

Session 1. HTML5

Programació Multiplataforma i Distribuïda

Grau d'informàtica. EPSEVG

Setembre 2023

Jordi Esteve jesteve@cs.upc.edu

HTML5

HTML (Hypertext Markup Language) is the standard [markup language](#) for documents designed to be displayed in a [web browser](#). It can be assisted by technologies such as [Cascading Style Sheets](#) (CSS) and [scripting languages](#) such as [JavaScript](#).

HTML is used to define the web pages with hyperlinks sent by the web servers that can be displayed in the web browsers.

History:

- Berners-Lee (physicist at CERN) specified HTML and wrote browser & server software in late 1990 for CERN researchers to use and share documents.
- HTML 2: November 24, 1995
- HTML 3: January 14, 1997
- HTML 4: December 18, 1997
- **HTML 5: October 28, 2014**

Since 1996, the HTML specifications have been maintained, with input from commercial software vendors, by the [World Wide Web Consortium](#) (W3C).

HTML element

HTML elements are defined by **HTML tags** enclosed in angle brackets like <p>, <a>, , ...

All HTML elements have the **content** between a **start tag** and **end tag**, except a few ones without any content that only have a start tag. All the start tags can have attributes but they are optional.

Example of a paragraph element containing "Example text" and belonging to the "type1" class:

<p class="type1">Example text</p>

<	p	class="type1"	>	Example text	</	p	>
	label	attribute = value				label	
Start tag				Content	End tag		

Example of an image element contained in the picture.png file (it has no content or end tag):

<	img	src="picture.png"	>
	label	attribute = value	
Start tag			

HTML identifier & class attributes. Comments.

id attribute uniquely identifies an HTML element in a page. An HTML element should only have one id attribute and this must be unique on the page.

```
<section id="sec1">  
  This is the <u id="c2">content</u> of the page.  
</section>
```

class attribute defines a class of HTML elements. An element can have several class attributes.

```
<section class="normal indexed">  
  This is the <u class="special">content</u> of the page.  
</section>
```

In the last example, section element belongs to two classes: normal and indexed.

To comment one or several lines of a HTML file you can use this syntax:

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

HTML document

An HTML document must have a minimum of 3 elements: <html>, <head> and <body>:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Page title</title>
  </head>
  <body>
    This is the content of the page.
  </body>
</html>
```

Inside the <head> tags we define auxiliary data like the charset (character codification set, UTF-8 supports all the languages) and the page title that will be shown at the top of the window Web browser.

Inside the <body> tags we define the content of the Web page.



Exercise: Create a file named "index.html" with a text editor, copy the previous HTML content, save the file and open it in a Web browser.

Visual Studio Code editor

The Visual Studio Code editor includes several helpers and tricks to improve the productivity while we are writing code.

- [Intellisense](#) for HTML5, CSS and Java-Script, a contextual helper when you put the mouse cursor over any web element (tag or attribute).
- [Emmet](#) snippets and expansions. Test these examples, for example typing the ! and the Tab key:

! + Tab

tag + Tab

tag#id + Tab

tag.class + Tab

tag[attr=value] + Tab

tag1+tag2 + Tab

tag1>tag2 + Tab

tag*3 + Tab

tag#id\$*3 + Tab

tag{\$}*3 + Tab

lorem + Tab

lorem*3 + Tab

link:css + Tab

link:favicon + Tab

You can install an extension for Visual Studio Code editor like **Live Preview** to preview your web projects. Click the right mouse button while editing an html file and choose the *Show Preview* option. The changes on your code are refreshed automatically in the preview window.

HTML character entities

The characters reserved in HTML (<, >, ", ', &) and characters that are not present on the keyboard can be replaced by entities. See [Character Entity Reference Chart](#)

Some examples:

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
€	euro	€	€
©	copyright	©	©

HTML. Block & Inline elements

The **block** elements occupy all the width of the page and force a new line before and after.

Example of block elements are related to paragraphs, headers, content blocks, lists, ... like:

<article> <aside> <canvas> <div> <footer> <h1> <h2> <h3> <h4> <h5> <h6> <header> <p>
<pre> <section> ...

The **inline** elements only occupy the necessary width and not force new lines.

Example of inline elements are related to character semantics, links, buttons, form fields, ... like:

<a>
 <button> <input> <select> <textarea> <u> ...

- **bold** or **strong** text: (do not use)
- **italic** or **emphasized** text: (do not use <i> </i>)
- **underlined** text: <u> </u>
- **inserted** text: <ins> </ins>
- **crossed** or **deleted** text:
- Generic container of some text without semantic information:

HTML paragraphs

The HTML `<p>` element defines a **paragraph**.

The HTML `
` element defines a **line break**. Use `
` if you want a line break (a new line) without starting a new paragraph.

The HTML `<pre>` element defines **preformatted** text. The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

Headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the most important heading. `<h6>` defines the least important heading.

```
<h1>Chapter 1</h1>
  <h2>Section 1.1</h2>
    <h3>Subsection 1.1.1</h3>
    <h3>Subsection 1.1.2</h3>
  <h2>Section 1.2</h2>
    <h3>Subsection 1.2.1</h3>
    <h3>Subsection 1.2.2</h3>

<h1>Chapter 2</h1>
  <h2>Section 2.1</h2>
    <h3>Subsection 2.1.1</h3>
    <h3>Subsection 2.1.2</h3>
```

HTML links

A link or hyperlink is defined with the `<a>` tag:

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

External link:

```
<a href="https://www.upc.edu">Visit UPC Web Page</a>
```

Local link:

```
<a href="download/html_tutorial.pdf">HTML tutorial</a>
```

The `target` attribute specifies where to open the linked document. It can have one of these values:

- `_blank`: Opens the linked document in a new window or tab
- `_self`: Opens the linked document in the same window/tab as it was clicked (this is default)
- `_parent`: Opens the linked document in the parent frame
- `_top`: Opens the linked document in the full body of the window
- `framename`: Opens the linked document in a named frame

```
<a href="https://www.upc.edu" target="_blank">Visit UPC Web Page</a>
```

HTML images

Images (PNG, GIF, JPEG, WebP, SVG) are defined with the `` tag.

The `` tag does not have content or closing tag, it contains attributes only.

The `src` attribute specifies the URL (web address) of the image. It can be local or external.

The `alt` attribute provides an alternate text for the image (should describe the image), if the user for some reason cannot view it (slow connection, an error in the `src` attribute, or if the user uses a screen reader). The `width` and `height` attributes define the size of the image. These attributes are optional.

```

```

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

```
<a href="html_tutorial.pdf">  
    
</a>
```

To join an image with its caption use the `<figure>` tag:

```
<figure>  
    
  <figcaption>The official logo of UPC university</figcaption>  
</figure>
```

HTML lists

An unordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with bullets (small black circles) by default:

- First item
- Second item
- Third item

```
<ul>  
  <li>First item</li>  
  <li>Second item</li>  
  <li>Third item</li>  
</ul>
```

An ordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with Arabic numbers and started counting from 1 by default (`start` attribute can change it):

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

```
<ol>  
  <li>First item</li>  
  <li>Second item</li>  
  <li>Third item</li>  
</ol>
```

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

`type="A"` With uppercase letters

`type="a"` With lowercase letters

`type="I"` With uppercase roman numbers

`type="i"` With lowercase roman numbers

HTML5 navigation bars

We can define navigation bars mixing lists with links inside a `<nav>` tag:

```
<nav>
  <ul>
    <li><a href="#">Option 1</a></li>
    <li><a href="#">Option 2</a></li>
    <li><a href="#">Option 3</a></li>
  </ul>
</nav>
```

Or side bars with a `<aside>` tag:

```
<aside>
  <ul>
    <li><a href="#">Option 1</a></li>
    <li><a href="#">Option 2</a></li>
    <li><a href="#">Option 3</a></li>
  </ul>
</aside>
```

HTML 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.

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Score</th>
  </tr>
  <tr>
    <td>Marc</td>
    <td>Ripoll</td>
    <td>7.5</td>
  </tr>
  <tr>
    <td>Cèlia</td>
    <td>Romeu</td>
    <td>8.3</td>
  </tr>
</table>
```

Firstname	Lastname	Score
Marc	Ripoll	7.5
Cèlia	Romeu	8.3

Exercise: Create an HTML page explaining your personal skills and hobbies



We will learn to create an HTML page and publish it in a web server.

1. Create an HTML Web page named `index.html` explaining your personal knowledge, skills and hobbies using a text editor like Sublime. Use some of the previous HTML elements to define the structure of the Web page.

2. Open it in your favorite web browser (Open a File or Ctrl+O) to check it is displayed properly.

3. Configure your personal web space in ubiwan. Login to the remote server using this command:

```
ssh user@ubiwan.epsevg.upc.edu
```

By default your **user** is your DNI changing the first number to a letter (0 → a, 1 → b, 2 → c ...) and your password has 8 digits: day, month and year of your birthday and the last 2 digits of your DNI.

4. Follow the instructions described in <https://ubiwan.epsevg.upc.edu/>

```
mkdir $HOME/public_html  
chmod o+x $HOME/public_html  
chmod o+x $HOME
```

5. Copy the file to **ubiwan.epsevg.upc.edu** inside the **public_html** directory created in your home

```
scp index.html user@ubiwan.epsevg.upc.edu:public_html
```

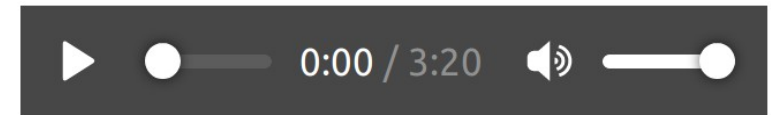
You can use other file transfer programs supporting sftp or scp transfer methods, e.g. file browser in *Ubuntu* or *Remote Explorer* extension in the *Visual Studio Code* editor.

6. Check if you can access your Web page using the URL <https://ubiwan.epsevg.upc.edu/~user>

HTML5 audios

Example with two alternative audio files; the browser will use the first recognized format:

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



Media types: MP3: audio/mpeg, OGG: audio/ogg, WAV: audio/wav

Attributes:

- autoplay: The audio starts playing as soon as it is loaded.
- controls: The audio displays the controls (play, pause, volume, ...).
- loop: The audio starts over again when finished.
- preload: The audio is loaded when the page loads.
- src: The URL of the audio file (when alternative audio files are not given).

HTML5 videos

Example with two alternative video files; the browser will use the first recognized format:

```
<video controls width="360" height="240">  
  <source src="movie.ogg" type="video/ogg" />  
  <source src="movie.mp4" type="video/mp4" />  
  Your browser does not support the video element.  
</video>
```

Media types: MP4: video/mp4, OGG: video/ogg, WebM: video/webm

Attributes:

- autoplay: The video starts playing as soon as it is loaded.
- controls: The video displays the controls (play, pause, volume, ...).
- loop: The video starts over again when finished.
- poster: Substitution image when video is not available.
- preload: The video is loaded when the page loads.
- src: The URL of the video file (when alternative video files are not given).



Add an audio of your favorite song and a video with the trailer of your favorite movie in your web page.

HTML forms

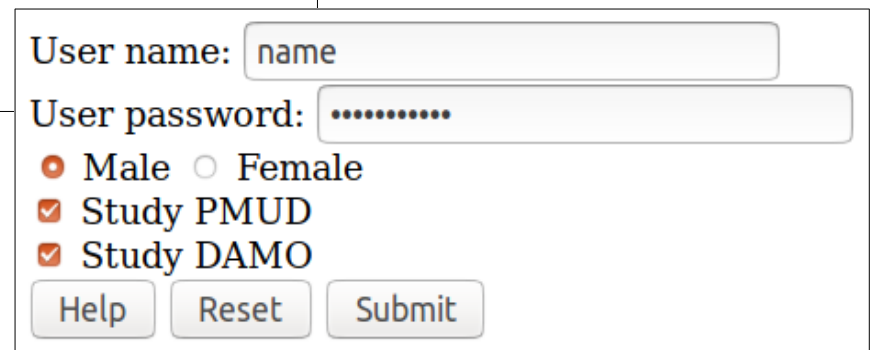
Forms are used to collect user input. Forms are built by grouping several `input` elements of different types into a `form` element. When the submit button is pressed, the field's values are sent to the *action* using the *method* (**get** or **post**, **get** by default).

```
<form action="action_url" method="post">
  User name:  <input type="text" name="username" value="name"><br>
  User password: <input type="password" name="password"><br>

  <input type="radio" name="gender" value="male" checked> Male
  <input type="radio" name="gender" value="female"> Female<br>

  <input type="checkbox" name="pmud" value="PMUD"> Study PMUD<br>
  <input type="checkbox" name="damo" value="DAMO"> Study DAMO<br>

  <input type="button" onclick="alert('Fill all the fields.')" value="Help">
  <input type="reset" value="Reset">
  <input type="submit" value="Submit">
</form>
```



User name:

User password:

☒ Male ☐ Female

☒ Study PMUD

☒ Study DAMO

HTML forms. Methods and input types

When submitting form data with **GET** method:

- The form data is appended into the URL in name/value pairs (don't use GET to send sensitive data).
- The length of a URL is limited (about 3000 characters).
- Useful for form submissions where a user wants to bookmark the result.

When submitting form data with **POST** method:

- The form data is included inside the body of the POST request.
- POST has no size limitations, and can be used to send large amounts of data.

<code><input type="text"></code>	Defines a one-line text input field
<code><input type="password"></code>	Defines a one-line text input field (typed chars converted into asterisks)
<code><input type="textarea"></code>	Defines a multi-line input field (a text area)
<code><input type="radio"></code>	Defines a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Defines a checkbox button (for selecting zero or more of many choices)
<code><input type="submit"></code>	Defines a submit button (for submitting the form)
<code><input type="button"></code>	Defines a clickable button
<code><input type="reset"></code>	Defines a clickable button that resets the form to initial values

HTML forms. Select input type

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

```
<select name="topic">
  <option value="html5">HTML5</option>
  <option value="css3">CSS3</option>
  <option value="javascript">javaScript</option>
  <option value="jquery">jQuery</option>
</select>
```

A single HTML5 select dropdown menu with a light gray border and a downward arrow on the right. The text "HTML5" is displayed in a dark gray font.

Add the **selected** attribute to the option to define a pre-selected option.

Use the **size** attribute to specify the number of visible values.


Use the **multiple** attribute to allow the user to select more than one value.

```
<select name="topic" size="4" multiple>
  <option value="html5">HTML5</option>
  <option value="css3" selected>CSS3</option>
  <option value="javascript">javaScript</option>
  <option value="jquery">jQuery</option>
</select>
```

A multiple HTML5 select menu with a light gray border. It displays four options: "HTML5", "CSS3", "javaScript", and "jQuery". The "CSS3" option is highlighted with an orange background, indicating it is the selected option.

HTML5 forms. New input types

`<input type="date">` Input field that should contain a date.



`<input type="datetime-local">` Input field that should contain a date and a time.

`<input type="email">` Input field that should contain an e-mail address (it can be validated).

`<input type="file">` Defines a "Browse" button for select a file to upload.



`<input type="month">` Allows the user to select a month and year.

`<input type="number">` Numeric input field.

`<input type="range">` Control for entering a number like a slider control. Default range is [0, 100].



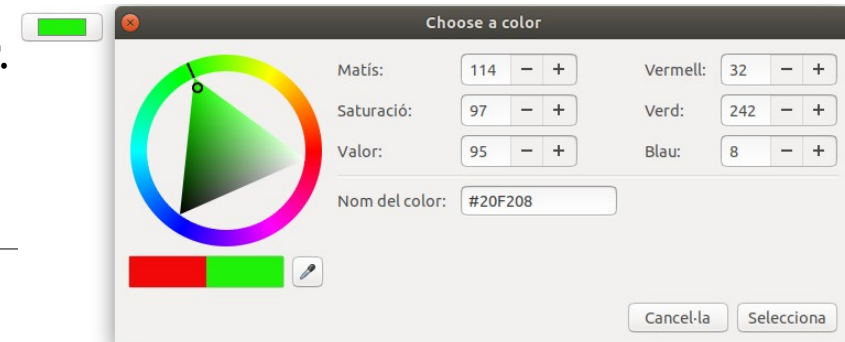
`<input type="search">` Search field (a search field behaves like a regular text field).

`<input type="tel">` Input field that should contain a telephone number.

`<input type="url">` Input field that should contain a URL address.

`<input type="week">` Allows the user to select a week and year.

`<input type="color">` Input field that should contain a color.



HTML5 forms. Input restrictions

Attribute	Description
checked	Specifies that an input field should be pres-elected (for 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 character 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

HTML forms. Grouping input fields

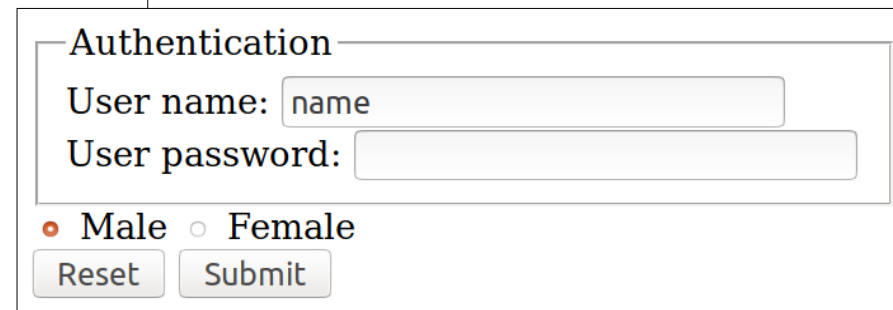
The `<fieldset>` tag is used to group related elements in a form.

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

```
<form action="action_url" method="post">
  <fieldset>
    <legend>Authentication</legend>
    User name: <input type="text" name="username"><br>
    User password: <input type="password" name="password"><br>
  </fieldset>

  <input type="radio" name="gender" value="male" checked> Male
  <input type="radio" name="gender" value="female"> Female<br>

  <input type="reset" value="Reset">
  <input type="submit" value="Submit">
</form>
```



Add in your web page a form to collect the personal information of people who want to be your friends. You could ask about their name, gender, birthday date, email, telephone, the spoken languages (in a multiselection), the favorite color, a description of their hobbies (in a textarea), ...

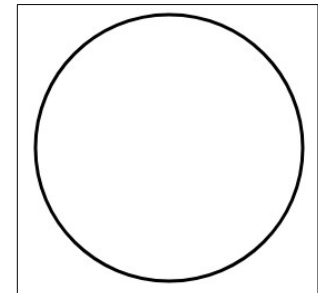
Submit the form to <http://httpbin.org/post> with POST method to check the data is properly sent.

HTML5 graphics. SVG & Canvas

SVG (Scalable Vector Graphics) is a language for describing 2D graphics in XML. If attributes of an SVG object are changed, the browser can automatically redraw the shape. SVG images could be created with vector graphics programs, like Inkscape, Adobe Illustrator or [Vector Paint](#).

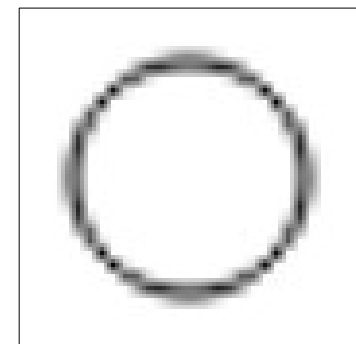
SVG tags: `line`, `circle`, `ellipse`, `rect`, `polygon`, `polyline`, `text`, ...

- Resolution independent
- Support for event handlers
- Best suited for applications with large rendering areas
- Slow rendering if complex
- Not suited for game applications



Canvas draws 2D graphics on the fly (with JavaScript). Canvas defines a rectangular container where JavaScript draws the graphics pixel by pixel. Once the graphic is drawn, it is forgotten by the browser.

- Resolution dependent
- No support for event handlers
- Poor text rendering capabilities
- You can save the resulting image as .png or .jpg
- Well suited for graphic-intensive games



HTML5 graphics. SVG

Example using absolute lengths: px (pixels, by default), cm, mm, in (inches), pt (points), pc (picas).

```
<svg height="150" width="400">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:blue"/>
      <stop offset="100%" style="stop-color:lime"/>
    </linearGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)"/>
  <text fill="white" font-size="45" font-family="Verdana" x="150" y="86">SVG</text>
  Your browser does not support inline SVG.
</svg>
```



Example using relative lengths: % (relative to parent element), vw (relative to the viewport's width), vh (viewport's height), em (font-size of the element), rem (font-size of the root element), ...

```
<svg width="100%" height="40vw" >
  <ellipse cx="50%" cy="50%" rx="40%" ry="40%" fill="blue"/>
  <text fill="white" font-size="45" font-family="Verdana" x="50%" y="50%"
    dominant-baseline="middle" text-anchor="middle">SVG</text>
  Your browser does not support inline SVG.
</svg>
```



HTML5 graphics. SVG external file

The SVG can be defined in an external file and can be inserted in the Web page with the `img` object, `embed` or `iframe` tags:

```

```

The SVG could be created by hand with a text editor or with a vector graphics program like Inkscape or [Vector Paint](#). The attribute `xmlns="http://www.w3.org/2000/svg"` must be added to the `svg` tag:

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100">  
  <polygon points="10,90 90,90 50,10" fill="green"/>  
</svg>
```

The optional `viewbox` attribute defines the viewport or area occupied by the graphic.

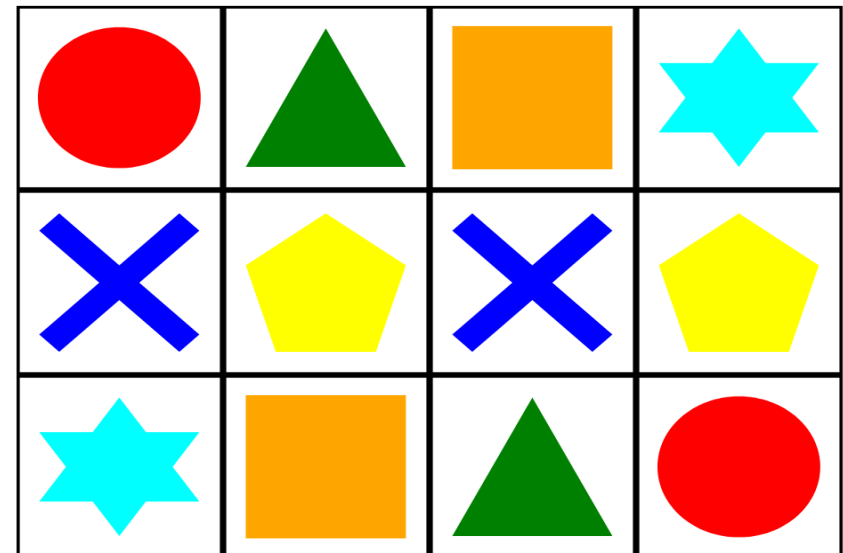
Exercise: Create an HTML page containing a memory game

We want to create a classic memory game: There are several pairs of cards with the same symbol turned face down. You have to reconstruct the pairs of cards by turning them 2 by 2. If the pair of cards turned face up are the same you win and the pair disappears from the game, otherwise the cards are automatically turned face down and you need to make a new try. Find all the pairs in the least moves and in a shortest time.

You can see an example: <https://www.memozor.com/memory-games/for-kids/stencil-4>

We start creating game board with 12 cards with their symbols. The symbols could be figures defined with SVG. The web page must be responsive: It will adjust automatically to the size of the viewport (the space in the web browser where the web page is rendered) so it fits several screen sizes.

Save your source code in `memory_game.html` file. Upload it to the `public_html` folder in the `ubiwan.epsevg.upc.edu` server.



Exercise: Online HTML quiz & HTML exercises

You can review your knowledge about HTML doing these online resources:

- [HTML quiz](#)
- [HTML exercises](#)