

# **HTML**

# **Notes**



**- By Dipesh Adelkar**

# HTML Introduction

TML 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

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

### 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>`

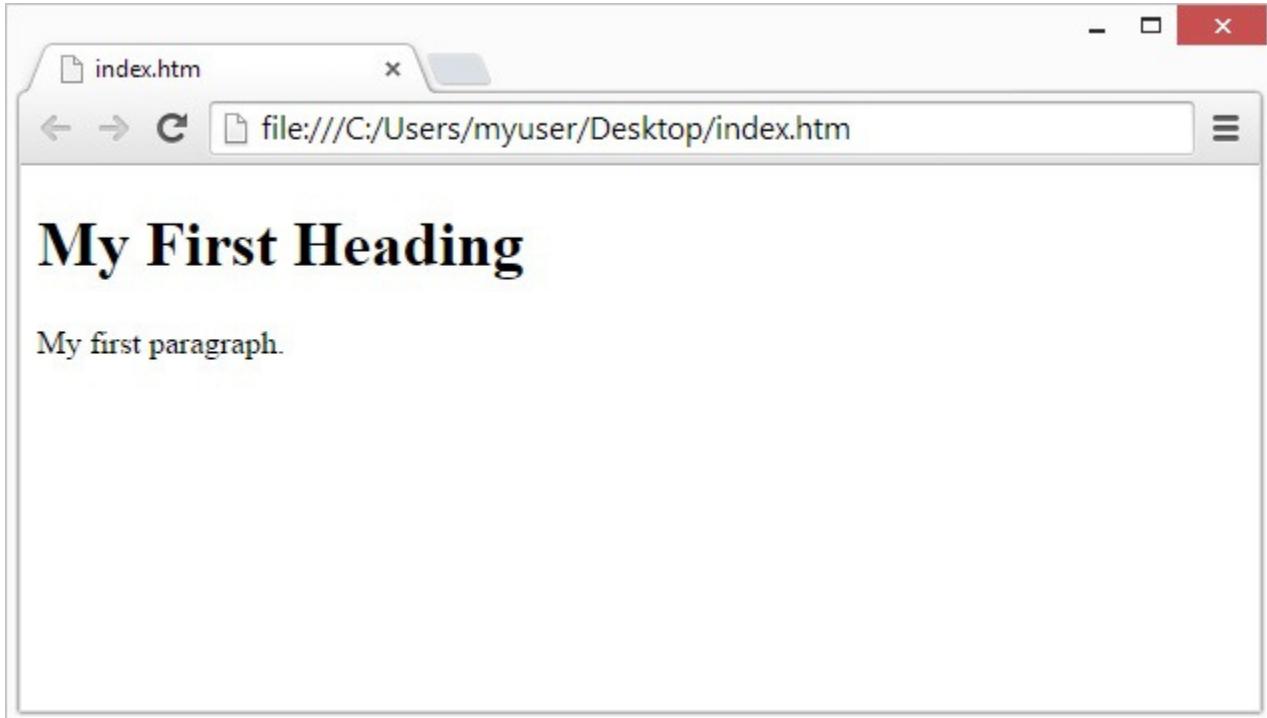
`<p>My first paragraph.</p>`

Start tag	Element content	End tag
<code>&lt;h1&gt;</code>	My First Heading	<code>&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	My first paragraph.	<code>&lt;/p&gt;</code>
<code>&lt;br&gt;</code>	<i>None</i>	<i>none</i>

# Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:



# HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

**Note:** The content inside the `<body>` section (the white area above) will be displayed in a browser. The content inside the `<title>` element will be shown in the browser's title bar or in the page's tab.

## HTML History

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

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2

1999 W3C Recommendation: HTML 4.01

2000 W3C Recommendation: XHTML 1.0

2008 WHATWG HTML5 First Public Draft

2012 WHATWG HTML5 Living Standard

2014 W3C Recommendation: HTML5

2016 W3C Candidate Recommendation: HTML 5.1

2017 W3C Recommendation: HTML5.1 2nd Edition

2017 W3C Recommendation: HTML5.2

# HTML Basic Examples

## 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>`.

## Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

# 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:

```
<!DOCTYPE html>
```

# 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

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

# HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

## Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

# HTML Links

HTML links are defined with the `<a>` tag:

## Example

```
<a href="https://www.game.com">This is a link</a>
```

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:

## Example

```

```

# HTML Elements

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

## HTML Elements

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

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

Examples of some HTML elements:

<**h1**>My First Heading</**h1**>

<**p**>My first paragraph.</**p**>

Start tag	Element content	End tag
< <b>h1</b> >	My First Heading	</ <b>h1</b> >
< <b>p</b> >	My first paragraph.	</ <b>p</b> >
< <b>br</b> >	<i>none</i>	<i>none</i>

**Note:** Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

# 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

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

## 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:

```
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
```

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 are two other elements: `<h1>` and `<p>`:

```
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

The `<h1>` element defines a heading.

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

```
<h1>My First Heading</h1>
```

The `<p>` element defines a

paragraph. It has a start tag `<p>` and `</p>`:

an end tag `</p>`My first paragraph.

```
</p>
```

## Never Skip the End Tag

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

### Example

```
<html>
<body>

<p>This is a paragraph
<p>This is a paragraph

</body>
</html>
```

**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:

## Example

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

# 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 W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

# HTML Tag Reference

Tag	Description
<a href="#"><u>&lt;html&gt;</u></a>	Defines the root of an HTML document
<a href="#"><u>&lt;body&gt;</u></a>	Defines the document's body
<a href="#"><u>&lt;h1&gt; to &lt;h6&gt;</u></a>	Defines HTML headings

# HTML Attributes

HTML attributes provide additional information about HTML elements.

## 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:

### Example

```
<a href="https://www.game.com">Visit Game</a>
```

## The src Attribute

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

### Example

```

```

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.game.com/images/img_girl.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="img_girl.jpg"`. If the URL begins with a slash, it will be relative to the domain. Example: `src="/images/img_girl.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 `width` and `height` attributes, which specifies the width and the height of the image (in pixels):

### Example

```

```

## 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.

### Example

```

```

### Example

See what happens if we try to display an image that does not exist:

```

```

# The style Attribute

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

## Example

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

# The lang Attribute

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

Country codes can also be added to the language code in the `lang` attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

The following example specifies English as the language and United States as the country:

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

You can see all the language codes in our [HTML Language Code Reference](#).

---

# 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:

## Example

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

# 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

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

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

## Example

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

# HTML Styles

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

## Example

I am Red

I am Blue

I am Big

## 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:

```
<tagname style="property:value;">
```

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:

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

## Example

Set background color for two different elements:

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

# Text Color

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

## Example

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

# Fonts

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

## Example

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

# Text Size

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

## Example

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

# Text Alignment

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

## Example

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

# HTML Text Formatting

HTML contains several elements for defining text with a special meaning.

This is subscript and superscript

# HTML Formatting Elements

Formatting elements were designed to display special types of text:

- ② `<b>` - Bold text
- ② `<strong>` - Important text
- ② `<i>` - Italic text
- ② `<em>` - Emphasized text
- ② `<mark>` - Marked text
- ② `<small>` - Smaller text
- ② `<del>` - Deleted text
- ② `<ins>` - Inserted text
- ② `<sub>` - Subscript text
- ② `<sup>` - Superscript text

# HTML **<b>** and **<strong>** Elements

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

## Example

```
<b>This text is bold</b>
```

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

## Example

```
<strong>This text is important!</strong>
```

# HTML *<i>* and *<em>* Elements

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

**Tip:** The **<i>** tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

## Example

```
<i>This text is italic</i>
```

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

**Tip:** A screen reader will pronounce the words in **<em>** with an emphasis, using verbal stress.

## Example

```
<em>This text is emphasized</em>
```

# HTML <small> Element

The HTML <small> element defines smaller text:

## Example

```
<small>This is some smaller text.</small>
```

# HTML <mark> Element

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

## Example

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

# 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

```
<p>My favorite color is <del>blue</del> red.</p>
```

# HTML <ins> Element

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

## Example

```
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
```

# 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 H<sub>2</sub>O:

## Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

# 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<sup>[1]</sup>:

## Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

# HTML Formatting Elements

Formatting elements were designed to display special types of text:

- <b>** - Bold text
- <strong>** - Important text
- <i>** - Italic text
- <em>** - Emphasized text
- <mark>** - Marked text
- <small>** - Smaller text
- <del>** - Deleted text
- <ins>** - Inserted text
- <sub>** - Subscript text
- <sup>** - Superscript text

## HTML **<b>** and **<strong>** Elements

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

### Example

```
<b>This text is bold</b>
```

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

### Example

```
<strong>This text is important!</strong>
```

## HTML **<i>** and **<em>** Elements

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

**Tip:** The **<i>** tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

### Example

```
<i>This text is italic</i>
```

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

**Tip:** A screen reader will pronounce the words in `<em>` with an emphasis, using verbal

## Example

```
<em>This text is emphasized</em>
```

# HTML `<small>` Element

The HTML `<small>` element defines smaller text:

## Example

```
<small>This is some smaller text.</small>
```

# HTML `<mark>` Element

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

## Example

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

# 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

```
<p>My favorite color is <del>blue</del> red.</p>
```

# HTML <ins> Element

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

## Example

```
<p>My favorite color is <del>blue</del> <ins>red</ins>. </p>
```

# 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 H<sub>2</sub>O:

## Example

```
<p>This is <sub>subscripted</sub> text. </p>
```

# 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<sup>[1]</sup>:

## Example

```
<p>This is <sup>superscripted</sup> text. </p>
```

# HTML Text Formatting Elements

Tag	Description
<u><b>&lt;b&gt;</b></u>	Defines bold text
<u><em>&lt;em&gt;</em></u>	Defines emphasized text
<u><i>&lt;i&gt;</i></u>	Defines a part of text in an alternate voice or mood
<u><small>&lt;small&gt;</small></u>	Defines smaller text
<u><strong>&lt;strong&gt;</strong></u>	Defines important text
<u><sub>&lt;sub&gt;</sub></u>	Defines subscripted text
<u><sup>&lt;sup&gt;</sup></u>	Defines superscripted text
<u><ins>&lt;ins&gt;</ins></u>	Defines inserted text
<u><del>&lt;del&gt;</del></u>	Defines deleted text

<a href="#"><u>&lt;mark&gt;</u></a>	Defines marked/highlighted text
-------------------------------------	---------------------------------

# HTML Formatting Elements

Formatting elements were designed to display special types of text:

- ② [<b>](#) - Bold text
- ② [<strong>](#) - Important text
- ② [<i>](#) - Italic text
- ② [<em>](#) - Emphasized text
- ② [<mark>](#) - Marked text
- ② [<small>](#) - Smaller text
- ② [<del>](#) - Deleted text
- ② [<ins>](#) - Inserted text
- ② [<sub>](#) - Subscript text
- ② [<sup>](#) - Superscript text

## HTML **<b>** and **<strong>** Elements

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

### Example

```
<b>This text is bold</b>
```

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

### Example

```
<strong>This text is important!</strong>
```

# HTML *<i>* and *<em>* Elements

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

**Tip:** The *<i>* tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

## Example

```
<i>This text is italic</i>
```

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

**Tip:** A screen reader will pronounce the words in *<em>* with an emphasis, using verbal stress.

## Example

```
<em>This text is emphasized</em>
```

# HTML *<small>* Element

The HTML *<small>* element defines smaller text:

## Example

```
<small>This is some smaller text.</small>
```

# HTML *<mark>* Element

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

## Example

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

# 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

```
<p>My favorite color is <del>blue</del> red.</p>
```

# HTML <ins> Element

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

## Example

```
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
```

# 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 H<sub>2</sub>O:

## Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

# 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

```
<p>This is <sup>superscripted</sup> text.</p>
```

# HTML Text Formatting Elements

Tag	Description
<code>&lt;b&gt;</code>	Defines bold text
<code>&lt;em&gt;</code>	Defines emphasized text
<code>&lt;i&gt;</code>	Defines a part of text in an alternate voice or mood
<code>&lt;small&gt;</code>	Defines smaller text
<code>&lt;strong&gt;</code>	Defines important text
<code>&lt;sub&gt;</code>	Defines subscripted text

<sup> Defines superscripted text

<ins> Defines inserted text

<del> Defines deleted text

<mark> Defines marked/highlighted text

## HTML Comments

HTML comments are not displayed in the browser, but they can help document your HTML source code.

## HTML Comment Tag

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

<!-- Write your comments here -->

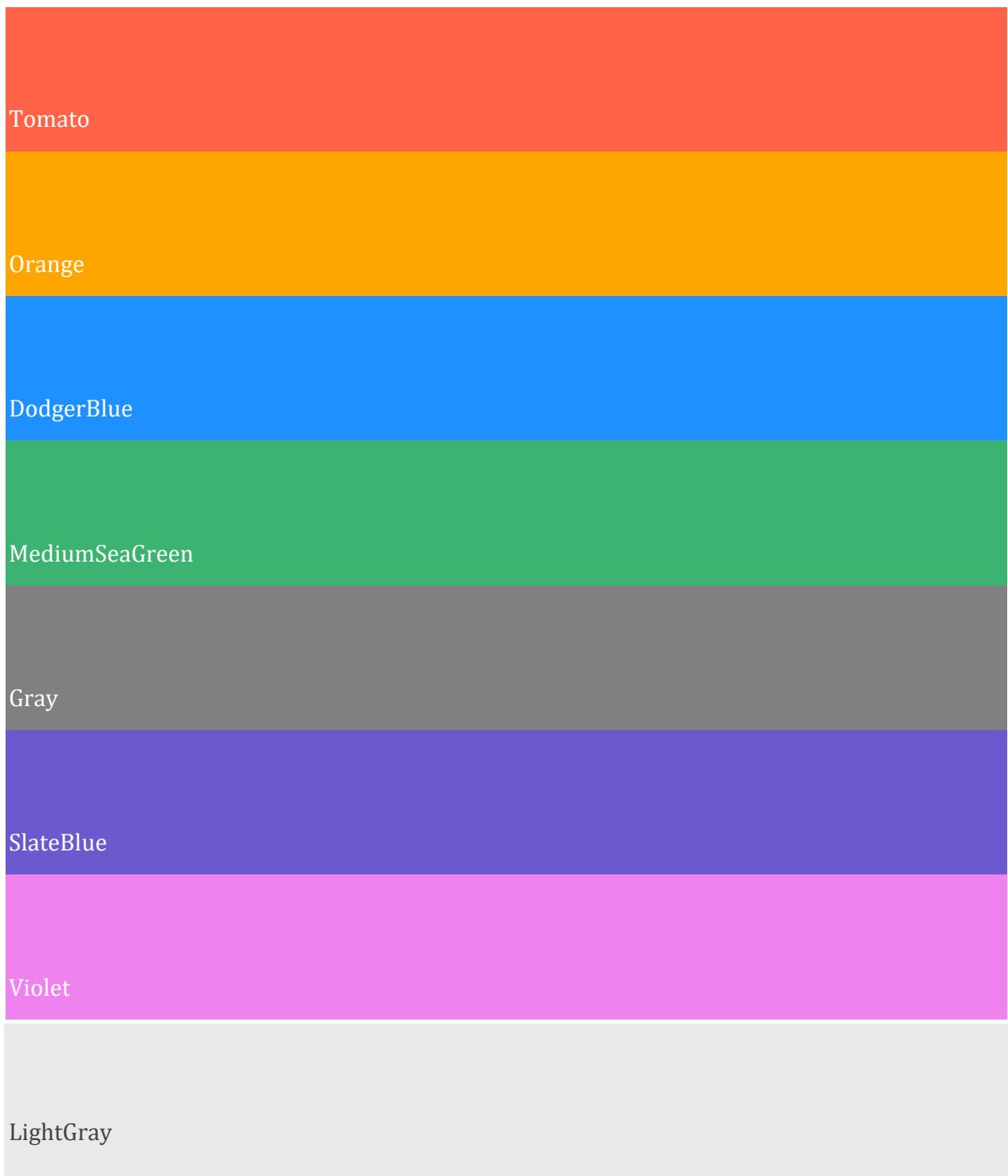
Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

## HTML Colors

HTML colors are specified with predefined color names, or with RGB

## Color Names

In HTML, a color can be specified by using a color name:



# Background Color

You can set the background color for HTML elements:

## Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

### Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1><p  
style="background-color:Tomato;">Lorem ipsum...</p>
```

## Text Color

You can set the color of text:

### Hello World

Lore*m* ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

### Example

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

## Border Color

You can set the color of borders:

# Hello World

# Hello World

# Hello World

## Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

## Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values.

The following three `<div>` elements have their background color set with RGB, HEX, and HSL values:

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

The following two `<div>` elements have their background color set with RGBA and HSLA values, which adds an Alpha channel to the color (here we have 50% transparency):

## Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

# HTML Links

Links are found in nearly all web pages. Links allow users to click their way from page to page.

## HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

## HTML Links - Syntax

The HTML `<a>` tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

## Example

This example shows how to create a link to Game.com:

```
<a href="https://www.game.com/">Visit Game.com!</a>
```

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

- ② An unvisited link is underlined and blue
- ② A visited link is underlined and purple
- ② 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:

- ② **\_self** - Default. Opens the document in the same window/tab as it was clicked
- ② **\_blank** - Opens the document in a new window or tab
- ② **\_parent** - Opens the document in the parent frame
- ② **\_top** - Opens the document in the full body of the window

### Example

Use target="\_blank" to open the linked document in a new browser window or tab:

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

## Absolute URLs vs. Relative URLs

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

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

### Example

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>
```

```
<h2>Relative URLs</h2>
```

```
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

## HTML Links - Use an Image as a Link

To use an image as a link, just put the `<img>` tag inside the `<a>` tag:

### Example

```
<a href="default.asp">

</a>
```

## Link to an Email Address

Use `mailto:` inside the `href` attribute to create a link that opens the user's email program (to let them send a new email):

### Example

```
<a href="mailto:someone@example.com">Send email</a>
```

## Button as a Link

To use an HTML button as a link, you have to add some JavaScript code.

JavaScript allows you to specify what happens at certain events, such as a click of a button:

### Example

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

## Link Titles

The `title` attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

### Example

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our
HTML Tutorial</a>
```

# More on Absolute URLs and Relative URLs

## Example

Use a full URL to link to a web page:

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

## Example

Link to a page located in the html folder on the current web site:

```
<a href="/html/default.asp">HTML tutorial</a>
```

## Example

Link to a page located in the same folder as the current page:

```
<a href="default.asp">HTML tutorial</a>
```

# HTML Link Colors

By default, a link will appear like this (in all browsers):

- ② An unvisited link is underlined and blue
- ② A visited link is underlined and purple
- ② An active link is underlined and red

You can change the link state colors, by using CSS:

## Example

Here, an unvisited link will be green with no underline. A visited link will be pink with no underline. An active link will be yellow and underlined. In addition, when mousing over a link (a:hover) it will become red and underlined:

```
<style>
a:link {
color: green;
background-color: transparent;
text-decoration: none;
}

a:visited {
```

```
color: pink;  
background-color: transparent;  
text-decoration: none;  
}  
  
a:hover {  
color: red;  
background-color: transparent;  
text-decoration: underline;  
}  
  
a:active {  
color: yellow;  
background-color: transparent;  
text-decoration: underline;  
}  
/style>
```

# HTML Images

Images can improve the design and the appearance of a web page.

## Example

```

```

## Example

```

```

## Example

```

```

# HTML Images Syntax

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

## Syntax

```

```

# The src Attribute

The required **src** attribute specifies the path (URL) to the image.

**Note:** When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon and the **alt** text are shown if the browser cannot find the image.

## Example

```

```

# The alt Attribute

The required **alt** attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the **src** attribute, or if the user uses a screen reader).

The value of the **alt** attribute should describe the image:

## Example

```

```

If a browser cannot find an image, it will display the value of the **alt** attribute:

## Example

```

```

# Image Size - Width and Height

You can use the `style` attribute to specify the width and height of an image.

## Example

```

```

Alternatively, you can use the `width` and `height` attributes:

## Example

```

```

# Animated Images

HTML allows animated GIFs:

## Example

```

```

# Image as a Link

To use an image as a link, put the `<img>` tag inside the `<a>` tag:

## Example

```
<a href="default.asp">

</a>
```

# Image Floating

Use the CSS `float` property to let the image float to the right or to the left of a text:

## Example

```
<p>
The image will float to the right of the text.</p>
```

```
<p>  
The image will float to the left of the text.</p>
```

# HTML Image Maps

With HTML image maps, you can create clickable areas on an image.

## Image Maps

The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.

Try to click on the computer, phone, or the cup of coffee in the image below:



### Example

Here is the HTML source code for the image map above:

```

```

```
<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>
```

## The Image

The image is inserted using the `<img>` tag. The only difference from other images is that you must add a `usemap` attribute:

```

```

The `usemap` value starts with a hash tag `#` followed by the name of the image map, and is used to create a relationship between the image and the image map.

**Tip:** You can use any image as an image map!

## Create Image Map

Then, add a `<map>` element.

The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:

```
<map name="workmap">
```

The `name` attribute must have the same value as the `<img>`'s `usemap` attribute .

## The Areas

Then, add the clickable areas.

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

You must also define some coordinates to be able to place the clickable area onto the image.

## Shape="rect"

The coordinates for **shape="rect"** in pairs, one for the x-axis and one for the y-axis.

So, the coordinates **34,44** is located 34 pixels from the left margin and 44 pixels from the top:



The coordinates **270,350** is located 270 pixels from the left margin and 350 pixels from the top:



Now we have enough data to create a clickable rectangular area:

## Example

```
<area shape="rect" coords="34, 44, 270, 350" href="computer.htm">
```

This is the area that becomes clickable and will send the user to the page "computer.htm":



## **Shape="circle"**

To add a circle area, first locate the coordinates of the center of the circle:

337,300



Then specify the radius of the circle:

44 pixels



Now you have enough data to create a clickable circular area:

## Example

```
<area shape="circle" coords="337, 300, 44" href="coffee.htm">
```

This is the area that becomes clickable and will send the user to the page "coffee.htm":



# 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:

```
<p style="background-image: url('img_girl.jpg');">
```

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

## Example

Specify the background image in the `<style>` element:

```
<style>
p {
background-image: url('img_girl.jpg');
}
</style>
```

# 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:

```
<style>
body {
background-image: url('img_girl.jpg');
}
</style>
```

# Background Repeat

If the background image is smaller than the element, the image will repeat itself, horizontally and vertically, until it reaches the end of the element:

## Example

```
<style>
body {
background-image: url('example_img_girl.jpg');
```

```
}
```

```
</style>
```

To avoid the background image from repeating itself, set the `background-repeat` property to `no-repeat`.

## Example

```
<style>
body {
background-image: url('example_img_girl.jpg');
background-repeat: no-repeat;
}
</style>
```

# HTML Tables

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

## Example

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria

Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

## Define an HTML Table

A table in HTML consists of table cells inside rows and columns

### Example

A simple HTML table:

```
<table>
<tr>
<th>Company</th>
<th>Contact</th>
<th>Country</th>
</tr>
<tr>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Germany</td>
</tr>
<tr>
    <td>Centro comercial Moctezuma</td>
<td>Francisco Chang</td>
<td>Mexico</td>
</tr>
</table>
```

# able Cells

Each table cell is defined by a `<td>` and a `</td>` tag.

`td` stands for table data.

Everything between `<td>` and `</td>` are the content of the table cell.

## Example

```
<table>
<tr>
<td>Emil</td>
<td>Tobias</td>
<td>Linus</td>
</tr>
</table>
```

**Note:** table data elements are the data containers of the table.

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

# Table Rows

Each table row starts with a `<tr>` and end with a `</tr>` tag.

`tr` stands for table row.

## Example

```
<table>
<tr>
  <td>Emil</td>
  <td>Tobias</td>
  <td>Linus</td>
</tr>
<tr>
  <td>16</td>
  <td>14</td>
  <td>10</td>
</tr>
</table>
```

# Table Headers

Sometimes you want your cells to be headers, in those cases use the `<th>` tag instead of the `<td>` tag:

## Example

Let the first row be table headers:

```
<table>
<tr>
<th>Person 1</th>
<th>Person 2</th>
<th>Person 3</th>
</tr>
<tr>
<td>Emil</td>
<td>Tobias</td>
<td>Linus</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>10</td>
</tr>
</table>
```

By default, the text in `<th>` elements are bold and centered, but you can change that with CSS.

# HTML Table Borders

HTML tables can have borders of different styles and shapes.

## How To Add a Border

When you add a border to a table, you also add borders around each table cell:

--	--	--


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

## Example

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

# HTML Table Padding & Spacing

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

With Padding		
hello	hello	hello
hello	hello	hello
hello	hello	hello

With Spacing		
hello	hello	hello
hello	hello	hello
hello	hello	hello

# HTML Table - Cell Padding

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

By default the padding is set to 0.

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

## Example

```
th, td {  
padding: 15px;  
}
```

To add padding only above the content, use the `padding-top` property.

And the others sides with the `padding-bottom` , `padding-left` , and `padding-right` properties:

## Example

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

# HTML Table - Cell Spacing

Cell spacing is the space between each cell.

By default the space is set to 2 pixels.

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

## Example

```
table {  
border-spacing: 30px; }
```

# HTML Table Colspan & Rowspan

`colspan` attribute represents the number of columns to span.

**Note:** The value of the

## HTML Table - Rowspan

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

### Example

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

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

## HTML Lists

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

### Example

An unordered HTML list:

- ❑ Item
- ❑ Item
- ❑ Item
- ❑ Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

## 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

#### `<ul> Ordered HTML List`

An ordered list starts with the `<ol>` tag. Each list item starts with `<li>` tag. The list items will be marked with numbers by default:

### Example

```
<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

```
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

## 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:

## Example

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

## HTML List Tags

Tag	Description
<a href="#"><u>&lt;ul&gt;</u></a>	Defines an unordered list
<a href="#"><u>&lt;ol&gt;</u></a>	Defines an ordered list
<a href="#"><u>&lt;li&gt;</u></a>	Defines a list item
<a href="#"><u>&lt;dl&gt;</u></a>	Defines a description list
<a href="#"><u>&lt;dt&gt;</u></a>	Defines a term in a description list
<a href="#"><u>&lt;dd&gt;</u></a>	Describes the term in a description list

## HTML Unordered Lists

The HTML `<ul>` tag defines an unordered (bulleted) list.

## 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

```
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

## 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:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square

none	The list items will not be marked
------	-----------------------------------

## Example - Disc

```
<ul style="list-style-type:disc;">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

## Example - Circle

```
<ul style="list-style-type:circle;">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

## Example - Square

```
<ul style="list-style-type:square;">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

## Example - None

```
<ul style="list-style-type:none;">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

# Nested HTML Lists

Lists can be nested (list inside list):

## Example

```
<ul>
<li>Coffee</li>
```

```
<li>Tea  
<ul>  
    <li>Black tea</li>  
    <li>Green tea</li>  
</ul>  
</li>  
<li>Milk</li>  
</ul>
```

## HTML Ordered Lists

<ol>

The HTML `<ol>` tag defines an ordered list. An ordered list can be numerical or alphabetical.

## Ordered HTML List

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

The list items will be marked with numbers by default:

### Example

```
<ol>  
<li>Coffee</li>  
<li>Tea</li>  
<li>Milk</li>  
</ol>
```

## Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)

type="A" The list items will be numbered with uppercase letters

type="a" The list items will be numbered with lowercase letters

type="I" The list items will be numbered with uppercase roman numbers

type="i" The list items will be numbered with lowercase roman numbers

## Numbers:

```
<ol type="1">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

## Uppercase Letters:

```
<ol type="A">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

## Lowercase Letters:

```
<ol type="a">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

## Uppercase Roman Numbers:

```
<ol type="I">
<li>Coffee</li>
<li>Tea</li>
```

```
<li>Milk</li>
</ol>
```

## Lowercase Roman Numbers:

```
<ol type="i">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

# HTML Audio

The HTML `<audio>`

element is used to play an audio file on a web page.

## The HTML `<audio>` Element

To play an audio file in HTML, use the `<audio>` element:

### Example

```
<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

## HTML Audio - How It Works

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

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

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

# HTML <audio> Autoplay

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

## Example

```
<audio controls autoplay>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

Add `muted` after `autoplay` to let your audio file start playing automatically (but muted):

## Example

```
<audio controls autoplay muted>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

# HTML Video

The HTML `<video>`

element is used to show a video on a web page.

## Example

Courtesy of [Big Buck Bunny](#):

# The HTML <video> Element

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

## Example

```
<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

## How it Works

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

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

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

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

## HTML <video> Autoplay

To start a video automatically, use the `autoplay` attribute:

## Example

```
<video width="320" height="240" autoplay>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Add `muted` after `autoplay` to let your video start playing automatically (but muted):

## Example

```
<video width="320" height="240" autoplay muted>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

# HTML Forms

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

## Example

First name:

Last name:

## The <form> Element

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

```
<form>
```

```
  . . .
```

```
</form>
```

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

## The <input> Element

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

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

Here are some examples:

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

## Text Fields

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

### Example

A form with input fields for text:

```
<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>
```

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

name:

Last name:

## 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.

### Example

A form with radio buttons:

```
<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>
```

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

Choose your favorite Web language:

HTML

CSS

JavaScript

## Checkboxes

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

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

### Example

A form with checkboxes:

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

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

I have a bike

I have a car

I have a boat

## The 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:

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

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

First name:

Last name:

Submit

---

## 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:

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

# HTML Form Attributes

# The Action Attribute

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:

## Example

On submit, send form data to "action\_page.php":

```
<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>
```

**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:

Value	Description
_blank	New blank window

_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

The default value is `_self` which means that the response will open in 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 be used when submitting the form data.

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

The default HTTP method when submitting form data is GET.

## 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">
```

# HTML Input Types

# HTML Input Types

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

```
② <input type="button">
② <input type="checkbox">
② <input type="color">
② <input type="date">
② <input type="datetime-local">
② <input type="email">
② <input type="file">
② <input type="hidden">
② <input type="image">
② <input type="month">
② <input type="number">
② <input type="password">
② <input type="radio">
② <input type="range">
② <input type="reset">
② <input type="search">
② <input type="submit">
② <input type="tel">
② <input type="text">
② <input type="time">
② <input type="url">
② <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:

### 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>
```

»

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>
```

»

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>
```



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

First name:

Last name:

Submit

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>
```

»

## 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>
```

»

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

First name:

Last name:

Submit	Reset
--------	-------

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>
```



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>
```

»

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!">
```

»

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>
```

»

## 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>
```

»

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>
```

»

# 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>
```

»

# 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>
```

»

# 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>
```

»

# 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>
```

»

# 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>
```

»

# 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>
```

»

# 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 (only applicable when 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 characters 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>
```

»

# 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>
```

»

# 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>
```

»

# 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>
```

»

# 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>
```

»

## 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>
```

»

## 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.

# HTML Input Attributes

## Example

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

»

## 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>
```

## 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>
```

## 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>
```

## 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>
```

## 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>
```

## 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>
```

## 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>
```

## 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** attribute to describe the pattern to help the user. Tip:

Learn more about [regular expressions](#) 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>
```

## 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>
```

## 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>
```

# 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>
```

# 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>
```

# the height and width Attributes

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

## 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>
```

## 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>
```

## 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>
```