

HTML. Unit 1. Basic HTML file structure.

[1. Introduction](#)

[1.1. Elements](#)

[1.1.1. Void elements](#)

[1.2. Attributes](#)

[1.3. Internet and web pages](#)

[1.4. The web server](#)

[1.5. The web browser](#)

[2. <!doctype> and <html> elements](#)

[3. The <head> section](#)

[3.1. The <meta> element](#)

[3.1.1. Metadata](#)

[3.1.2. Charset](#)

[3.1.3. Viewport](#)

[3.2. The <title> element](#)

[3.3. The <link> element](#)

[3.4. The <script> element](#)

[4. The <body> section](#)

[4.1. Proposed exercise: Hellow world](#)

[4.2. Paragraphs and sections](#)

[4.2.1. Proposed exercise: Paragraphs](#)

[4.2.2. Proposed exercise: Headings and paragraphs](#)

[4.2.3. Proposed exercise: Headings and paragraphs](#)

[4.2.4. Proposed exercise: Headings](#)

[4.2.5. Proposed exercise: Headings and paragraphs](#)

[4.3. The
 element](#)

[4.3.1. Proposed exercise: Poems](#)

[4.3.2. Proposed exercise: Addresses](#)

[4.4. The <pre> element](#)

[4.4.1. Proposed exercise: ASCII art](#)

1. Introduction

HTML is the standard markup language for creating web pages. HTML stands for HyperText Markup Language. This language consists on some **elements** which define some kind of content. An HTML file can be edited by any simple text editor but there are some editors that use colors and tips to help the developer. One of the best free ones is Visual Studio Code. It can be downloaded [here](#).

1.1. Elements

Each element has two parts, a start tag and an end tag, and the content inserted in between:

```
1. | <tagname>Content goes here...</tagname>
```

The HTML **element** is defined by everything from the start tag to the end tag. For example:

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

1.1.1. Void elements

There are a few elements which are made of only a start tag (usually called “void elements” or “self-closing tags”). A self-closing tag is a special form of start tag with a slash immediately before the closing right angle bracket. These indicate that the element is to be closed immediately, and has no content. For example, in the following code, the void element `
` is used to print a new line in the middle of a paragraph:

```
1. | <p>The quick brown fox<br />jumps over the lazy dog.</p>
```

1.2. Attributes

HTML **attributes** provide additional information about HTML elements. We must keep in mind the following considerations:

- All HTML elements can have **attributes**.
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

For example, using the following code we can jump to the page <https://www.wikipedia.org> (the `<a>` tag defines an hyperlink, and the `href` attribute specifies the URL of the page where the link goes to):

```
1. <a href="https://www.wikipedia.org">Wikipedia</a>
```

1.3. Internet and web pages

So that an end user can get and read the contents of a web site, some basic steps must be followed:

1. To make web pages, the developer has to create files written in HTML (with any text editor) and place them on a web server (e.g., using an ftp client).
2. The HTML in any web page tells the browser what it needs to know to display the page. And, if the developer develops the web site in the right way, the pages will even display well on both desktop and mobile devices.
3. Once the files are put on a web server, any browser can retrieve the web pages over the Internet.
4. Finally, the end users will use PCs and other devices connected to the Internet, all running web browsers.

1.4. The web server

The web server is just a computer connected to the Internet waiting for requests from browsers:

1. The developer stores HTML, files, pictures, sounds and other file types.
2. Browsers make requests for HTML pages or other resources, like images.
3. If the server can locate the resources, it sends them to the browsers.

1.5. The web browser

The web browser is just an application which is in charge of displaying websites:

1. The server delivers web pages sending them to the browsers.
2. The browsers retrieve the pages.
3. The browsers display the HTML pages to the end users.

2. <!doctype> and <html> elements

All HTML documents must start with a document type declaration, meaning we are using the latest version of HTML (HTML5). The source code of the document itself will be enclosed between the tags `<html>` and `</html>` :

```
1. <!doctype html>
2. <html lang="es">
3. ...
4. </html>
```

3. The <head> section

The <head> element will mainly contain information which is not displayed on the screen (metadata, links to libraries, etc.). For example:

```
1. <!doctype html>
2. <html lang="en">
3.   <head>
4.     <!-- Required meta tags -->
5.     <meta charset="utf-8" />
6.     <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
7.
8.     <!-- Bootstrap CSS -->
9.     <link rel="stylesheet" href="bootstrap.min.css" />
10.    ...
11.  </head>
12.  ...
13. </html>
```

3.1. The <meta> element

3.1.1. Metadata

It is important that your webpages contain a list of metadata items in order to provide search engines (e.g., Google, DuckDuckGo, Bing, etc.) with semantic information such as the author, keywords, description, etc. Since this kind of information will not be shown to the user by the browser, metadata items must be placed inside the <head> tag:

```
1. <head>
2.   <meta name="description" content="My first web page" />
3.   <meta name="keywords" content="HTML, CSS, JavaScript" />
4.   <meta name="author" content="John Doe" />
5. </head>
```

3.1.2. Charset

One of the most important items inside the `<head>` section is the definition of the `charset`, so that the browser will be able to correctly display special characters such as á, é, ü, ñ, etc.:

```
1. <meta charset="utf-8" />
```

3.1.3. Viewport

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag. The viewport is the user's visible area of a web page. It varies with the device, and it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta name="viewport">` element inside all of your web pages:

```
1. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
```

3.2. The `<title>` element

We can define a title for our document with a tag also in the `<head>` section. This text will appear in the tab of the browser.

```
1. <title>My first page</title>
```

3.3. The `<link>` element

The `<link>` element is used to link to external style sheets (CSS):

```
1. <link rel="stylesheet" href="mystyle.css" />
```

3.4. The `<script>` element

The `<script>` element is used to include or define blocks of client-side JavaScript code:

```
1. <!-- Include external JavaScript file -->
2. <script src="myScript.js"></script>
3. ...
4. <!-- Define a JavaScript piece of code inside the html file -->
5. <script>
6.   function myFunction() {
7.     document.getElementById("demo").innerHTML = "Hello JavaScript!";
8.   }
9. </script>
```

4. The `<body>` section

Inside the `<body>` section we must insert the content to be shown by the browser to the user. It will be opened after closing the `<head>` section and it will be closed before closing the `<html>` section. For example, using the following code, only the text “Hello!” will be displayed on the screen by the browser:

```
1. <!doctype html>
2. <html lang="es">
3. <head>
4.   <meta charset="utf-8" />
5.   <meta name="description" content="My first web page" />
6.   <meta name="keywords" content="HTML, CSS, JavaScript" />
7.   <meta name="author" content="John Doe" />
8.   <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
9.   <title>Sample page</title>
10. </head>
11. <body>
12.   Hello!
13. </body>
14. </html>
```

4.1. Proposed exercise: Hellow world

Create a web page to display the text “Hi World, my name is ... !” inside the `<body>` section. You must fill also the `<head>` section so that the browser tab shows the “Greeting” title. Finally, insert the rest of the `<meta>` elements (charset, viewport, author, description and keywords), and check the results using a web browser.

4.2. Paragraphs and sections

For a better understanding of the text of our web page, we should divide it in paragraphs and sections.

A paragraph can be inserted between the `<p>` and `</p>` tags, and there are some tags for the section headings such as: `<h1>` , `<h2>` , `<h3>` , `<h4>` , `<h5>` and `<h6>` (from largest to smallest):

```
1. ...
2. <h1>First level heading</h1>
3. <p>My first paragraph</p>
4. <p>My second paragraph</p>
5. <p>My third paragraph</p>
6.
7. <h2>Second level heading</h2>
8. <p>My fourth paragraph</p>
9. <p>My fifth paragraph</p>
10. ...
```

4.2.1. Proposed exercise: Paragraphs

Create a web page to display several paragraphs enclosed by the `<p>` and `</p>` tags, all of them containing several lines of any text you like. Finally,

insert the rest of the `<meta>` elements (charset, viewport, author, description, keywords and title), and check the results using a web browser.

4.2.2. Proposed exercise: Headings and paragraphs

Create a web page to display several headings of different levels using all tags from `<h1>` to `<h6>`. Also put some paragraphs (enclosed by the `<p>` and `</p>` tags) with any content you like, each one after a heading. Finally, insert the rest of the `<meta>` elements (charset, viewport, author, description, keywords and title), and check the results using a web browser.

4.2.3. Proposed exercise: Headings and paragraphs

Use the correct HTML code and also open and close the `<h1>` and `<p>` tags in the right place to create a web page to display the following text:

Alicante

Alicante is a city on the Valencian Community, Spain. It is the capital of the province of Alacant and a historic Mediterranean port. The population of the city was 330,525 as of 2016, the second-largest in Valencian Community.

4.2.4. Proposed exercise: Headings

Use the correct HTML tags to create a web page containing six headings with the text “Hello”. Start with the most important heading (the largest) and end with the least important heading (the smallest).

4.2.5. Proposed exercise: Headings and paragraphs

Mark up the following text with appropriate tags (`<h1>` , `<h2>` , `<h3>` and `<p>`). Start with the most important heading (the largest) and end with the least important heading (the smallest) followed by a paragraph:

Valencia

Valencia is the capital of the autonomous community of Valencia and the third-largest city in Spain after Madrid and Barcelona, surpassing 800,000 inhabitants in the municipality. The wider urban area also comprising the neighbouring municipalities has a population of around 1.6 million people.

Geography

Location

Located on the eastern coast of the Iberian Peninsula and the western part of the Mediterranean Sea, fronting the Gulf of Valencia, Valencia lies on the highly fertile alluvial silts accumulated on the floodplain formed in the lower course of the Turia River.

Climate

Valencia has a subtropical Mediterranean climate with mild winters and hot, dry summers. Snowfall is extremely rare; the most recent occasion snow accumulated on the ground was on 11 January 1960.

4.3. The `
` element

The `
` element produces a line break in text (carriage-return). It is useful for example for writing a poem or an address, where the division of lines is significant. Keep in mind that you should not use `
` to create margins between paragraphs, since the `<p>` tag is much more suitable for those cases.

As you can see from the example below, a `
` element is included at each point where we want the text to break. The text after the `
` begins again at the start of the next line of the text block:

```
1. <p>
2.     O'er all the hilltops<br />
3.     Is quiet now,<br />
4.     In all the treetops<br />
5.     Hearest thou<br />
6.     Hardly a breath;<br />
7.     The birds are asleep in the trees:<br />
8.     Wait, soon like these<br />
9.     Thou too shalt rest.
10. </p>
```

4.3.1. Proposed exercise: Poems

Create a web page containing several poems from different authors, like the one below (you must use `<h1>` and `<h2>` headings, paragraphs and line breaks):

— Note: You should use an `<h1>` heading for the main heading and `<h2>` for the rest of the headings. Also keep in mind that you should not use `
` to create margins between each poem; use paragraphs to enclose each poem, and line breaks for each line.

POEMS

William Shakespeare

Be not afraid of greatness.
Some are born great,
some achieve greatness,
and others have greatness thrust upon them.

...

4.3.2. Proposed exercise: Addresses

Create a document to display several addresses like the one below (you must use `<h1>` and `<h2>` headings, paragraphs and line breaks):

— Note: You should use an `<h1>` heading for the main heading and `<h2>` for the rest of the headings. Also keep in mind that you should not use `
` to create margins between each address; as before, use paragraphs to enclose each address, and line breaks for each line.

ADDRESSES

Mozilla

331 E. Evelyn Avenue
Mountain View, CA
94041
USA

...

4.4. The `<pre>` element

The `<pre>` element represents **preformatted** text which is to be presented exactly as written in the HTML file. The text is typically rendered using a non-proportional (“monospace”) font. Whitespace inside this element is displayed as written.

4.4.1. Proposed exercise: ASCII art

Create a document to display some images drawn using characters, similar to the one below:

— Note: You may find some examples at the following links: <https://www.asciart.eu/> , https://en.wikipedia.org/wiki/ASCII_art , <https://asciart.website/> , <https://fsymbols.com/text-art/> , etc.

```
-----  
< I'm an expert in my field. >  
-----  
 \  ^__^  
 \ (oo)\_____  
   (__)\       )\/\  
       ||----w |  
       ||     ||
```