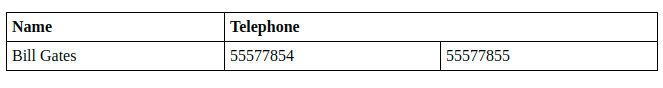
## HTML Table - Cell that Span Many Columns

To make a cell span more than one column, use the colspan attribute:

### Example



The output is as follows:

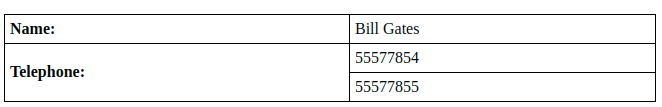


## HTML Table - Cell that Span Many Rows

To make a cell span more than one row, use the rowspan attribute:



The output is as follows:



## HTML Table - Add a Caption

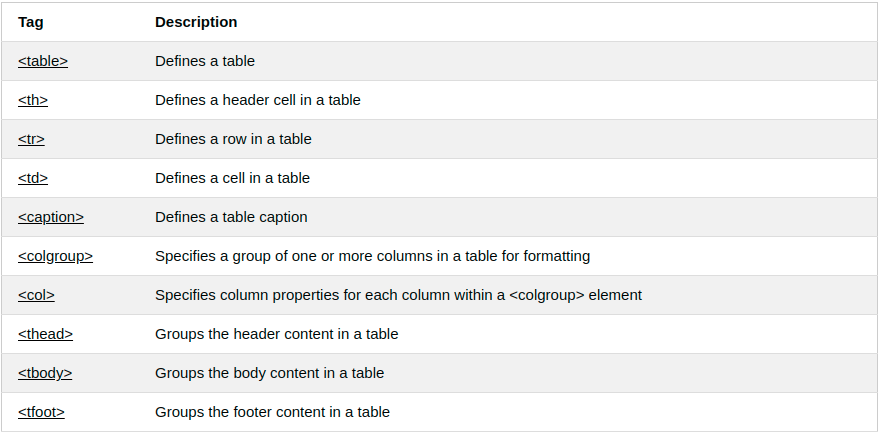
To add a caption to a table, use the <caption> tag:

Note: The <caption> tag must be inserted immediately after the <table> tag.



## Key Points

* Use the HTML <table> element to define a table
* Use the HTML <tr> element to define a table row
* Use the HTML <td> element to define a table data
* Use the HTML <th> element to define a table heading
* Use the HTML <caption> element to define a table caption
* Use the CSS border property to define a border
* Use the CSS border-collapse property to collapse cell borders
* Use the CSS padding property to add padding to cells
* Use the CSS text-align property to align cell text
* Use the CSS border-spacing property to set the spacing between cells
* Use the colspan attribute to make a cell span many columns
* Use the rowspan attribute to make a cell span many rows
* Use the id attribute to uniquely define one table



# HTML Lists

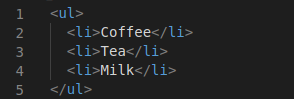
HTML lists allow web developers to group a set of related items in lists.

## Unordered HTML List

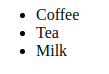
An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

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

### Example



Output is as follows:

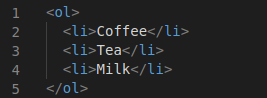


## Ordered HTML List

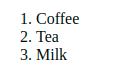
An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

### Example



Output is as follows:



## HTML Description Lists

HTML also supports 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:

## HTML List Tags

Tag and Description

<ul> Defines an unordered list

<ol> Defines an ordered list

<li> Defines a list item

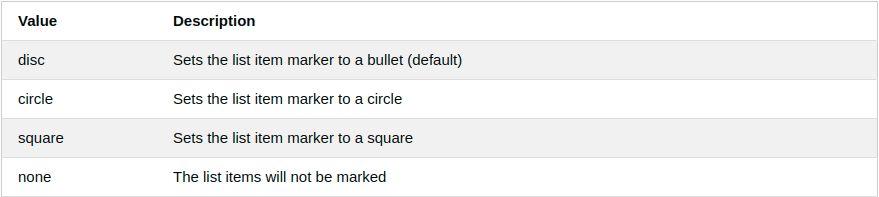
<dl> Defines a description list

<dt> Defines a term in a description list

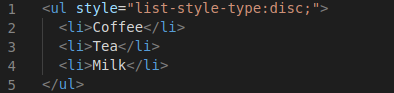
<dd> Describes the term in a description list

## Unordered HTML List - Choose List Item Marker

The CSS list-style-type property is used to define the style of the list item marker. It can have one of the following values:

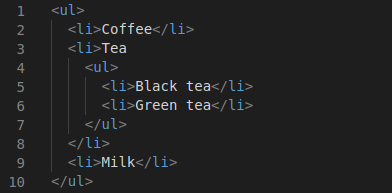


### Example - Disc



## Nested HTML Lists

Lists can be nested (list inside list):



**Note**: A list item (<li>) can contain a new list, and other HTML elements, like images and links, etc.

## Horizontal List with CSS

HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a navigation menu:

<!DOCTYPE html>

<html>

<head>

<style>

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333333;

}

li {

float: left;

}

li a {

display: block;

color: white;

text-align: center;

padding: 16px;

text-decoration: none;

}

li a:hover {

background-color: #111111;

}

</style>

</head>

<body>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#news">News</a></li>

<li><a href="#contact">Contact</a></li>

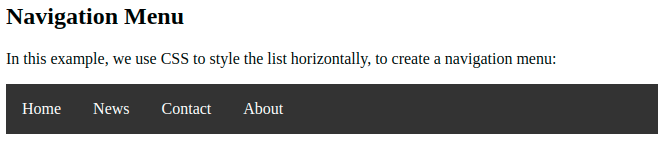
<li><a href="#about">About</a></li>

</ul>

</body>

</html>

Output is as follows:



# HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.

There are two display values: block and inline.

## Block-level Elements

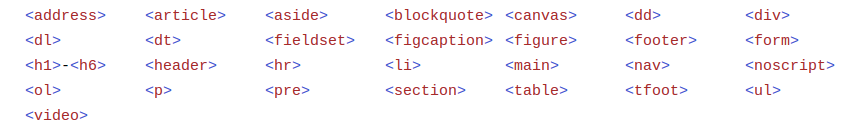
A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

### Example

<div>Hello World</div>

Here are the block-level elements in HTML:



## Inline Elements

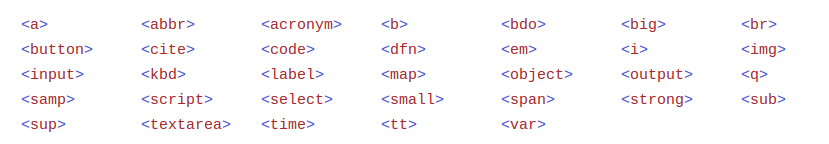
An inline element does not start on a new line and it only takes up as much width as necessary.

This is a <span> element inside a paragraph.

### Example

<span>Hello World</span>

Here are the inline elements in HTML:



## The <div> Element

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.

When used together with CSS, the <div> element can be used to style blocks of content:

* There are two display values: block and inline
* A block-level element always starts on a new line and takes up the full width available
* An inline element does not start on a new line and it only takes up as much width as necessary
* The <div> element is a block-level and is often used as a container for other HTML elements
* The <span> element is an inline container used to mark up a part of a text, or a part of a document

# HTML class Attribute

The HTML class attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

## Using The class Attribute

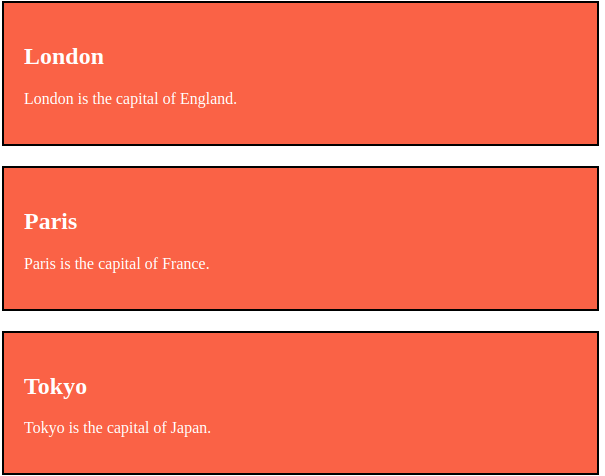
The class attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

In the following example we have three <div> elements with a class attribute with the value of "city". All of the three <div> elements will be styled equally according to the .city style definition in the head section:

### Example



The output for the above code is as follows:



In the following example we have two <span> elements with a class attribute with the value of "note". Both <span> elements will be styled equally according to the .note style definition in the head section:

**Tip**: The class attribute can be used on any HTML element.

**Note**: The class name is case sensitive!

**Tip**: You can learn much more about CSS in our CSS Tutorial.

## The Syntax For Class

To create a class; write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}:



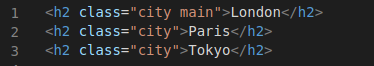
## Multiple Classes

HTML elements can belong to more than one class.

To define multiple classes, separate the class names with a space, e.g. <div class="city main">. The element will be styled according to all the classes specified.

In the following example, the first <h2> element belongs to both the city class and also to the main class, and will get the CSS styles from both of the classes:

### Example



## Different Elements Can Share Same Class

Different HTML elements can point to the same class name.

In the following example, both <h2> and <p> points to the "city" class and will share the same style:

### Example



# HTML id Attribute

The HTML id attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

## Using The id Attribute

The id attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document.

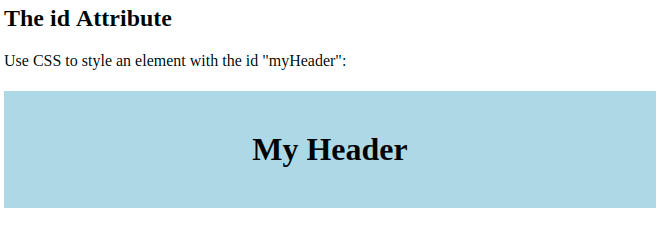
The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

The syntax for id is: write a hash character (#) character, followed by an id name. Then, define the CSS properties within curly braces {}.

In the following example we have an <h1> element that points to the id name "myHeader". This <h1> element will be styled according to the #myHeader style definition in the head section:



**Output**



## Difference Between Class and ID

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page:

## HTML Bookmarks with ID and Links

HTML bookmarks are used to allow readers to jump to specific parts of a webpage.

Bookmarks can be useful if your page is very long.

To use a bookmark, you must first create it, and then add a link to it.

Then, when the link is clicked, the page will scroll to the location with the bookmark.

## Example

First, create a bookmark with the id attribute:

<h2 id="C4">Chapter 4</h2>

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

<a href="html\_demo.html#C4">Jump to Chapter 4</a>

# HTML Iframes

An HTML iframe is used to display a web page within a web page.

## HTML Iframe Syntax

The HTML <iframe> tag specifies an inline frame.

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

### Syntax

<iframe src="*url*" title="*description*">

Tip: It is a good practice to always include a title attribute for the <iframe>. This is used by screen readers to read out what the content of the iframe is.

## Iframe - Set Height and Width

Use the height and width attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

### Example

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

Or you can add the style attribute and use the CSS height and width properties:

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

## Iframe - Remove the Border

By default, an iframe has a border around it.

To remove the border, add the style attribute and use the CSS border property:

### Example

<iframe src="demo\_iframe.htm" style="border:none;" title="Iframe Example"></iframe>

## Iframe - Target for a Link

An iframe can be used as the target frame for a link.

The target attribute of the link must refer to the name attribute of the iframe:

<iframe src="demo\_iframe.htm" name="iframe\_a" title="Iframe Example"></iframe>

<p><a href="https://www.hackerearth.com" target="iframe\_a">Hackerearth.com</a></p>

## The HTML <script> Tag

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.

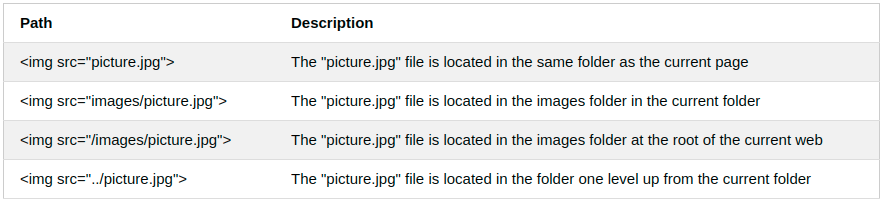
To select an HTML element, JavaScript most often uses the document.getElementById() method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

### Example



# HTML File Paths



A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files, like:

* Web pages
* Images
* Style sheets
* JavaScripts

## Absolute File Paths

An absolute file path is the full URL to a file:

### Example

<img src="https://www.hackerearth.com/images/picture.jpg" alt="Logo">

The HTML <meta> Element

The <meta> element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but are used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

## Examples

Define the character set used:

<meta charset="UTF-8">

Define keywords for search engines:

<meta name="keywords" content="HTML, CSS, JavaScript">

Define a description of your web page:

<meta name="description" content="Free Web tutorials">

Define the author of a page:

<meta name="author" content="John Doe">

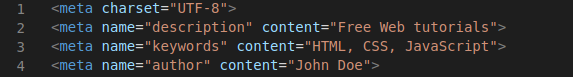
Refresh document every 30 seconds:

<meta http-equiv="refresh" content="30">

Setting the viewport to make your website look good on all devices:

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

Example of <meta> tags:



## Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following <meta> element in all your web pages:

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

This gives the browser instructions on how to control the page's dimensions and scaling.

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.

## The HTML <base> Element

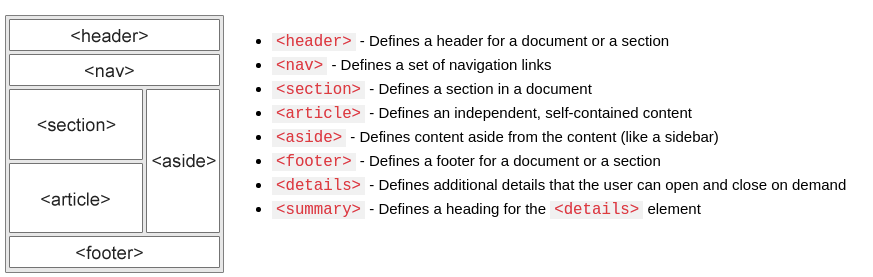
The <base> element specifies the base URL and/or target for all relative URLs in a page.

The <base> tag must have either an href or a target attribute present, or both.

There can only be one single <base> element in a document!

## HTML Layout Elements

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



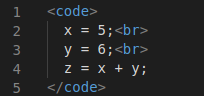
There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

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

# HTML Computer Code Elements

HTML contains several elements for defining user input and computer code.

### Example



Result:



## HTML <kbd> For Keyboard Input

The HTML <kbd> element is used to define keyboard input. The content inside is displayed in the browser's default monospace font.

### Example

Define some text as keyboard input in a document:

<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>

## HTML <samp> For Program Output

The HTML <samp> element is used to define sample output from a computer program. The content inside is displayed in the browser's default monospace font.

### Example

Define some text as sample output from a computer program in a document:

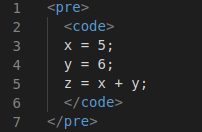
<p>Message from my computer:</p>

<p><samp>File not found.<br>Press F1 to continue</samp></p>

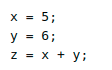
Notice that the <code> element does not preserve extra whitespace and line-breaks.

To fix this, you can put the <code> element inside a <pre> element:

### Example



Result:

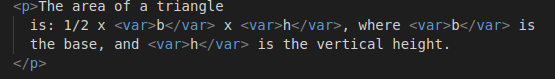


## HTML <var> For Variables

The HTML <var> element is used to defines a variable in programming or in a mathematical expression. The content inside is typically displayed in italic.

### Example

Define some text as variables in a document:



Result:



# HTML Semantic Elements

Semantic elements = elements with a meaning.

## What are Semantic Elements?

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

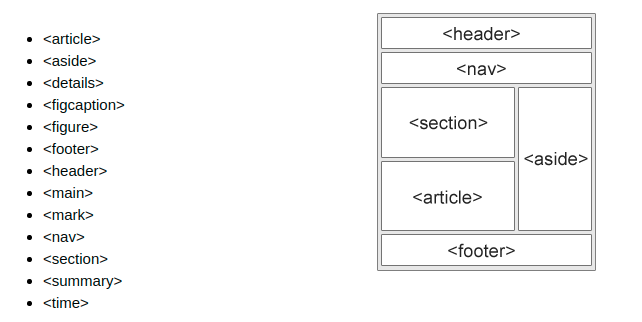
Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.

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:



## HTML <section> Element

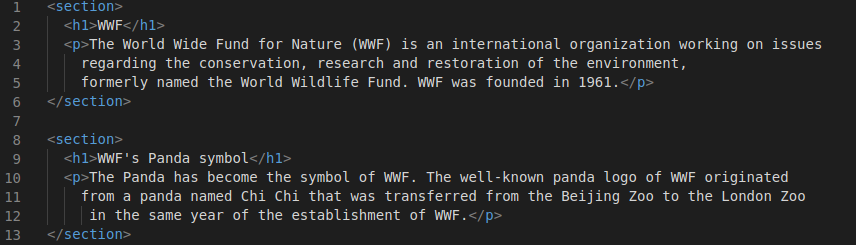
The <section> element defines a section in a document.

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

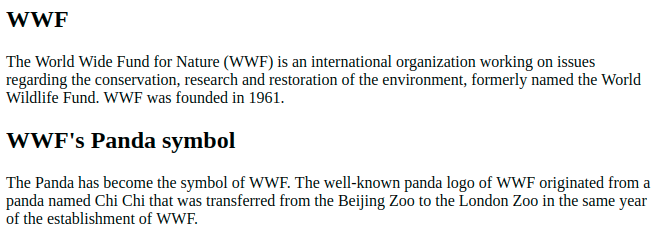
A web page could normally be split into sections for introduction, content, and contact information.

### Example

Two sections in a document:



Gives output as



## HTML <article> Element

The <article> element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where an <article> element can be used:

* Forum post
* Blog post
* Newspaper article

## Example



The output for the above code is as follows:



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

The <article> element specifies independent, self-contained content.

The <section> element defines section in a document.

Can we use the definitions to decide how to nest those elements? No, we cannot!

So, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <section> elements.

## HTML <header> Element

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

* one or more heading elements (<h1> - <h6>)
* logo or icon
* authorship information

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

## HTML <footer> Element

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

* authorship information
* copyright information
* contact information
* sitemap
* back to top links
* related documents

You can have several <footer> elements in one document.

A footer section in a document:

<footer>

<p>Author: Hege Refsnes</p>

<p><a href="mailto:hege@example.com">hege@example.com</a></p>

</footer>

## HTML <nav> Element

The <nav> element defines a set of navigation links.

Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

## HTML <aside> Element

The <aside> element defines some content aside from the content it is placed in (like a sidebar).

The <aside> content should be indirectly related to the surrounding content.

## HTML <figure> and <figcaption> Elements

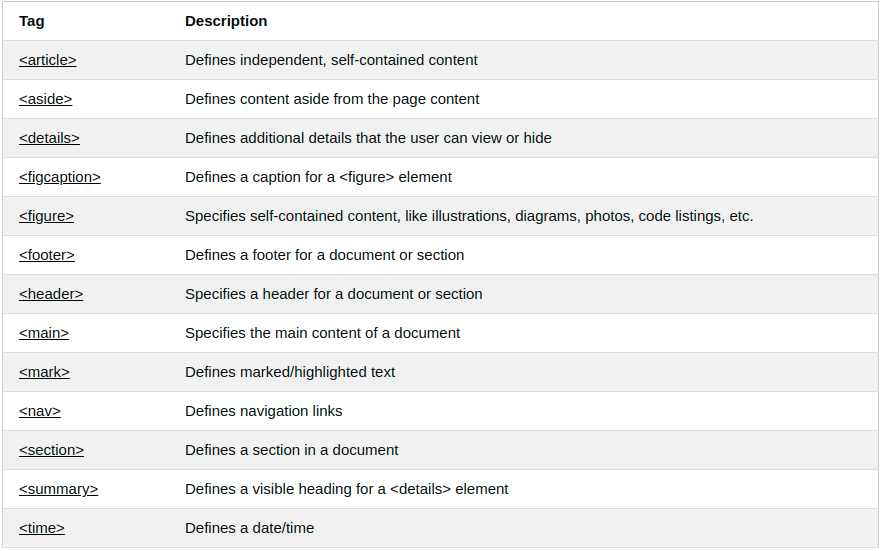
The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

The <figcaption> tag defines a caption for a <figure> element. The <figcaption> element can be placed as the first or as the last child of a <figure> element.

The <img> element defines the actual image/illustration.

## Semantic Elements in HTML

Below is a list of some of the semantic elements in HTML.

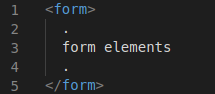


# HTML Forms

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:



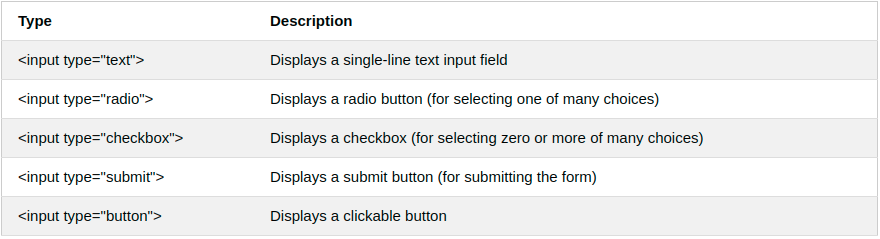
The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

## The <input> Element

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

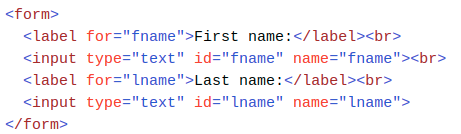
An <input> element can be displayed in many ways, depending on the type attribute.

Here are some examples:

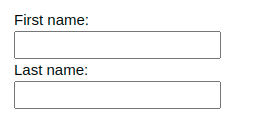


## Text Fields

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



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



Note: The default width of an input field is 20 characters.

## The <label> Element

Notice the use of the <label> element in the example above.

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 focus on the input element.

The <label> element also help 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.

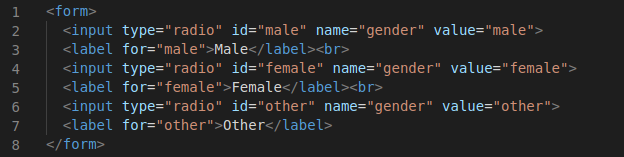
The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

## Radio Buttons

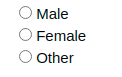
The <input type="radio"> defines a radio button.

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

A form with radio buttons:



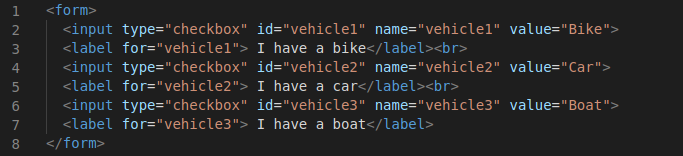
This is how the HTML code above will be displayed in a browser:



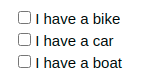
## Checkboxes

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

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



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



## The Submit Button

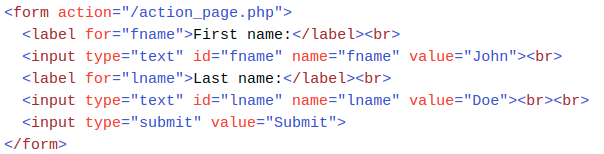
The <input type="submit"> defines a button for submitting the form data to a form-handler.

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

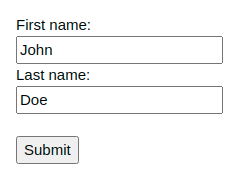
The form-handler is specified in the form's action attribute.

### Example

A form with a submit button:



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



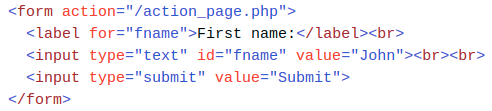
## The Name Attribute for <input>

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.

### Example

This example will not submit the value of the "First name" input field:



# HTML Form Attributes

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

## The Action Attribute

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.

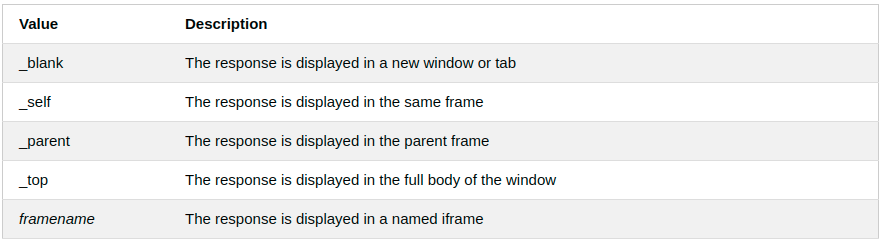
In the example below, the form data is sent to a file called "action\_page.php". This file contains a server-side script that handles the form data:

Tip: If the action attribute is omitted, the action is set to the current page.

## 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:



The default value is \_self which means that the response will open i the current window.

### Example

Here, the submitted result will open in a new browser tab:

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

## The Method Attribute

The method attribute specifies the HTTP method to use 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.

### Example

This example uses the GET method when submitting the form data:

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

### Example

This example uses the POST method when submitting the form data:

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

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

## 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.

### Example

A form with autocomplete on:

<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.

### Example

A form with a novalidate attribute:

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

# HTML Form Elements

This chapter describes all the different HTML form elements.

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>

The <input> Element

One of the most used form element is the <input> element.

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

## The <label> Element

The <label> element defines a label for several form elements.

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

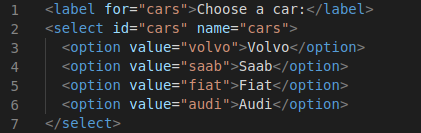
The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

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

## The <select> Element

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

### Example



The output for the above code is as follows:



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

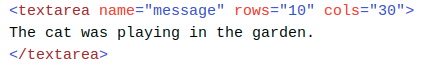
By default, the first item in the drop-down list is selected.

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

## The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

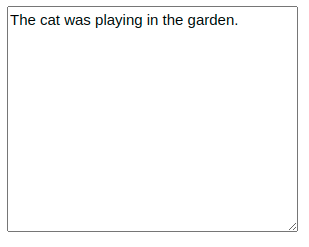
### Example



The rows attribute specifies the visible number of lines in a text area.

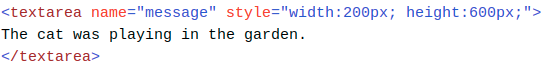
The cols attribute specifies the visible width of a text area.

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



You can also define the size of the text area by using CSS:

### Example



## The <button> Element

The <button> element defines a clickable button:

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

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



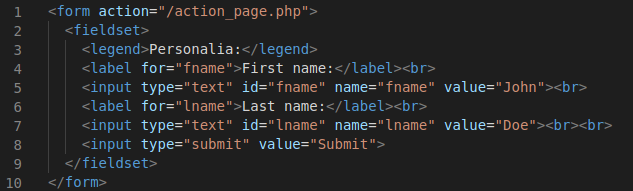
**Note**: Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

## The <fieldset> and <legend> Elements

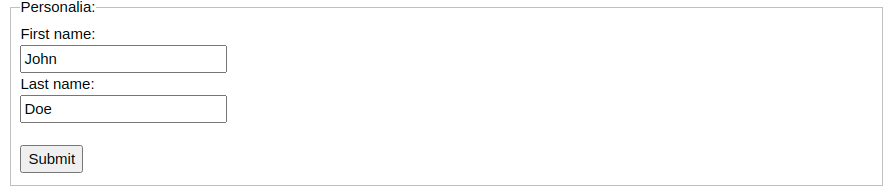
The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

### Example



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



## The <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

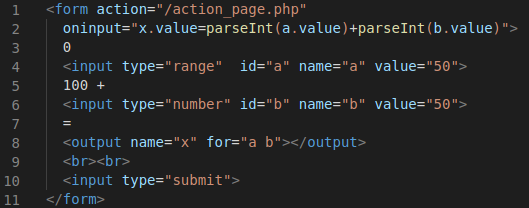
The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

## The <output> Element

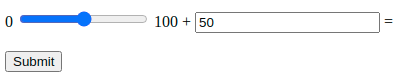
The <output> element represents the result of a calculation (like one performed by a script).

### Example

Perform a calculation and show the result in an <output> element:



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





## HTML Input Types

## 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">

**Tip**: The default value of the type attribute is "text".

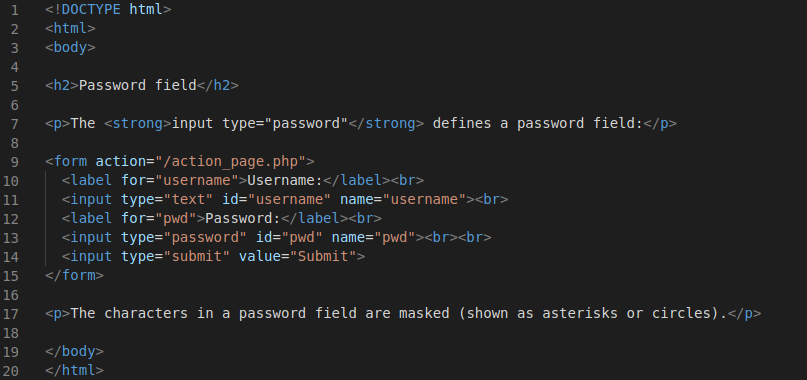
## Input Type Text

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

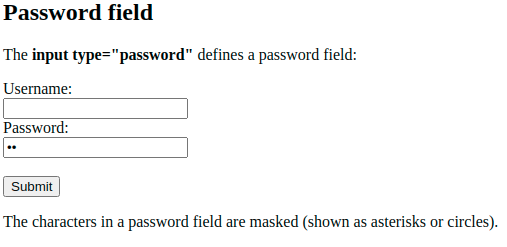
## Input Type Password

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

### Example



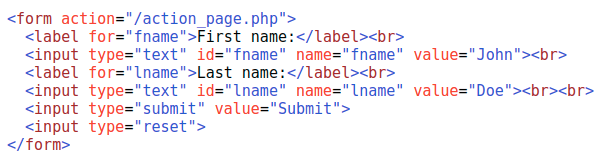
This is how the HTML code above will be displayed in a browser:



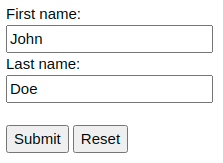
## Input Type Reset

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

### Example



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



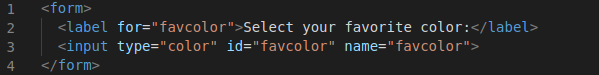
If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

## 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

.

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

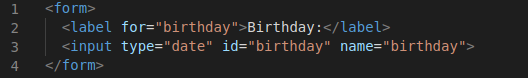


## Input Type Date

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

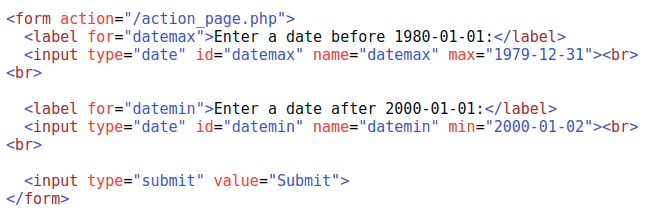


This is how the HTML code above will be displayed in a browser:

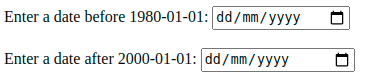


You can also use the min and max attributes to add restrictions to dates:

### Example



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



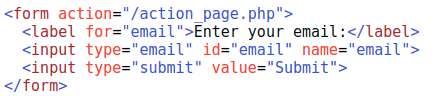
## Input Type Email

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.

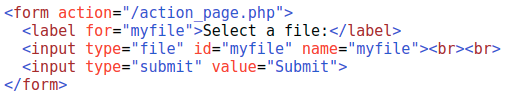
### Example



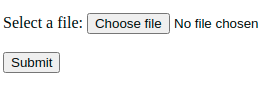
## Input Type File

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

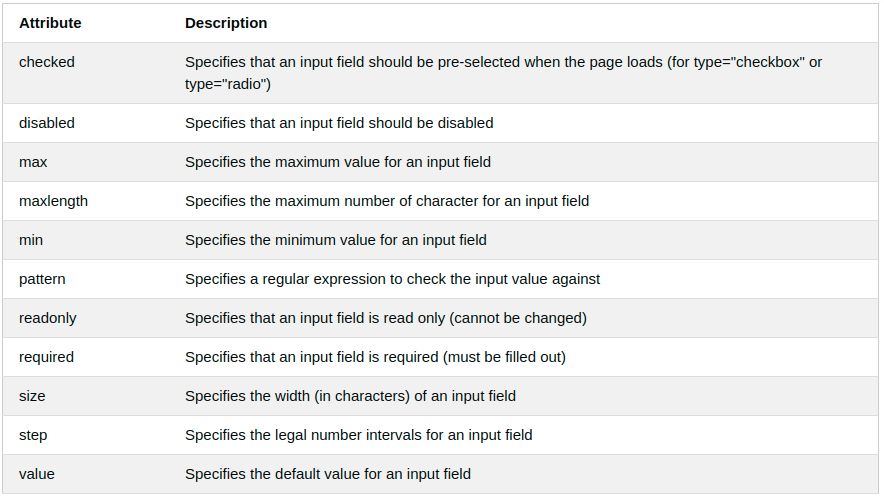
### Example



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



## Input Restrictions



# 

# HTML Canvas Graphics

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.



## What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## Browser Support

The numbers in the table specify the first browser version that fully supports the <canvas> element.



## Canvas Examples

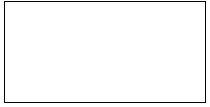
A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

<canvas id="myCanvas" width="200" height="100"></canvas>

Note: Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Here is an example of a basic, empty canvas:



### Example

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">

</canvas>

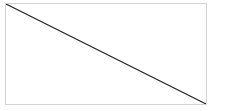
### Example

## Add a JavaScript

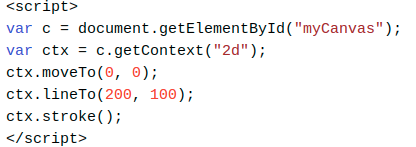
After creating the rectangular canvas area, you must add a JavaScript to do the drawing.

Here are some examples:

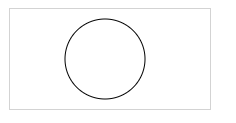
### Draw a Line



### Example



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



# HTML SVG Graphics

SVG defines vector-based graphics in XML format.

## What is SVG?

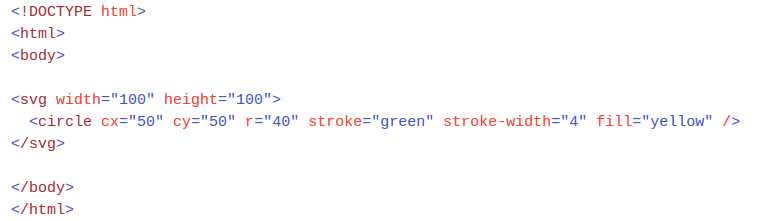
* SVG stands for Scalable Vector Graphics
* SVG is used to define graphics for the Web
* SVG is a W3C recommendation

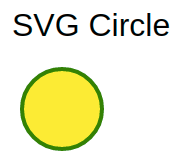
## The HTML <svg> Element

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

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

### Example





## Differences Between SVG and Canvas

SVG is a language for describing 2D graphics in XML.

Canvas draws 2D graphics, on the fly (with a JavaScript).

SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

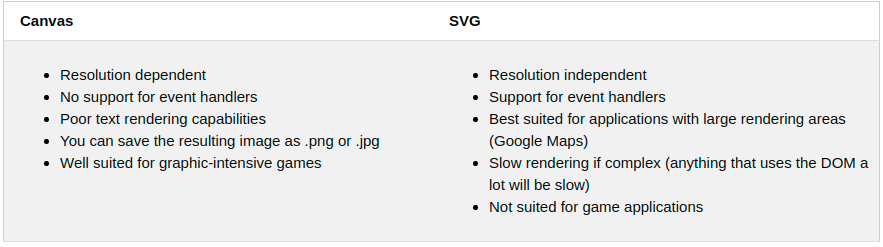
In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic

## 

## Comparison of Canvas and SVG

The table below shows some important differences between Canvas and SVG:



# HTML Multimedia

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

## What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

## Browser Support

The first web browsers had support for text only, limited to a single font in a single color.

Later came browsers with support for colors, fonts, images, and multimedia!

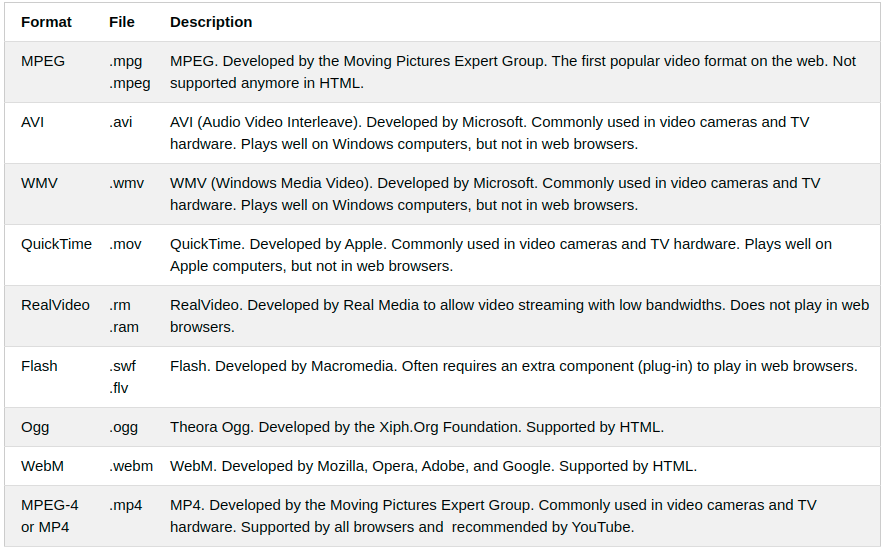
## Multimedia Formats

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.

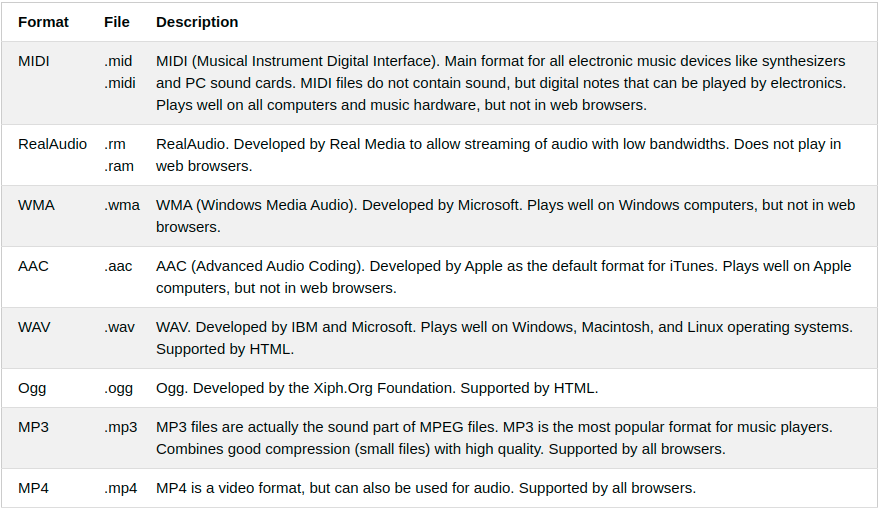
## Common Video Formats



## Common Audio Formats

MP3 is the best format for compressed recorded music. The term MP3 has become synonymous with digital music.

If your website is about recorded music, MP3 is the choice.

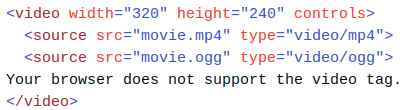


# HTML Video

## The HTML <video> Element

To show a video in HTML, use the <video> element:

### Example



## 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.

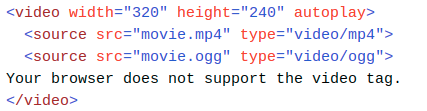
The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

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:

### Example



# HTML Plug-ins

Plug-ins are computer programs that extend the standard functionality of the browser.

## Plug-ins

Plug-ins were designed to be used for many different purposes:

* To run Java applets
* To run Microsoft ActiveX controls
* To display Flash movies
* To display maps
* To scan for viruses
* To verify a bank id

## The <object> Element

The <object> element is supported by all browsers.

The <object> element defines an embedded object within an HTML document.

It was designed to embed plug-ins (like Java applets, PDF readers, and Flash Players) in web pages, but can also be used to include HTML in HTML:

### Example

<object width="100%" height="500px" data="snippet.html"></object>

### Example

<object data="audi.jpeg"></object>

# HTML YouTube Videos

The easiest way to play videos in HTML, is to use YouTube.

## Struggling with Video Formats?

Converting videos to different formats can be difficult and time-consuming.

An easier solution is to let YouTube play the videos in your web page.

## YouTube Video Id

YouTube will display an id (like tgbNymZ7vqY), when you save (or play) a video.

You can use this id, and refer to your video in the HTML code.

## Playing a YouTube Video in HTML

To play your video on a web page, do the following:

* Upload the video to YouTube
* Take a note of the video id
* Define an <iframe> element in your web page
* Let the src attribute point to the video URL
* Use the width and height attributes to specify the dimension of the player
* Add any other parameters to the URL (see below)

### Example

<iframe width="420" height="315"

src="https://www.youtube.com/embed/tgbNymZ7vqY">

</iframe>

## YouTube Autoplay + Mute

You can let your video start playing automatically when a user visits the page, by adding autoplay=1 to the YouTube URL.

### YouTube - Autoplay + Muted

<iframe width="420" height="315"

src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&mute=1">

</iframe>

----------------------Reference Credits: W3SCHOOLS,TutorialsPoint-------------------------