LEARN HTML

- CLICK TO JUMP TO THE PAGE -

1. Learn the basics of html

<!DOCTYPE>: Declaration

All HTML documents must start with it.
Is not a tag, but a declaration about what type of document to expect. Currently we are using the one for **HTML 5**.

Back to index

```
<!DOCTYPE html>
```

<html>: Element

Represents the root of an HTML document and it is the parent container of all the other HTML elements. The **lang attribute** should always be included to assist search engines and browsers to know the language of the webpage.

Back to index

```
<html lang="en"></html>
```

<head>: Element

It's a container for metadata and it is placed between the **<html>** tags.

Back to index

```
<html lang="en">
  <head></head>
</html>
```

<meta>: Element

It is a self-closing tag situated inside the **<head>** that defines the metadata for an HTML document. This metadata will not be displayed in the browser.

Back to index

```
<head>
  <meta charset="UTF-8">
</head>
```

<meta>: attribute charset

Specifies the character set encoding for the HTML document.

Back to index

```
<meta charset="UTF-8">
```

<meta>: attribute name

Specifies the type of metadata (application-name, author, description, generator, keywords, viewport).

Back to index

```
<meta name="application-name" content="My app name">
```

<meta>: attribute content

Specifies the value of the metadata type it goes with.

Back to index

```
<meta name="application-name" content="My app name">
```

<meta>: attribute name “application-name”

Specifies the name of the web or application, and it is used for SEO purposes.

Back to index

```
<meta name="application-name" content="My app name">
```

<meta>: attribute name “author”

Defines the author of the webpage so users and search engines can know who wrote the page.

Back to index

```
<meta name="author" content="Jane Doe">
```

<meta>: attribute name “description”

It is a short summary of what the page is about.

Back to index

```
<meta name="description" content="All the information about the trees in our planet.">
```

<meta>: attribute name “generator”

Identifies the software that generated the app.

Back to index

```
<meta name="generator" content="WordPress 3.0.1">
```

<meta>: attribute name “keywords”

Defines words relevant to the page content for SEO reasons.

Back to index

```
<meta name="keywords" content="plants, types of trees">
```

<meta>: attribute name “viewport”

Specifies the initial size of the viewport:

- **width ->** sets the width of the viewport. For responsiveness the best is to set the width to be the device-width, but a number can be passed as well to symbolise the pixels it will occupy.
- **height ->** sets the height of the viewport. It works the same as the width, but it is not used by any browser.
- **initial-scale ->** sets the initial zoom level of the webpage.
- **maximum-scale ->** defines the maximal zoom from 0.0 to 10.0.
- **minimum-scale ->** defines the minimal zoom from 0.0 to 10.0.
- **user-scalable ->** defines if the user is able to zoom or not by using “yes” and “no”. For accessibility reasons, it is not recommended to disable the zoom.
- **viewport-fit ->** defines how the viewport adapts to the display.
    - **auto** doesn’t affect the initial layout of the viewport.
    - **contain** scales the viewport to fit the largest rectangle inscribed within the display.
    - **cover** scales the viewport to fill the device display.

All the previous properties can be added for SEO and accessibility reasons, but the main reason why it should always be added is for responsiveness. Each device has a different screen size, so it is really important to add at least the **width** and the **initial-scale** to make sure the viewport size will adapt to the size of the device.

Back to index

```
<meta name="viewport" content="width=device-width, initial-scale=1.0 , maximum-scale=10.0, minimum-scale=0.0, user-scalable=yes, viewport-fit=auto">
```

<meta>: attribute name “color-scheme”

Specifies one or more color schemes which the document is compatible with, and the browser will use it in tandem with the user’s browser or device settings to determine what color to use.
In the content attribute the scheme specification is passed:

- **normal ->** the document is unaware of color schemes.
- **[light | dark] + ->** indicates one or more schemes supported by the browser.
- **only light ->** indicates the document only supports light mode.

Back to index

```
<meta name=“color-scheme” content=“normal”>
```

<meta>: other metadata names

There are many more metadata names that have not been formally accepted yet, but some of them are quite a common practice.
In this **mdn web docs** you can further explore this other metadata.

Back to index

```
<meta name=“ robots ”>
```

<title>: Element

It defines the title of the document and it is shown in the browser’s tab. It is important for SEO and accessibility reasons.
You cannot have more than one title per HTML document!

Back to index

```
<head>
  <title>Document</title>
</head>
```

<body>: Element

It defines the document’s body, the part that it is displayed in the browser.

Back to index

```
<html  lang=”en”>
  <body></body>
</html>
```

: Element

It defines a division or section in the HTML document and it is a parent container for elements that should be grouped together.
It is useful for styling but for the structure it is better to use semantic tags instead.

Back to index

```
<div></div>
```

Global attributes in html elements

Inside the html elements, **attributes** can be added within the **opening tag** to modify the behaviour or to locate and target that specific element.
There are **attributes** that are **specific to an html element** and can only be used within that element.
There are also **global attributes** that **can be used in all of the html elements**.
Some global attributes:

- **id** -> specifies a unique id for an html element. It can be used to target or locate the element.
- **class** -> specifies one or more classnames for an element. This can be used to target specific elements to style them with CSS.
- **contenteditable** -> specifies if the content of the element can be edited or not.
- **style** -> specifies a specific CSS inline style for the element.
- **title** -> specifies extra information about an element.

Back to index

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="author" content="Jane Doe">
    <meta name="viewport"
    content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <div
          id="section1"
          class="sections"
          contenteditable="true"
    ></div>
    <div
          id="section2"
          class="sections"
          contenteditable="false"
          title="The second section"
          style="color:red;"
    ></div>
  </body>
</html>
```

2. Learn semantic HTML for structure (for SEO and accessibility)

<header>: Element

It's a container for introductory content or navigational links.
There can be several headings in one HTML document, but they cannot be placed inside the **<footer>**, the **<address>** or another **<header>**.

Back to index

```
<header id="header" class="header" ></header>
```

<nav>: Element

It defines a set of navigation links.
Not all navigation links should be inside a **<nav>**, only the major ones.

Back to index

```
<nav id="navbar-right" class="navbar"></nav>
<nav id="navbar-left" class="navbar"></nav>
```

<main>: Element

It defines the main content of a document. It should not hold any other content that is repeated across documents, such as sidebars. Also, there must not be more than one **<main>** per html and it can't be a child to **<article>**, **<aside>**, **<footer>**, **<header>** or **<nav>**.

Back to index

```
<main id="main-content" class="main"></main>
```

<article>: Element

It specifies self-contained content, independent of the other content in the document as it could be taken out and inserted somewhere else.
It doesn't add any specific styling, but gives semantic value for SEO and accessibility.

Back to index

```
<article id="article1" class="article"></article>
<article id="article2" class="article"></article>
<article id="article3" class="article"></article>
<article id="article4" class="article"></article>
```

<section>: Element

It defines a section in a document. The content inside is connected to the content on other sections and it should have a heading.
It doesn't add any specific styling in the browser, it just gives semantic value for SEO and accessibility.

Back to index

```
<section id="section1" class="section"></section>
<section id="section2" class="section"></section>
<section id="section3" class="section"></section>
<section id="section4" class="section"></section>
```

<aside>: Element

It defines content that is less important inside other content. It is usually used for sidebars, areas that add complementary but nonessential information.

Back to index

```
<aside id="side-navbar" class="aside"></aside>
<aside id="side-explanation" class="aside" ></aside>
<aside id="extraInfo" class="aside"></aside>
```

<footer>: Element

It defines the content at the bottom of the page. It normally includes information about copyright, contact and navigation links.

Back to index

```
<footer id="footer" class="footer"></footer>
```

3. Learn semantic HTML for text (they add formatting)

<h1>: Element

It defines the top level heading and there should be only one per page because of SEO reasons. The browser gives it a default styling.

Back to index

```
<h1>Heading 1</h1>
```

<h2> to <h6>: Element

It defines the subheadings of the page depending on the importance. There can be multiple headings of the same level in a page.
The default styling is bold and the text size varies, with the **<h2>** being the bigger one and the **<h6>** the smallest.

Back to index

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

<p>: Element

It defines a standalone paragraph of text.
It adds by default some margins before and after each <p> element.

Back to index

```
<p>Lorem</p>
```

<a>: Element

It defines a hyperlink used to link from one page to another.

Back to index

```
<a title="Google" href="www.google.com"
target="_blank">Google</a>
```

<a>: attribute title

Defines the name or the description of the resource where the link redirects. It becomes visible when the mouse hovers over the link and it should be added for SEO and accessibility.

Back to index

```
<a
   title="About us"
>About us</a>
```

<a>: attribute href

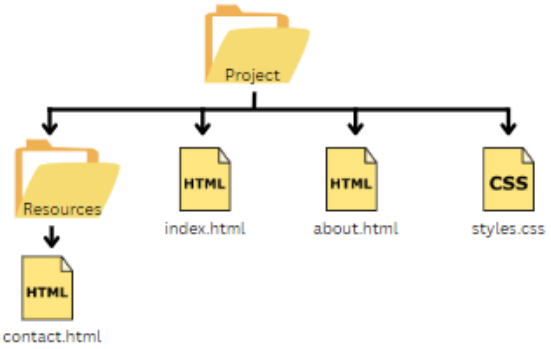
The **href attribute** specifies the address of the link's destination, and can accept as its value a **relative path**, an **id** or a **url**.

Back to index

```
<a href="www.google.com"  title="Google">Google</a>
```

<a>: attribute href - Relative path

Linking local documents via the **href attribute** is done using **relative paths**. **Relative paths** describe the location of a file in relation to another file. For that, it is important to look at the hierarchy of the files and folders in the file system.



For an anchor tag in **index.html**:

Pointing at files in the same folder or deeper, is done with a `**./**`, followed by the name of the file, ie. "**./about.html**". If the file is nested inside of one or more folders, simply add all folders separated by slashes until the file is reached, ie. "**./resources/contact.html**".

If a file is at another location, traversing up one folder is done with `**../**`. You can go up as many folders as needed by chaining the command. As an example, the relative path `**../../my-other-project/src/faq.html**` goes up two folders, then into the folder `**my-other-project**`, that contains an `**src**` folder, and inside it the desired `**faq.html**` file.

Back to index

index.htm

```
<a href="./about.html" title="About">About</a>
```

```
<a href="./resources/contact.html"
title="Contact">Contact</a>
```

contact.htm

```
<a href="../contact.html" title="Contact">Contact</a>
```

<a>: attribute href - URL

The href passes the url of the place to redirect with the link.

Back to index

```
<a href="www.google.com"  title="Google">Google</a>
```

<a>: attribute href - ID

The href passes the value of the **id attribute** of the HTML element it wants to redirect to preceded by the symbol **#**. By using an id in the href, you can redirect only to another element within the same document.

Back to index

```
<a href="#contact" title="Contact" >Contact</a>
```

<a>: attribute target

The **target** attribute determines if the link will open within the same tab or in a new tab:

- **_blank** -> opens the link in a new tab or window depending on the browser's settings.
- **_self** -> opens the link in the current tab. It is the default.
- **_parent** -> opens the link in the parent frame.
- **_top** -> opens the link in the full body of the window.

Back to index

```
<a
   title="About us"
   href="./resources/aboutus.html"
   target="_blank"
>About us</a>
```

: Element

It creates an ordered list.
It adds some margins and the list styling with numbers or letters.

Back to index

```
<ol id="ordered-list" class="list"></ol>
```

: Element

It creates an unordered list.
It adds some margins and the list styling with bullet points.

Back to index

```
<ul id="unordered-list" class="list">Document</ul>
```

: Element

It creates an item inside a list, so it always goes inside the **<ul>** or **<ol>** elements.

Back to index

```
<li id="item1" class="listItem">Item 1</li>
<li id="item2" class="listItem">Item 2</li>
<li id="item3" class="listItem">Item 3</li>
```

<q>: Element

It defines a short quotation.

Back to index

```
<q id="quotation1" class="quotation"
>Our quote</q >
```

: Element

It defines emphasised text.
It is typically displayed in italic and the screen reader will stress the word to display the emphasis.
The **<i>** tag creates the same italic effect without adding emphasis to it.

Back to index

```
<em id="emphasis" class="emphasis">Important</em>
```

: Element

It defines a text of high importance.
It is normally displayed in bold. If there is no need to imply importance, the **<b>** tag can be used to achieve the same bold styling.

Back to index

```
<strong id="bold-important1"
class="bold-important">Bold</strong>
<b id="bold1" class="bold">Bold</b>
```

<code>: Element

It defines a piece of computer's code.
The content inside is displayed in the default monospace font of the browser.

Back to index

```
<code id="code1" class="code">button</code>
```

4. Learn how to create a table

<table>: Element

It creates a table.
It can use the border attribute to give a border to all the elements inside the **<table>** tags.
It has an opening and a closing tag that surround all the other elements of the table.

Back to index

```
<table border="1"></table>
```

<thead>: Element

It defines where inside the **<table>** tags the titles are going to be.
It is not necessary but adds semantic value.

Back to index

```
<table border="1">
  <thead></thead>
</table>
```

<tbody>: Element

It defines where the data is going to be inside the **<table>** tags.
It is not necessary to use but adds semantic value.

Back to index

```
<table border="1">
  <thead></thead>
  <tbody></tbody>
</table>
```

<tr>: Element

It creates a row in the table.
It can go directly inside the **<table>** tags or inside the **<thead>** and **<tbody>**.
There needs to be as many **<tr>** as rows are needed in the table.

Back to index

```
  <thead>
     <tr></tr>
  </thead>
  <tbody>
      <tr></tr>
  </tbody>
```

<th>: Element

It creates a cell in the **<tr>** and gives it a bolder styling to define it as the header of the column.
It is used inside the **<tr>** tags that are inside the **<thead>** if there is one. If there is none, it is used inside a row whenever the styling of a title wants to be added to the text of the cell.

Back to index

```
<tr>
  <th>Title</th>
</tr>
```

<td>: Element

It creates a cell in the **<tr>** but gives no special styling to it.
It is used inside the **<tr>** tags that are inside the **<tbody>** if there is one. If there is none, it is used inside a row to display text inside a cell with no special styling to it.

Back to index

```
<tr>
  <td>Data</td>
</tr>
```

5. Learn how to create a form

<form>: Element
It creates a form element.
Nested inside this element are all the other elements of the form (fieldsets, labels, inputs, etc). The method attribute determines the http method used when sending the data (POST or GET).

Back to index

```
<form></form>
```

<fieldset>: Element

It defines a group of inputs and labels that refer to the same type of information and styles them with a border around them.
There can be multiple fieldsets inside a form.

Back to index

```
<fieldset></fieldset>
```

<legend>: Element

It defines the title or caption for a specific fieldset. This refers to the type of information the inputs of that fieldset are collecting.
It appears in between the border of the fieldset on the top left corner and it is styled bold.

Back to index

```
<form>
  <fieldset>
    <legend>Personal Info</legend>
  </fieldset>
  <fieldset>
    <legend>Personal Info</legend>
  </fieldset>
</form>
```

<label>: Element

It defines a caption for an input.
The **for attribute** in the label connects it with its input by passing the input's id. This is very useful for accessibility, as the connection allows for screen readers to read the label in which the user is currently on or to activate the input by clicking its label.

Back to index

```
<label for="fname" >First name</label>
<input id="fname"  />
```

<input>: Element

It creates an input field to enter data.
It can be displayed in different ways depending on the type attribute it's given to it.
Some of the most important attributes:

- **Type** -> specifies the type of info the input is supposed to collect and what input needs to be displayed for that.
- **Id** -> helps to connect with the label of the input.
- **Name** -> it specifies the name of an input. It is necessary for the value of the input to be submitted with the form.
- **Value** -> it defines the default value of the input but its behaviour will depend on the **type attribute** (in most types it will be overwritten but not all).
- **Placeholder** -> specifies a short hint that describes the expected value.
- **Required** -> boolean value. When present it specifies that the input must be filled out to be able to submit the form.
- **Readonly** -> boolean value. When present, it specifies that an input field is read-only and the user will not be able to interact with it.
- **Disabled** -> boolean value. When present, it specifies an input element is disabled.
- **Min and max** -> they specify the minimum and maximum values of number or date related input types.
- **Autocomplete** -> boolean value. When present it specifies if a certain input should be autocomplete enabled.

It is a self-closing tag.

Back to index

```
<input
  type="text"
  id="fname"
  name="fname"
  value="John"
  placeholder="John"
  required
  autocomplete
 />

<input
  type="number"
  id="favourite-number"
  name="favourite-number"
  value="5"
  placeholder="20"
  min="1"
  max="100"
  readonly
/>

<input
  type="submit"
  id="submit"
  name="submit"
  value="Send"
  disabled
/>
```

<input>: type text

It defines a single line text field. The default width is 20 characters. It is the default type.

Back to index

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

<input type="text" id="fname" name="fname"
value="John" placeholder="John"  required
autocomplete  />
```

<input>: type email

It defines a field for an email address.
The value is validated automatically to ensure that the data entered is a properly formatted email.
The **multiple attribute** can be used so an input can accept several email addresses.

Back to index

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

<input type="email" id="email" name="email"
placeholder="email" />
```

<input>: type password

It defines a password field.
It provides a way for the user to securely enter a password. The text is obscured so it can't be read by replacing each character with an asterisk or a dot.

Back to index

```
<label for="password" >Password</label>

<input type="password" id="password"
name="password" placeholder="password" />
```

<input>: type tel

It defines a field for entering a telephone number.

Back to index

```
<label for="phone">Phone</label>

<input type="tel" id="phone" name="phone"
placeholder="phone" />
```

<input>: type number

It defines a field for entering a number
The **min** and **max attribute** can be used with it.

Back to index

```
<label for="favourite-number">Which one is your
favourite number?</label>

<input type="number" id="favourite-number"
name="favourite-number" value="5" placeholder="20"
min="1" max="100" />
```

<input>: type date

It fines a field for a date picker that includes and displays the day, the month and the year.
The **min, max and value attribute** can be used with it by entering “year-month-day”.

Back to index

```
<label for="birthdate">When is your birthdate?</label>

<input type="date" id="birthdate" name="birthdate"
value="1990-01-01" min="1930-01-01" max="2023-01-01"
/>
```

<input>: type time

It defines a field used for entering a time.
The **min** and **max** can be used with it.

Back to index

```
<label for="opening-hours">Which are the opening
hours?</label>

<input type="time" id="opening-hours"
name="opening-hours" min="09:00" max="18:00" />
```

<input>: type week

It defines an input field that allows the entry of the week number of a year and the year.
The **min**, **max** and **value attributes** can be used in it by entering “year-weeknumber”.

Back to index

```
<label for="week">Week of the year</label>

<input type="week" id="week" name="week"
min="1990-W01" max="2023-W52" value="2000-W45" />
```

<input>: type month

It defines a field that allows the month and the year to be entered.
The **min**, **max** and **value attributes** can be used in it by entering year-month.

Back to index

```
<label for="month">Start month:</label>

<input type="month" id="month" name="month"
min="1990-01" max="2023-12" value="2015-03" />
```

<input>: type range

It defines a field for entering a number whose exact value is not important. It will display as a slider or a dial control.
The **min**, **max** and **value attributes** can be used in it.
The **step attribute** specifies the granularity that the value must follow and it has to match the specified stepping interval by **min**, **max** and **value attributes**.
With this it can be decided if the valid values are going to be integers numbers (default), decimals, any, odd numbers or even numbers.

Back to index

```
<label for="volume">Volume</label>

<input type="range" id="volume" name="volume"
min="0" max="100" value="50" step="1" />

<input type="range" id="volume" name="volume"
min="0" max="100" value="1.5" step="1" />

<input type="range" id="volume" name="volume"
min="0" max="100" value="4" step="2" />
```

<input>: type checkbox

It defines a field that can be checked.
It is rendered as a box by default.
The **value attribute** sets the value of the input that will be submitted when it's checked.
The **label** will be the one displaying the value of the input to the user.
It allows you to select single values. That means, checkbox inputs are not grouped together so one or more values can be checked.

Back to index

```
<label>Which genres do you like?</label>

<label for="fantasy">Fantasy</label>

<input  type="checkbox" id="fantasy"
name="liked-genres" value="fantasy" />

<label for="comedy">Comedy</label>

<input type="checkbox" id="comedy"
name="liked-genres" value="comedy" />
```

<input>: type radio

It defines a field that can be checked.
It is rendered as a small circle by default.
The **value attribute** sets the value of the input that will be submitted when it's checked.
The **label** will be the one displaying the value of the input to the user.
It is generally used in collections of radio groups that are connected by using the same **name attribute** for all the radio. Within the same group only one option can be checked.

Back to index

```
<label>Do you want to stay connected?</label>

<label for="yes">Yes</label>
<input type="radio" id="yes" name="connected"
value="yes" />

<label for="no">No</label>
<input type="radio" id="no" name="connected"
value="no" />
```

<input>: type submit

It defines an input that can submit the form.
It renders as a button.
The **value attribute** sets the value of the input but it also defines the text of the button.
AVOID! -> using a button instead of an input type submit is considered the best practice.

Back to index

```
<input type="submit" value="Send" />
```

<input>: type reset

It defines a button that resets all the current values of the form to its initial values (the default ones passed in the input).
The **value attribute** sets the value of the input but it also defines the text of the button.
AVOID! -> it can be frustrating for the user if they accidentally click it instead of the send button.

Back to index

```
<input type="reset" value="Reset" />
```

<button>: Element

It defines a clickable button.
It has an opening and closing tag. Inside this tags you can have:

- The text to be displayed.
- Other tags that add styling or importance to the text displayed on the button.
- An image tag.

The **type attribute** specifies which type of button it is.

Back to index

```
<button type="reset">
    <b>Reset</b>
</button>
<button>
    <img src= "./photo.png" />
</button>
```

<button>: type button

It defines a clickable button.

Back to index

```
<button type="button">
    <i>Click me</i>
</button>
```

<button>: type submit

It defines a submit button for a form.

Back to index

```
<button type="submit">Send</button>
```

<button>: type reset

It defines a reset button for a form.

Back to index

```
<button type="reset">
    <b>Reset</b>
</button>
```

6. Learn media and links

<img>: Element

It embeds an image into the HTML document.
The src attribute specifies resources to be embedded. It accepts as its value a **url** or **relative path**. The **alt attribute** should be included for accessibility and SEO reasons, describing the resource used.

Back to index

```
<img src="./dog.png" alt="dog" />
<img src="../dog.jpg" alt="dog" />
<img src="./resources/dog.jpg" alt="dog" />
<img src="https://dogpicture.com/dog" alt="dog" />
```

<source>: Element

It embeds a video into the HTML document.
It is used inside the video tags.
The src attribute specifies the resource to be embedded. It accepts as its value a **url** or **relative path**.

Back to index

```
<source src="./video1.mp4" type="video/mp4" />
<source src="./video.oog" type="video/ogg" />
<source src="./video1.webm" type="video/webm />
```

<video>: Element

It embeds a video into the HTML document.

- **Src ->** specifies the location of the image. It accepts as its value a **url** or **relative path**. You can use the source element instead.
- **Autoplay ->** allows the video to start playing automatically.
- **Loop ->** allows the video to go back to the start automatically after reaching the end.
- **Mute ->** mutes the video.
- **Controls ->** adds controls to the video to pause it and play it.
- **Width** and **height ->** add the dimensions the video will occupy.

Back to index

```
<video controls autoplay loop mute width="250">
  <source src="./video1.mp4" type="video/mp4" />
  <source src="./video.oog" type="video/ogg" />
 />
Your browser does not support the video tag.
</video>

<video controls width="250" src="./video.mp4"
>
 Your browser does not support the video tag.
</video>
```