1. The web language: HTML

1.1. Introduction

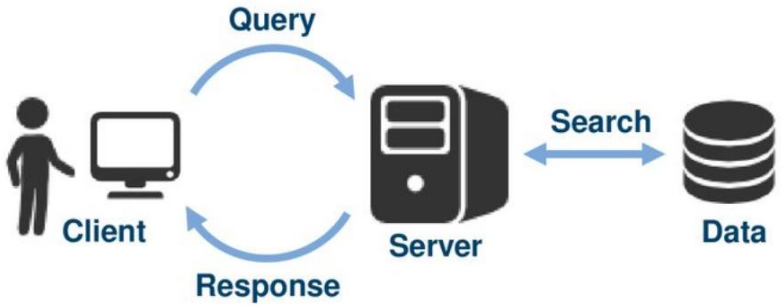
Learning objectives, concepts, and skills:

- Describe the different languages used by web applications
- Identify the differences between HTML and XHTML
- Identify the structure of an HTML document
- Identify the main elements of HTML language
- Use the HTML language for the creation and modification of web documents

1.2. Languages used in/by web applications

To create a web application, we must use several languages that allow us to design and program the application. Here we present some of the most used **languages** differentiating **client-side** and **server-side** ones.

The **Client-server model** is a distributed application structure that partitions tasks or workloads between the **providers** of a **resource** or **service**, called **servers**, and **service requesters** called **clients**. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources.



The **client** can be any computer that requests something from the server. For example, visiting any website we request the webpage from its domain. So here we act as a client.

On the other hand, the **server** is the computer that is designed to serve the requests to the client. For the same example as above, the client asks for the webpage, then the server responds with the webpage to the client.

In **web development,** the **client** is the **web browser** (Chrome, Firefox, Safari, etc.) Therefore, when we say client-side we mean anything that is happening within the web browser. This is considered the front-end because it is facing the user.

A **front-end developer** has one general responsibility: to ensure that website visitors can **easily interact** with the **page**. If the front-end of the website isn't user-friendly, the user may leave the website altogether. Therefore, front-end development is crucial to the overall design of the web experience.

A **back-end** developer oversees anything involved in the communication between the **database** and the **browser** and makes sure everything is done with precision. Basically, this is how the site works and becomes functional. Being able to make updates and create new information for how the page works will be the back-end developer's job.

A **full-stack** developer is a developer who works in the front-end and in the back-end of a web site.

1.2.1. Front-end languages

Front-end languages are those that can be directly **interpreted** by the **browser**. They are completely independent from the server, allowing the page to be hosted on any server. The only thing needed is that the browser must have the appropriate plugins installed. The code is visible by the client, which can affect safety. These are the most important:

- **HTML** (HyperText Markup Language). It is a markup language for creating static content in web documents, defining its **structure**.
- **CSS** (Cascading Style Sheets). It is a language used to define the **presentation** of a web document.
- **JavaScript**. It is a programming language that allows client-side scripting, for example, for controlling the browser or changing the content displayed in web documents.

1.2.2. Back-end languages

Back-end languages are languages that are recognized, implemented, and interpreted by the server and send information to the client in a format understandable to it. They are independent of the browser used. In addition, the code can be hidden to the client, who only sees the final HTML. Some of the most popular are:

- **JavaScript**. JavaScript is used widely in front-end development, but in recent years is used for back-end development too. Node.js (a JavaScript runtime) makes that possible by providing back-end functionality.
- **Python**. It is a multi-paradigm and multi-platform language, as it allows the use of different styles of programming. The language is simple, versatile, and quick for

development. It focuses on a syntax that promotes readable code. There are also many frameworks to choose which makes Python development very popular and accessible.

- **PHP**. Scripting language platform created specifically for generating dynamic web pages, since it can be embedded into HTML.
- **Java**. Is a versatile object-oriented programming language widely used for developing enterprise-scale web applications, desktop applications, scientific applications, big data and much more. The biggest advantage of using Java is that its code can run on any platform that supports Java.

1.3. HTML and XHTML

HTML is a markup language used to define the content and structure of web pages. The two standardization organizations **W3C** and the **WHATWG** signed an agreement to collaborate on de development of a single version of the HTML specifications. The current and unique specification can be found in the URL: https://html.spec.whatwg.org/multipage/.

XHTML (now XML syntax for HTML) is an adaptation of HTML to comply with the restrictions of XML, so it retains almost all tags and features, adding some restrictions and elements from XML. **XHTML** is a term that was historically used to describe HTML documents written to conform with XML syntax rules.

The following example shows an HTML with XML syntax:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US">
  <head>
    <title>HTML with XML syntax</title>
  </head>
  <body>
    <p>I am a HTML document with XML syntax</p>
  </body>
</html>
```

Differences between HTML and XML syntax for HTML

The **XML syntax** for **HTML** was formerly referred to as "**XHTML**". XML documents may contain a DOCTYPE if desired, but this is not required to conform the current specification.

All HTML documents with HTML syntax **must start** with a <!DOCTYPE> declaration.

The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

In the next sections, we will focus on latest **HTML** version.

1.3.1. Main elements

Tags

The tags define each element of an HTML document. They begin with the symbol < and are terminated by >. Tags are used to delimit the start and end of elements in the markup. Some elements have a start tag to indicate where they begin, and an end tag to indicate where they end. The start and end tags of certain elements can be omitted. **Void elements** only have a start tag. End tags must not be specified for void elements.

Foreign elements must either have a start tag and an end tag, or a start tag that is marked as self-closing, in which case they must not have an end tag. Elements from the **MathML** namespace and the **SVG** namespace are considered **foreign** elements.

```
<tag>
...
</tag>
```

A void element tag:

```
<tag>
```

Attributes

Tags can have attributes, which are used to configure some of the features of the tags where they are used. They will appear in the opening tag (never on the closing one), thus:

```
<tag attribute1= "valueAttribute1" attribute2="valueAttribute2" ...>
...
</tag>
```

An attribute value is a string. An element can have any number of attributes.

Comments

They are used to insert annotations in the code that are ignored by the browsers. They are written between symbols <!-- and -->. For example:

```
<!-- This is my comment -->
```

Basic structure

A basic HTML document looks like this:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>My first HTML document</title>
    </head>
    <body>
        <p>Test HTML document</p>
    </body>
</html>
```

The <!DOCTYPE> specifies the HTML version of the page.

The browser needs this information to know how to interpret the HTML.

This declaration must appear on the first line of the document.

Header

The header of an HTML document is delimited by the <head> tag. It contains information about the document that is not visible to the user except its title, indicated by the <title> tag, that browsers display in the top left corner of the browser window (or in modern browsers, in the tab header where the page is shown). This element is placed at the top of the page inside the <html> tag.

Body

The <body> tag is a container for all content on a web page.

A page can only have one <body> element.

Character entities

Sometimes we need to include in a web page characters that have a special meaning in HTML, like the symbols greater or less than, the quotes, or simply we want to avoid coding problems in browsers when including accents. Then we use character entities. Some of them can be seen in the following table:

Character	HTML Entity	Character	HTML Entity
á	á	Á	Á
é	é	É	É
í	í	Í	Í
ó	ó	Ó	Ó
ú	ú	Ú	Ú
ü	ü	Ü	Ü
ñ	ñ	Ñ	Ñ
<	<	&	&
>	>	"	&quote;
non-breaking space			

Example

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>HTML with entities</title>
    </head>
    <body>
        <p>Este &eacute; un documento con entidades de car&aacute;cter.</p>
    </body>
</html>
```

The following **link** shows a table with many more character entities.

Namespaces

Namespaces cannot be represented using the HTML syntax, but they are supported in the XML syntax.

Semantic HTML

Semantic HTML is an HTML code that uses HTML tags to effectively describe the purpose of page elements. Semantic HTML code communicates the meaning of its elements to both computers and humans, which helps web browsers, search engines, assistive technologies, and human developers understand the components of a web page.

Here's a basic example: The ** ** (paragraph) tag is a semantic HTML tag — all content between its opening ** ** tag and closing ** ** tag is a block of paragraph of text. Anyone or any device reading this tag will understand its purpose.

Semantic HTML not only makes your code neater and easier to read — it also makes it more accessible to assistive technology and search engines.

Semantic HTML tags for text are tags that convey the meaning of the text that they contain. Some semantic HTML tags for text are: (paragraph), <h1>, <h2>, <h3>, <h4>, <h5>, <h6> (headings), <ol> (ordered list), <ul> (unordered list), <a> (anchor), <q> (quote) and <blockquote>, <code>, and .

Semantic HTML tags for structure are tags that serve to group child elements into sections. Each section contains related content.

Some semantic HTML tags for structure are: <header>, <footer>, <main>, <nav>, <aside>, <article>, <section>.

Spacing, line breaks and paragraphs

In HTML, white spaces, tabulators, carriage returns, and line breaks are all considered 'whitespace'.

Paragraphs tags or tags in HTML help us create paragraphs on a web page. On web browsers, paragraphs are displayed as blocks of text separated from adjacent blocks by blank lines, white spaces, or first-line indentation.

HTML line break tags help when we don't want to start a new paragraph but want the sentence to start from a new line. Using the
 tag, we can break the sentence continuation and make it begin on a new line.

HTML horizontal line <hr> tag is used to separate out different topics on a page. With this tag when we can create a thematic break or separate items on an HTML page.

The <pre> tag in HTML is used to define the **block of preformatted text** which preserves the text spaces, line breaks, tabs, and other formatting characters which are ignored by web browsers. Text in the <pre> element is displayed in a fixed-width font, but it can be changed using CSS. The <pre> tag requires a start and an end tag.

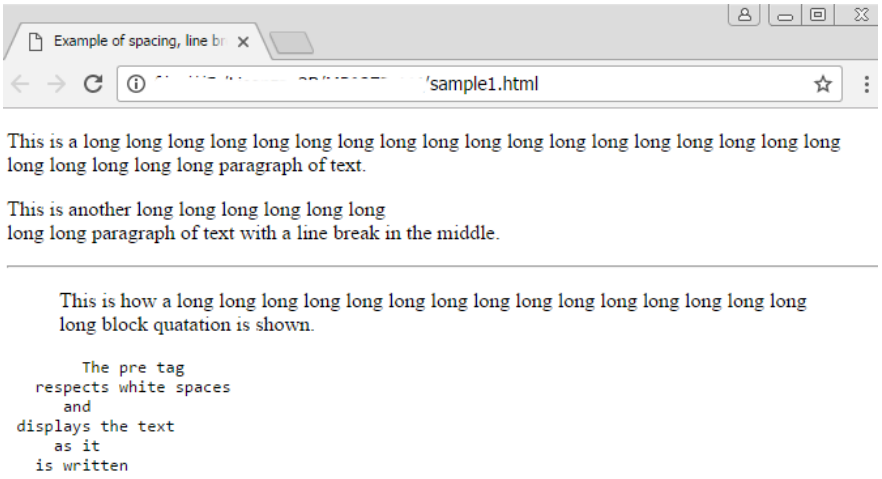
The <blockquote> tag in HTML is used to display the long **quotations** (a section that is quoted from another source). It changes the alignment to make it unique from others. It contains both opening and closing tags. In blockquote tag, we can use elements like heading, list, paragraph, etc.

Tag	Description
<p>	Marks out the content of a text paragraph. Defines a paragraph.
 	Inserts a line break.
<hr>	Sets a thematic break in an HTML page. Usually represented as a horizontal line.
<pre>	Pre-formatted text is presented 'as written', maintaining all white spaces.
<blockquote>	A block section that is quoted from another source.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example of spacing, line breaks and paragraphs</title>
    </head>
    <body>
        <p>This is a long long long long long long long long long long long long long
long long long long long long long long long long paragraph of text.</p>
        <p>This is another long long long long long long <br/> long long paragraph of
text with a line break in the middle.</p>
        <hr/>
        <blockquote>This is how a long long long long long long long long long long long
long long long long long block quotation is shown.</blockquote>
        <pre>        The pre tag
            respects white spaces
            and
            displays the text
            as it
            is written
        </pre>
        <br/>
    </body>
</html>
```

The following image shows the example document in the browser:



Heading tags

A **HTML heading** or HTML **h** tag can be defined as a **title** or a **subtitle** which you want to display on the webpage. When placing the text within the heading tags <h1>.....</h1>, it is displayed on the browser in the bold format and the size of the text depends on the number of the heading.

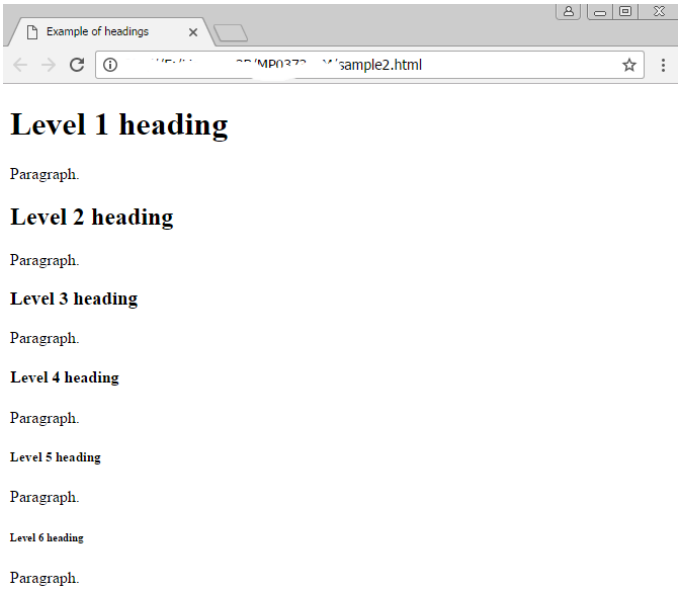
There are six different HTML headings which are defined with the <h1> to <h6> tags, from highest level h1 (main heading) to the least level h6 (least important heading).

h1 is the largest heading tag and h6 is the smallest one. So h1 is used for the most important heading and h6 is used for least important.

Tag	Description
<h1>, <h2>, <h3>, <h4>, <h5> and <h6>	Define the headers of the sections of the page. <h1> would be the highest level and <h6> the lowest level.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example of headings</title>
    </head>
    <body>
        <h1>Level 1 heading</h1>
        <p>Paragraph. </p>
        <h2>Level 2 heading</h2>
        <p>Paragraph. </p>
        <h3>Level 3 heading</h3>
        <p>Paragraph. </p>
        <h4>Level 4 heading</h4>
        <p>Paragraph. </p>
        <h5>Level 5 heading</h5>
        <p>Paragraph. </p>
        <h6>Level 6 heading</h6>
        <p>Paragraph. </p>
    </body>
</html>
```



Basic markup

HTML tag is a phrase tag which is used to represent the important text of a document on the browser. The text within **** text has **semantic importance** for the search engines and emphasises the text with special intonation.

The text within tag renders in bold font on the browser by default.

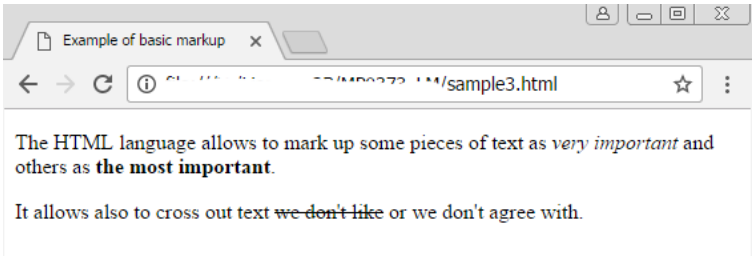
HTML **** tag is used to **stress emphasis** the text within a sentence or phrase. It gives semantic meaning to the text contained within it and renders in the italic form on the browser.

HTML <del> tag is used to represent a range of text that has been **deleted/removed** from the document. It is used as a Markup for the deleted content. The browser renders it by striking a line through the deleted text.

Tag	Description
	Defines as emphasised text. By default, cursive font.
	Defines an important text. By default, bold font.
<del>	Marks a piece of text as deleted. By default, strikeout.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example of basic markup</title>
    </head>
    <body>
        <p>The HTML language allows to mark up some pieces of text as
        <em>very important</em> and others as <strong>the most important</strong>.</p>
        <p>It allows also to cross out text <del>we don't like</del> or we don't agree
with.</p>
    </body>
</html>
```



Lists

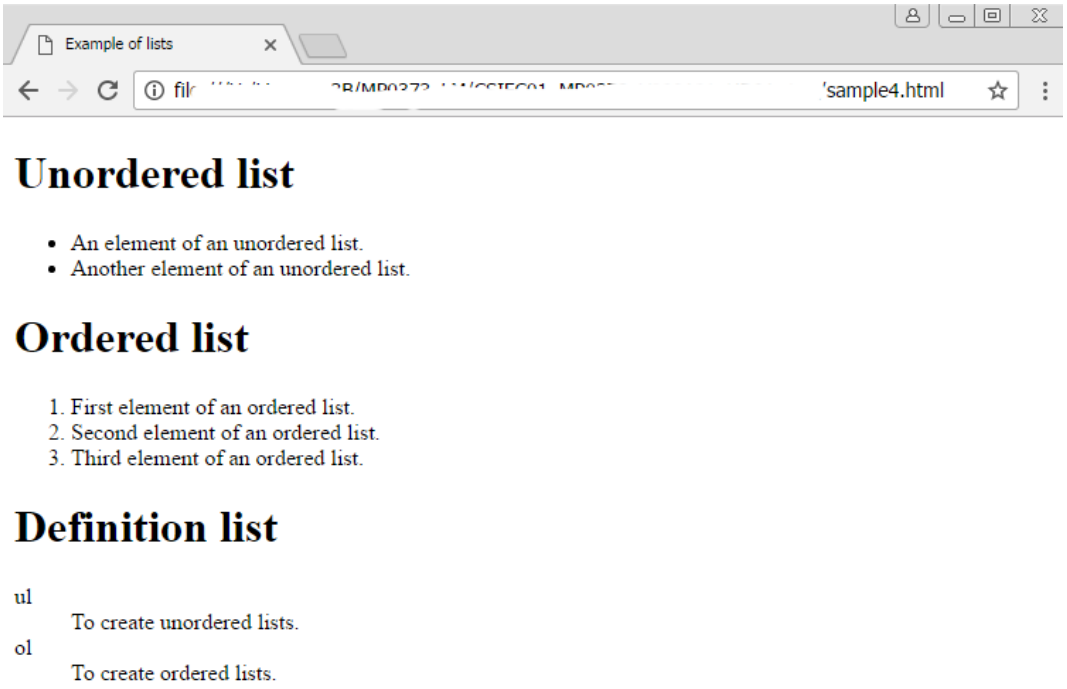
HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol)
2. Unordered List or Bulleted List (ul)
3. Description List or Definition List (dl)

Tag	Description
	Defines an unordered (bulleted) list.
	Defines an ordered list.
	To define each list element (ordered or unordered list).
<dl>	Used to create a definition list.
<dt>	Used to mark the entries of a definition list.
<dd>	Used to mark the definition texts of the definition list.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example of lists</title>
    </head>
    <body>
        <h1>Unordered list</h1>
        <ul>
            <li>An element of an unordered list.</li>
            <li>Another element of an unordered list.</li>
        </ul>
        <h1>Ordered list</h1>
        <ol>
            <li>First element of an ordered list.</li>
            <li>Second element of an ordered list.</li>
            <li>Third element of an ordered list.</li>
        </ol>
        <h1>Definition list</h1>
        <dl>
            <dt>ul</dt>
            <dd>To create unordered lists.</dd>
            <dt>ol</dt>
            <dd>To create ordered lists.</dd>
        </dl>
    </body>
</html>
```



Images

The **img** element represents an **image**, which is an **external resource** that can be embedded in the body of a document. The location of such an external resource must be specified in the **src** attribute.

The img element should not be used to insert images without a meaning for the document.

Tag	Description	
`<img>`	To include an image in a web document	
	Attribute	**Description**
	`src = "url"`	URL of the image
	`alt = "text"`	Brief description of the image
	`height = "measure"`	Specifies the display height of the image (may differ from the original height)
	`width = "measure"`	Specifies the display width of the image (may differ from the original width)

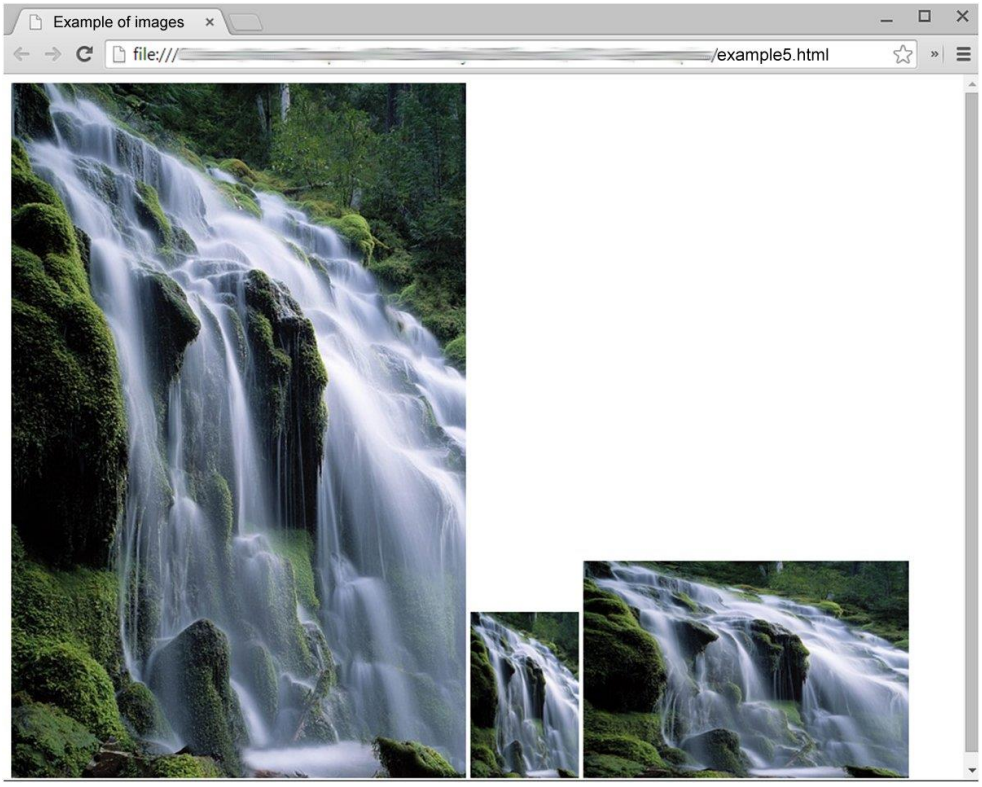
Images are visual elements. This is one of the reasons for the existence of the alt attribute, which is intended to provide an alternative version of the content or purpose of the image for those situations where images cannot be accessed properly (visually impaired users, unsupported browsers, configurations where images are disabled, etc.).

The **alt attribute** should provide an **alternative version** of the image by achieving the **same purpose of the image through text**. A description of the image content is not always the best approach.

Optional but **highly recommended** attributes are **height** and **width**. The height marks the height of the image, while the width marks the width. They are recommended because they help the browser to represent the image.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example of images</title>
    </head>
    <body>
        <img src="image_example5.jpg"/>
        <img src="image_example5.jpg" alt="waterfall" width="100"/>
        <img src="image_example5.jpg" alt="deformed waterfall" width="300" height="200"/>
    </body>
</html>
```



Links

A link or **hyperlink** is a connection from one web resource to another. Links allow users to move seamlessly from one page to another, on any server anywhere in the world.

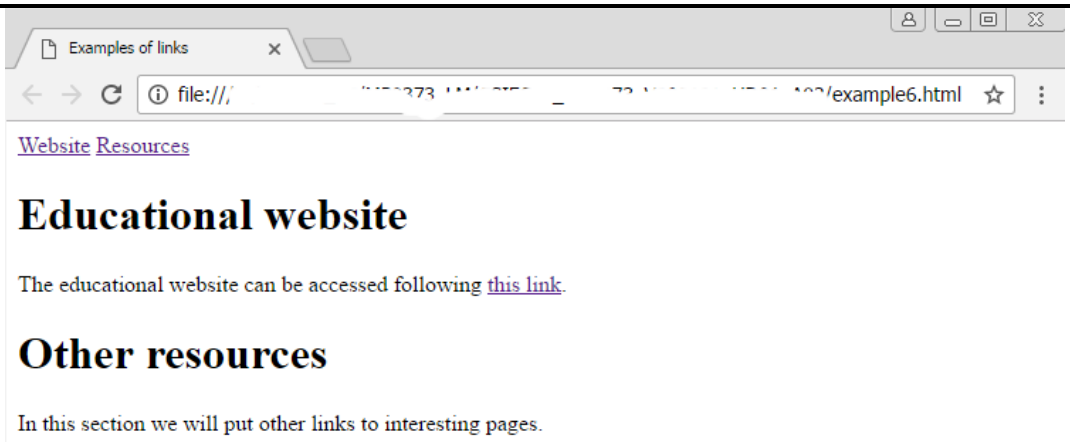
The **href** attribute specifies the target of the link. Its value can be an absolute or relative URL.

A link has two ends, called **anchors**. The link starts at the source anchor and points to the destination anchor, which may be any web resource, for example, an image, an audio or video clip, a PDF file, an HTML document, or an element within the document itself, and so on.

Tag	Description	
<a>	Defines a link to another document.	
	Attribute	**Description**
	href = "url"	The URL of the link's destination.
	name = "text"	Associates a text name to the link. This name can be used to refer to this link from other parts of the document, preceding this name with the # character.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Examples of links</title>
    </head>
    <body>
        <a href="#portal">Website</a>
        <a href="#resources">Resources</a>
        <h1>Educational website</h1>
        <a name="portal"/>
        <p>The educational website can be accessed following <a
href="http://edu.xunta.gal">this link</a>.</p>
        <h1>Other resources</h1>
        <a name="resources"/>
        <p>In this section we will put other links to interesting pages.</p>
    </body>
</html>
```



The **target** attribute tells the browser where to open the linked document. There are four defined targets, and each target name starts with an underscore(_) character:

- _blank — Opens the linked document in a new window or tab.
- _parent — Opens the linked document in the parent window.
- _self — Opens the linked document in the same window or tab as the source document. This is the default; hence it is not necessary to explicitly specify this value.
- _top — Opens the linked document in the full browser window.

The **name** attribute is used to create a **named anchor**. When using named anchors, you can create links to a specific section on a page, instead of letting your viewer scroll around to find what he/she is looking for.

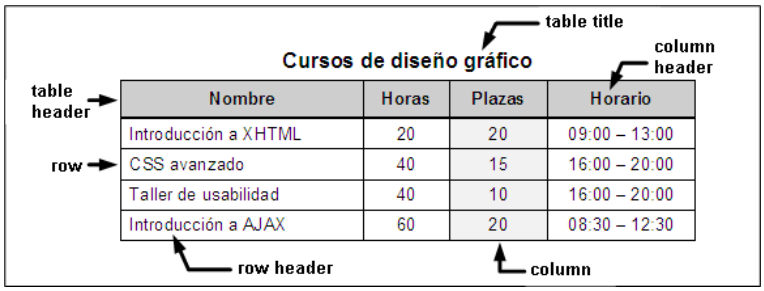
The name of the anchor can be any text you care to use. However, it should not have spaces in the name. Use a hyphen or underscore and all lowercase letters.

A named anchor is not displayed in any special way.

Tables

An **HTML table** is defined with the **table** tag. Each table row is defined with the **tr** tag. A table **header** is defined with the **th** tag. By default, table headings are bold and centered. A table data/cell is defined with the **td** tag.

The following image shows the parts of an HTML table:

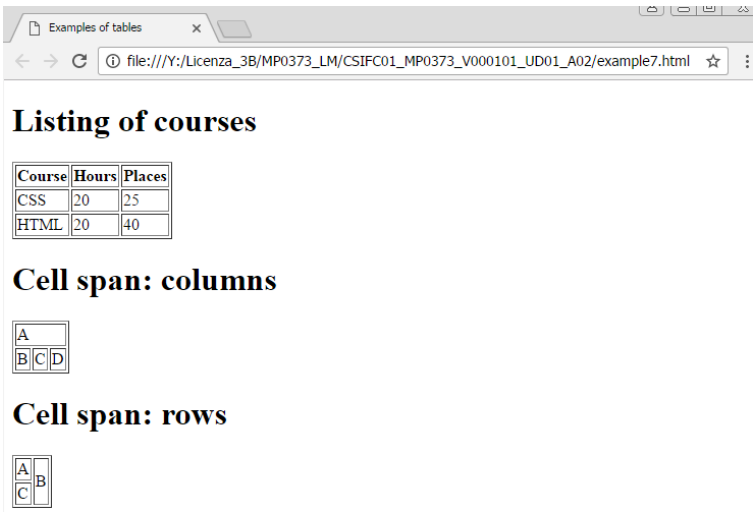


Tag	Description	
`<table>`	Defines a table.	
`<tr>`	Defines a table row.	
`<td>`	Defines a cell within a row (that is, a column).	
	Attribute	**Description**
	`colspan = "number"`	Number of columns that take up that cell.
	`rowspan = "number"`	Number of rows that take up that cell.
`<th>`	Used to define the cells of the table header.	
	Attribute	**Description**
	`colspan = "number"`	Number of columns that take up the cell.
	`rowspan = "number"`	Number of rows that take up the cell.
`<caption>`	Defines the title of the table.	
`<thead>`	Groups the header content in a table.	
`<tbody>`	Groups the body content in a table.	
`<tfoot>`	Groups the footer content in a table.	

Example 1

NOTE: In the example the border attribute is used to make the edge of cells visible, to improve the visualisation, but it should not be set in this way, CSS (cascaded style sheets) should be used instead.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Examples of tables</title>
    </head>
    <body>
        <h1>Listing of courses</h1>
        <table border="1">
            <tr>
                <th>
                    <strong>Course</strong>
                </th>
                <th>
                    <strong>Hours</strong>
                </th>
                <th>
                    <strong>Places</strong>
                </th>
            </tr>
            <tr>
                <td>CSS</td>
                <td>20</td>
                <td>25</td>
            </tr>
            <tr>
                <td>HTML</td>
                <td>20</td>
                <td>40</td>
            </tr>
        </table>
        <h1>Cell span: columns</h1>
        <table border="1">
            <tr>
                <td colspan="3">A</td>
            </tr>
            <tr>
                <td>B</td>
                <td>C</td>
                <td>D</td>
            </tr>
        </table>
        <h1>Cell span: rows</h1>
        <table border="1">
            <tr>
                <td>A</td>
                <td rowspan="2">B</td>
            </tr>
            <tr>
                <td>C</td>
            </tr>
        </table>
    </body>
</html>
```



Example 2

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Example using thead, tbody and tfoot</title>
  </head>
  <body>
    <table border="1">
      <caption>Shopping list</caption>
      <thead>
        <tr>
          <th>Product</th>
          <th>Quantity</th>
          <th>Price</th>
        </tr>
      </thead>
      <tfoot>
        <tr>
          <td colspan="2">Total Price</td>
          <td>20</td>
        </tr>
      </tfoot>
      <tbody>
        <tr>
          <td>Screwdriver</td>
          <td>1</td>
          <td>3</td>
        </tr>
        <tr>
          <td>Wrench</td>
          <td>2</td>
          <td>5</td>
        </tr>
        <tr>
          <td>Hammer</td>
          <td>1</td>
          <td>7</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Shopping list

Product	Quantity	Price
Screwdriver	1	3
Wrench	2	5
Hammer	1	7
Total Price		20

Forms

<form> is an HTML element to **collect input data** using interactive controls. It provides facilities to input text, number, values, email, password, and control fields such as checkboxes, radio buttons, submit buttons, etc., or form is a container that contains input elements like text, email, number, radio buttons, checkboxes, submit buttons, etc. Forms are used to collect data from the user. For example, a user wants to buy a bag online, so he/she must first enter their shipping address in the address form and then add their payment details in the payment form to place an order.

Forms are created by placing input fields within paragraphs, pre-formatted text, lists, and tables. This gives considerable flexibility in designing the layout of forms.

Tag	Description	
`<form>`	Defines a form within the page	
	Attribute	**Description**
	`action = "url"`	Indicates the URL which processes the form data
	`method = "POST" or "GET"`	Indicates the HTTP method used to submit the form
`<label>`	Used to specify a caption for an <input> element of a form. The caption can be associated with a specific form control, either using the for attribute, or by putting the form control inside the label element itself	
	Attribute	**Description**
	`for ="input_id"`	Specifies which form element a label is bound to
`<button>`	Used to create a clickable button within HTML form on your webpage. You can put content like text or image within the <button>.....</button> tag.	
	Attribute	**Description**
	`name`	Specifies the name of the button
`<input>`	Used to insert a control in a form	
	Attribute	**Description**
	`type = "text | password`	Sets the type of control

<table>
<tr><td rowspan="15"></td><td rowspan="13">| checkbox | radio | submit | reset | file | hidden | image | button | color | date | datetime-local | email | month | number | range | search | tel | time | url week"</td><td>text</td><td>Name</td></tr>
<tr><td>password</td><td>Password</td></tr>
<tr><td>checkbox</td><td>Jobs searching for
Manager Technical Employee</td></tr>
<tr><td>radio</td><td>Sex
Man Woman</td></tr>
<tr><td>submit</td><td>Search
Button to send the form to the server.</td></tr>
<tr><td>reset</td><td>Clear form
Button to reset the form to its initial state.</td></tr>
<tr><td>file</td><td>Choose file...
Choose File No file chosen</td></tr>
<tr><td>hidden</td><td>Used to include hidden content in the form.</td></tr>
<tr><td>image</td><td>
Customized button with an image.</td></tr>
<tr><td>button</td><td>Save changes
Button.</td></tr>
<tr><td>color</td><td>Select your favorite color:
Creates a color picker</td></tr>
<tr><td>date</td><td>Birthday: dd/mm/aaaa Enviar
Creates a field to accept a date</td></tr>
<tr><td>number</td><td>Quantity (between 1 and 5): 3
Defines a field for entering a number.</td></tr>
<tr><td>range</td><td>Volume (between 0 and 50):
Represents a field for the input of a number inside a range, with the caveat that choosing an exact value isn't important.</td></tr>
<tr><td>tel</td><td>Enter a telephone number: (555) 555-5555
Defines a field for entering a telephone number.</td></tr>
<tr><td>name = "text"</td><td colspan="2">Assigns a name to the control (essential for the server to process the form).</td></tr>
<tr><td>value = "text"</td><td colspan="2">Initial value of the control.</td></tr>
<tr><td>alt = "text"</td><td colspan="2">Description of the control</td></tr>
<tr><td><textarea></td><td colspan="3">Defines a multi-line input field (a text area) in the form.</td></tr>
<tr><td></td><td colspan="2">Attribute</td><td>Description</td></tr>
</table>

	rows = "number"	Number of rows that will show the text area.
	cols = "number"	Number of columns that will show the text area.
<select>	Used to create a drop-down list in a form.	
	Attribute	**Description**
	size = "number"	Number of rows initially shown (by default: one)
	multiple = "multiple"	If included, it allows to select more than one element simultaneously.
<option>	Used to define each element of the drop-down list.	
	Attribute	**Description**
	selected	Tells if the element appears initially selected when the page is loaded.
	value = "text"	The value sent to the server when the user selects this option.
<optgroup>	The <optgroup> tag groups dropdown options in a <select> control. This is useful when dropdown options fall into different categories. This tag displays an unselectable title above the items in the group.	
	Attribute	**Description**
	label= "text"	Option group label.
<fieldset>	Used to make a group of related elements in the form. It creates the box over the elements. The <legend> tag is used to define the title for the child's contents.	
	Attribute	**Description**
	disabled	Used to specify that the group of related form elements is disabled. A disabled fieldset is un-clickable and unusable.
<output>	The output element represents the output of a calculation or process, performed usually by a script	
	Attribute	**Description**
	for	The for attribute should contain a space-separated list of tokens, each of which should match the id attribute of an element or control used in the calculation.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Exemplo de formulario</title>
    </head>
    <body>
        <form>
            <br>
            <label>Name
                <input type="text" name="name" value="">
            </label>
            <br><br>
            <label for="contrasinal">Password </label>
            <input type="password" name="password" id="contrasinal"/>
            <br><br>
```

```
            Jobs serching for <br>
            <input name="manager" type="checkbox" value="manager" id="manager"/>
            <label for="manager">Manager</label>
            <input name="technical" type="checkbox" value="technical" id="technical"/>
            <label for="manager">Technical</label>
            <input name="employee" type="checkbox" value="empleado" id="employee"/>
Employee<br><br>
            Gender <br>
            <label><input type="radio" name="gender" value="male"> Male</label>
            <label><input type="radio" name="gender" value="female"> Female</label>
            <label><input type="radio" name="gender" value="unknown" checked> I prefer
not to say</label><br><br>
            <input type="submit" name="procurar" value="Search"/><br><br>
            Academic level <br>
            <select id="estudos" name="estudos">
                <option value="" selected="selected">- select -</option>
                <option value="not_school">Did not attend to school</option>
                <option value="primary">Primary education</option>
                <option value="secondary">Secondary education</option>
                <option value="bachelor">Post-16 education</option>
                <option value="technician">Medium level vocational education</option>
                <option value="technician2">Higher level vocational education</option>
                <option value="university">University education</option>
            </select><br><br>
            Write a short presentation:<br>
            <textarea id="presentation" name="presentation" cols="40"
rows="5"></textarea><br><br>
            <input type="reset" name="clear" value="Clear form"/><br><br>
            Choose file...<br>
            <input type="file" name="attached"/><br><br>
            <input type="hidden" name="enterprise" value="Enterprise1"/>
            <input type="image" name="submit" src="submit_img.png" alt="Submit"
width="100" height="32"/><br><br>
            <input type="button" name="save" value="Save changes"/>
        </form>
    </body>
</html>
```

Name

Password

Jobs serching for
Manager Technical Employee

Gender
Male Female I prefer not to say

Search

Academic level
- select -

Write a short presentation:

Clear form

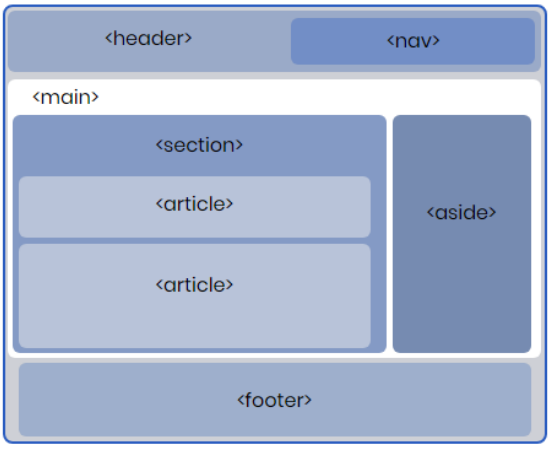
Choose file...
Seleccionar archivo Ninguno ...hivo selec.

Submit

Save changes

Semantic HTML Tags for Structure

HTML offers new semantic tags to define different parts of a web page. Some of these semantic tags are: <article>, <aside>, <details>, <footer>, <header>, <main>, <nav> and <section>.



Article

The <article> HTML Tag defines a piece of material that can stand on its own within a document, page, program, or web page. The <article> HTML tag material is comprehensible. It stands on its own as a stand-alone piece of material on the page. On forums, blogs, news stories, comments, interactive widgets or gadgets, or any other individual piece of material, and so on, the <article> HTML Tag is commonly used.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Section example</title>
    </head>
    <body>
        <article>
            <h1>Heading for Article</h1>
            <p>Text that appears under article</p>
        </article>
    </body>
</html>
```

Aside

The <aside> HTML tag indicates a section of a document whose content is only slightly related to the document's primary content. Commonly, asides are presented as sidebars or call-out boxes. The <aside> HTML Tag is used to describe the web page's primary object in a concise manner, like a highlighter.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Aside example</title>
    </head>
    <body>
        <article>
            <h1>HTML</h1>
            <p><strong>This is a title.</strong> More content.</p>

        </article>
        <aside>
            <h2>Related links</h2>
            <nav>
                <ul>
                    <li><a href="#">XML</a></li>
                    <li><a href="#">CSS</a></li>
                    <li><a href="#">JavaScript</a></li>
                </ul>
            </nav>
        </aside>
    </body>
</html>
```

Details

The **<details> tag** creates a control, a so-called disclosure widget.

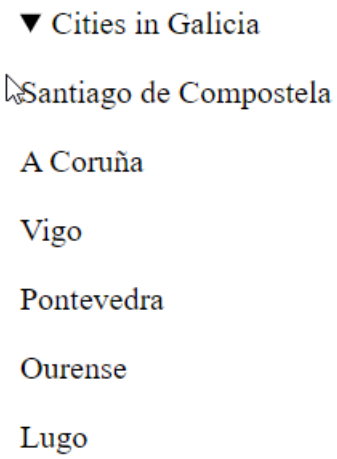
This control can be toggled (opened and closed) to reveal its details.

It is rendered with a triangular button that indicates the widget's state.

The caption for this element is specified with the <summary> tag.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Details example</title>
    </head>
    <body>
        <details>
            <summary>Cities in Galicia</summary>
            <p>Santiago de Compostela</p>
            <p>A Coruña</p>
            <p>Vigo</p>
            <p>Pontevedra</p>
            <p>Ourense</p>
            <p>Lugo</p>
        </details>
    </body>
</html>
```



Footer

The **<footer> tag** in HTML is used to define a footer of HTML document. This section contains the footer information (author information, copyright information, carriers, etc).

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Footer example</title>
    </head>
    <body>
        <footer>
```

```
            <a href="https://www.example.gal/about/">About Us</a>|
            <a href="https://www.example.gal/privacy-policy/">Privacy Policy</a>|
            <a href="https://www.example.gal/careers/">Careers</a>
            <p>Some rights reserved</p>
        </footer>
    </body>
</html>
```

Header

The **<header> tag** in HTML is used to define the header for a document or a section as it contains the information related to the title and heading of the related content. The <header> element is intended to usually contain the section's heading (an h1-h6 element), but this is not required. It can also be used to wrap a section's table of contents, a search form, or any relevant logos. There can be several <header> elements in one document. This tag cannot be placed within a <footer>, <address> or another <header> element.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Header example</title>
    </head>
    <body>
        <article>
            <header>
                <h2>Title of Article</h2>
                <p>by the author ...</p>
            </header>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam volutpat
sollicitudin nisi,
                at convallis nunc semper et. Donec ultrices odio ac purus facilisis, at
mollis urna finibus.
            </p>
        </article>
    </body>
</html>
```

Main

The **<main> tag** is used to denote the content of a webpage that relates to the central topic of that page or application. It should include content that is unique to that page and should not include content that is duplicated across multiple webpages, such as headers, footers, and primary navigation elements.

The <main> tag can only be used **once** in each **HTML file**. The <main> tag must be used outside other structural elements, such as <head> and its closing tag, and <footer> and its closing tag. It cannot overlap them or sit within them.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Main example</title>
    </head>
    <body>
        <main>
            <h1>Web development</h1>
            <p>Web development is the work involved in developing a website for the
Internet (World Wide Web) or an intranet (a private network). </p>
            <article>
                <h2>HTML</h2>
                <p>Learn HTML (HyperText Markup Language) to create website.</p>
            </article>

            <article>
                <h2>CSS</h2>
                <p>Learn CSS (Cascading Style Sheets) to create stunning designs and
effects. </p>
            </article>
        </main>
    </body>
</html>
```

Nav

The **<nav> tag** is used for declaring the navigational section in HTML documents. Websites typically have sections dedicated to navigational links, which enables users to navigate the site. These links can be placed inside a nav tag. In other words, the nav element represents a section of the page whose purpose is to provide navigational links, either in the current document or to another document. The links in the nav element may point to other webpages or to different sections of the same webpage. Common examples of the nav elements are menus, tables, contents, and indexes.

The nav tag is reserved for primary navigation areas, like the main menu across the top of the page or section. A **document** may have **several nav elements**, for example, site navigation and one for intra-page navigation. Links within nav tag can be codes within a ul list or simply coded as separate links, without ul element. This element makes it much easier to create a navigation menu, to create a neat horizontal menu of text links, and it helps screen readers software to correctly identify primary navigation areas in the document.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Nav example</title>
    </head>
    <body>
```

```
        <h1>Learn HTML</h1>
        <h2> HTML nav Tag</h2>
        <!-- nav tag starts -->
        <nav>
            <a href="#">Home</a> |
            <a href="#">Interview</a> |
            <a href="#">Languages</a> |
            <a href="#">Data Structure</a> |
            <a href="#">Algorithm</a>
        </nav>
        <!-- nav tag ends -->
    </body>
</html>
```

Section

The <section> element is designed to contain a thematically defined piece of document. The contents of a section are usually led by a heading. Any given web page or article could have many sections.

The <section> element should always be the last choice when there is no other element that is more relevant.

Example

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Section example</title>
</head>

<body>
<section>
  <h1>Heading for Section</h1>
  <p>Text that appears under section</p>
</section>
</body>

</html>
```