UNIT 6

HTML + CSS

Markup Languages and
Information Management Systems

ies SERPIS

INDEX (part I)

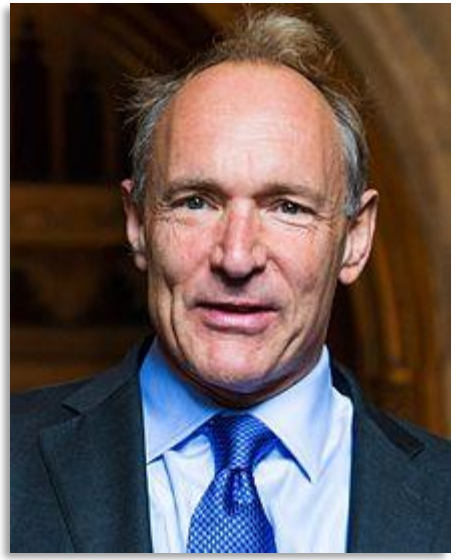
1. INTRODUCTION

UNIT 6 HTML + CSS, PART I

Introduction

- A web page is a plain text document written in some markup languages.
- It contains marks, or tags, that allow you to:
  - Modify the presentation.
  - Include images and other multimedia items.
  - Create links
  - ...
- Usually displayed in a web browser, who acts as user-agent and interprets and presents the document.

Introduction



Tim Berners-Lee

- "HTML": HyperText Markup Language
- "Hypertext": makes it possible to follow information in a non-sequential manner through the links that the various elements contain.
- It is the most important web page creation language. Its invention led to the appearance, development and expansion of the WWW.
- It is the display standard adopted by all the web browsers.

Introduction

- To create a web page, you only need:

Plain text editor

Web browser

- However, you could use more advanced software: Visual Studio Code, Adobe Dreamweaver...
- The created file must have a .html or .htm extension.

2. GENERAL CHARACTERISTICS

UNIT 6 HTML + CSS, PART I

General Characteristics

- In general, elements should have an opening and a closing tag.
- But some elements can work with the opening tag only. Others do not have a closing tag.
- Empty elements do not require a closing tag.
- Tag names are reserved words that the browser interprets.
  - You cannot create your own tag names.
- Not case-sensitive.

optional

```
<tagName [attributes]>
    …
<p>another_paragraph</p>
    …
</tagName>
```

ELEMENTS

General Characteristics

- Attributes are optional parameters that modify the effect that the elements provoke.
- If used, always in the opening tag.
- If the assigned value has spaces, it must be quoted
  - Single or double quotes.
- Else, quotation marks are optional

```
<tagName attrib1=value1
attrib2="value2 has spaces">

  …

</tagName>
```

ATTRIBUTES

General Characteristics

- We start with stripped HTML, no additions.
- HTML allows to have content, style and programming code within.
  - Difficult to maintain and update.
- CSS enables separation between content and presentation.

VERSIONS

General Characteristics

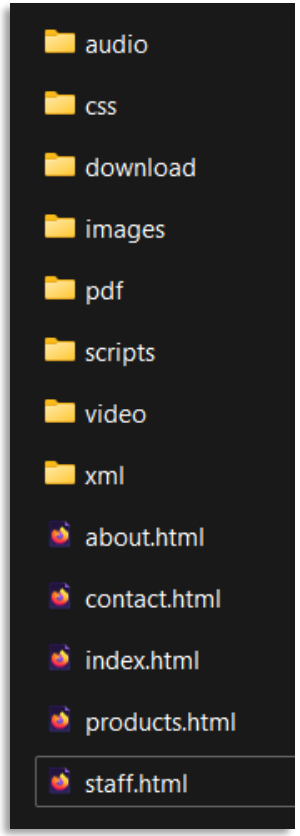
VALIDATION

- Validation here only refers to syntactic correction.
- W3C has both HTML and CSS validators you can use to check your code.
  - Works with pasted code, uploaded files and online files.
- Green, yellow and red warnings.
- A valid HTML document is a quality HTML document.

File Organization

ORGANIZATION

- A website can contain a huge amount of files.
- It is convenient to organize the project in folders:
  - Put HTML documents in the project root folder.
  - Store resources in separated folders within the project attending to their type: a folder for audio clips, another one for video clips, another one for CSS...
  - If a section of the site is big enough, you can create a folder to store the files related to this microsite.
    - It would contain its own audio folder, css folder, ...



3. STRUCTURE

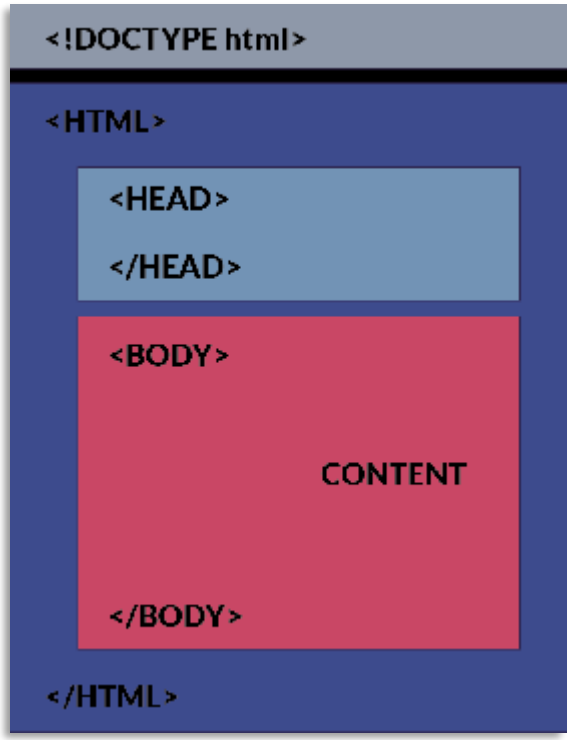
UNIT 6 HTML + CSS, PART I

Structure

- The first line in an HTML document identifies it as such.

```
<!DOCTYPE html>
```

  - This line tells the user-agent (web browser) the kind of document it is opening.
  - Not an HTML tag.



BASICS

Structure

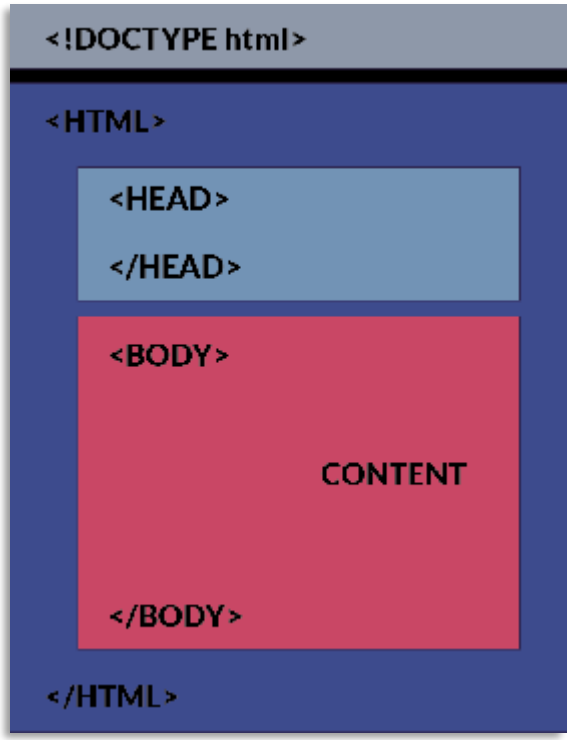
- The rest of the document is wrapped within an `<html>` root element.
- The language of the page can be declared using its `lang` attribute, whose value is a two-letter code from ISO 639 set 1.
- The head and the body of the document are usually found inside this root.

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

BASICS

Structure

- An HTML document consists of a <head> and a <body>.
  - The <head> carries information about the document.
  - The <body> carries the content displayed on the page.



BASICS

Structure

- Within the <head> and </head> tags you can find:
  - Metadata of the document
  - Information tags
  - `<title>` (mandatory)
    - Only element in the <head> that is displayed.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> …… </title>
        ……
</head>
<body>
        ……
</body>
</html>
```

HEAD

Structure

HEAD

- Metadata
  - `<meta>` tag
  - Many types, various purposes.
  - Example: search engines use them to find related sites.
- Some usual metadata:
  - `<meta charset="UTF-8">`
    Indicates the character set used on the page.
  - `<meta name="description" content="Free Web Tutorial">`
    A description of the page.
  - `<meta name="keywords" content="HTML5, CSS3, XML, JavaScript">`
    Defines key words of the page, useful for search engines.
- `<link>` tag: Used to link style sheets, JavaScript code...

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

Structure

- The <body> carries all the content displayed on the page.
- It is the bulk of the document's code:
  - Organization
  - Text formatting
  - Images
  - Links
  - ...

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> …… </title>
        ……
</head>
<body>
        …
</body>
</html>
```

BODY

Structure

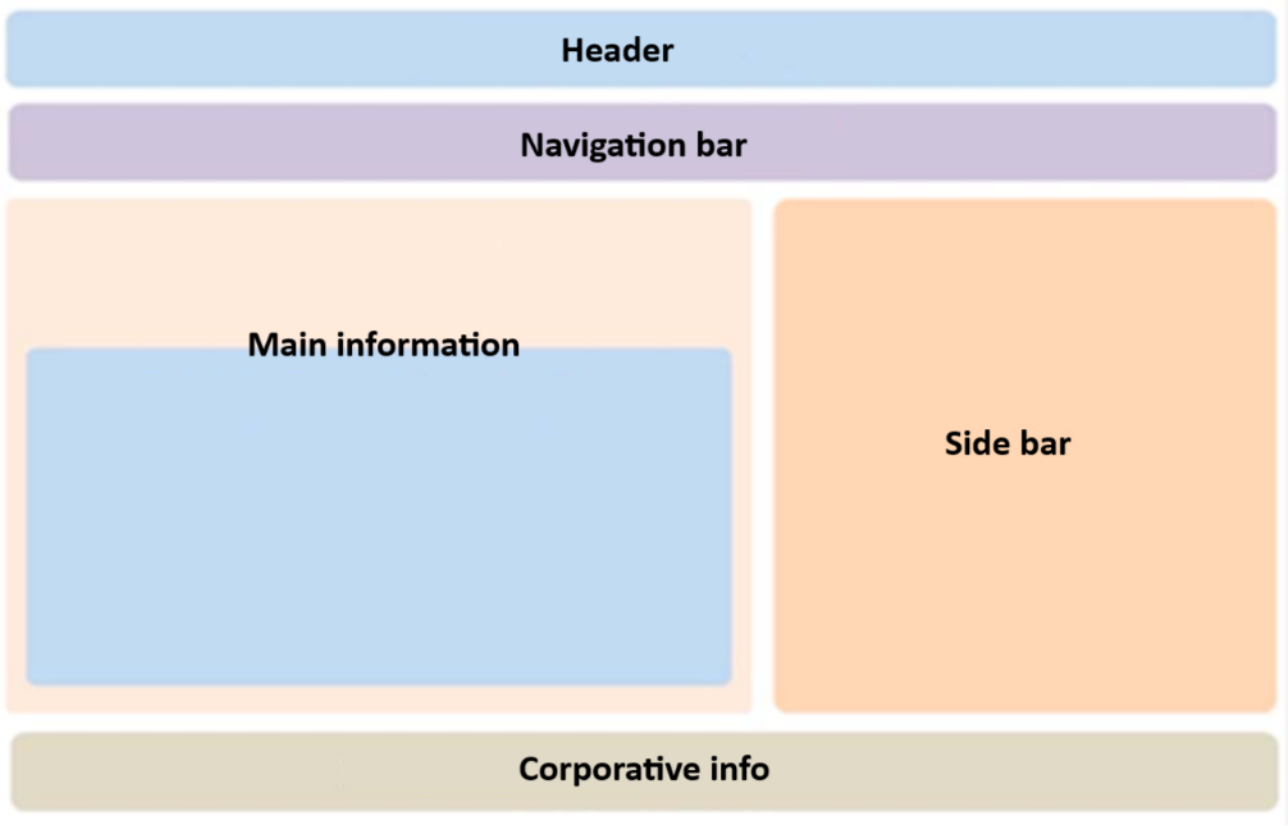
BODY

- In the <body> of an HTML document, two types of elements can be found:
  - Block elements:
    - Always start on a new line and cause a line break when closed.
    - Occupy the entire width of their parent.
  - Inline elements:
    - Neither start on a new line nor cause a line break when closed.
    - Are used to highlight parts within a block.

- Block elements can contain both block and inline blocks.

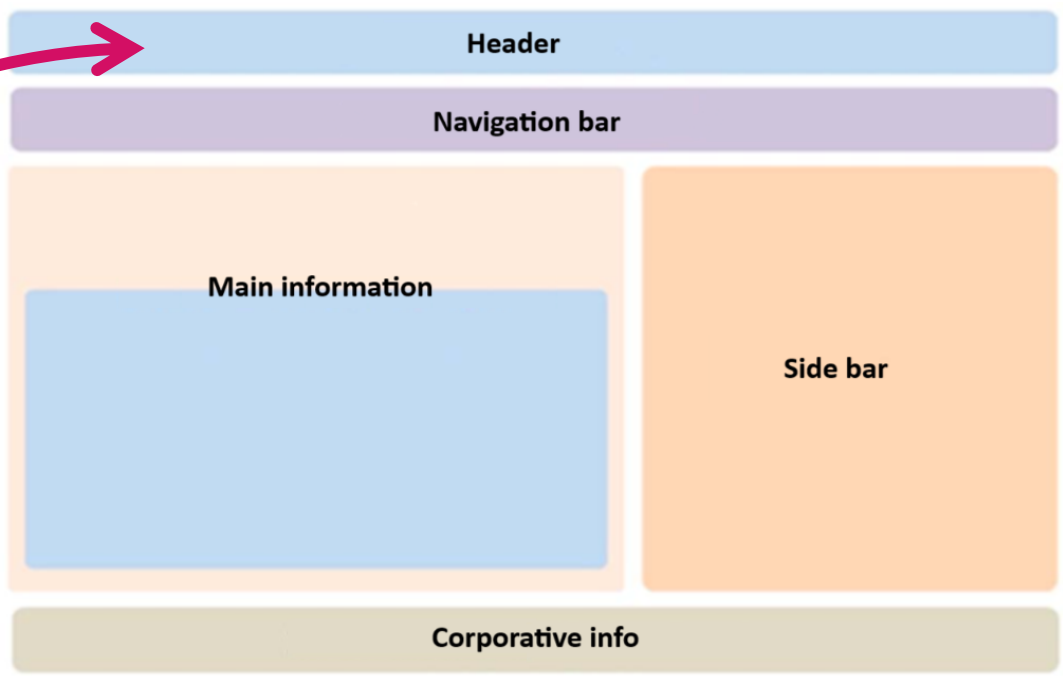
Structure

BODY



Throughout several practical exercises you will work to achieve web pages with this look.

Structure



The following items will be displayed in the header of the pages:

- Logotype
- Title
- Subtitle(s)
- Brief description

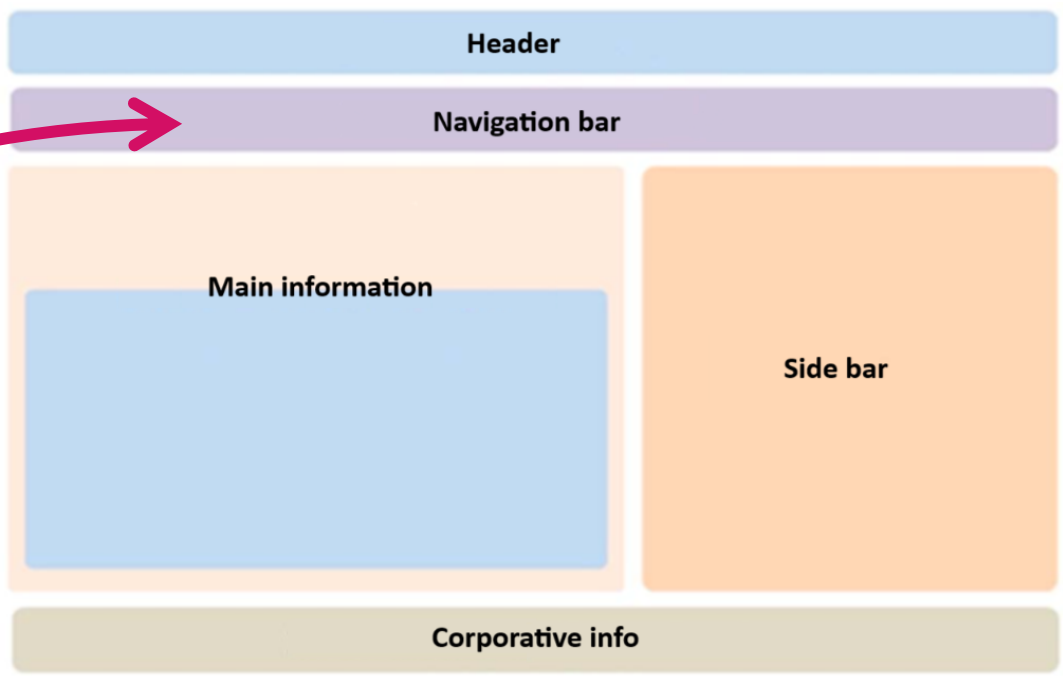
(Not to be mistaken for the <head>)

BODY

Structure

In the navigation bar:

- Menu or list of links.
- Navigation within the site

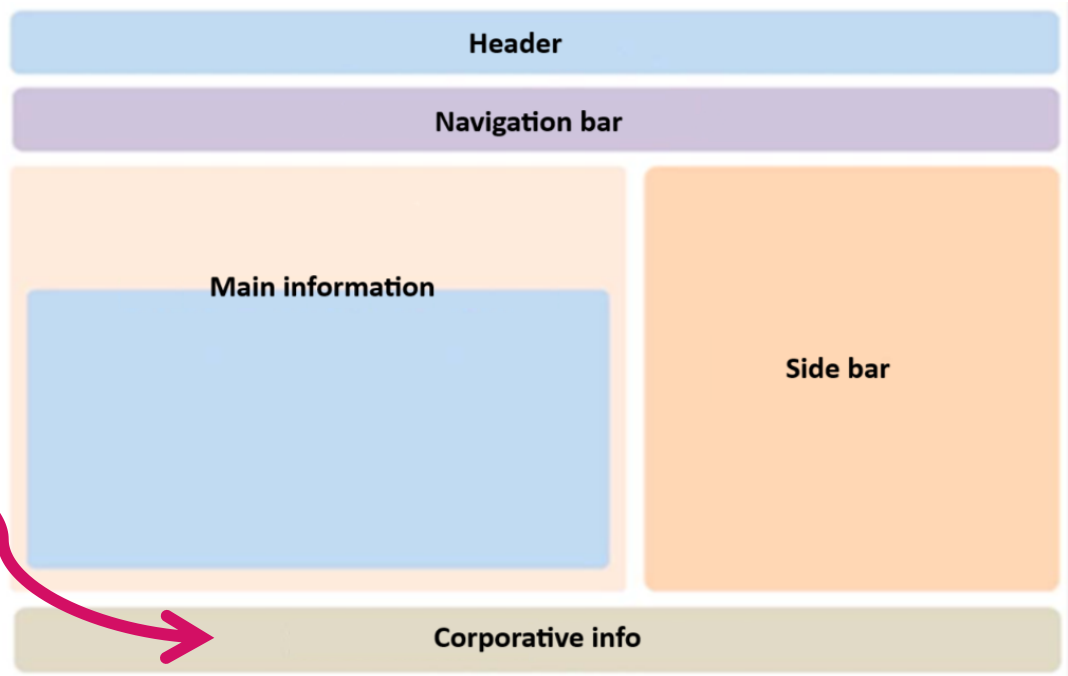


BODY

Structure

In the footer:

- Information about the web site.
- Author or company.
- Links to rules, terms and conditions.

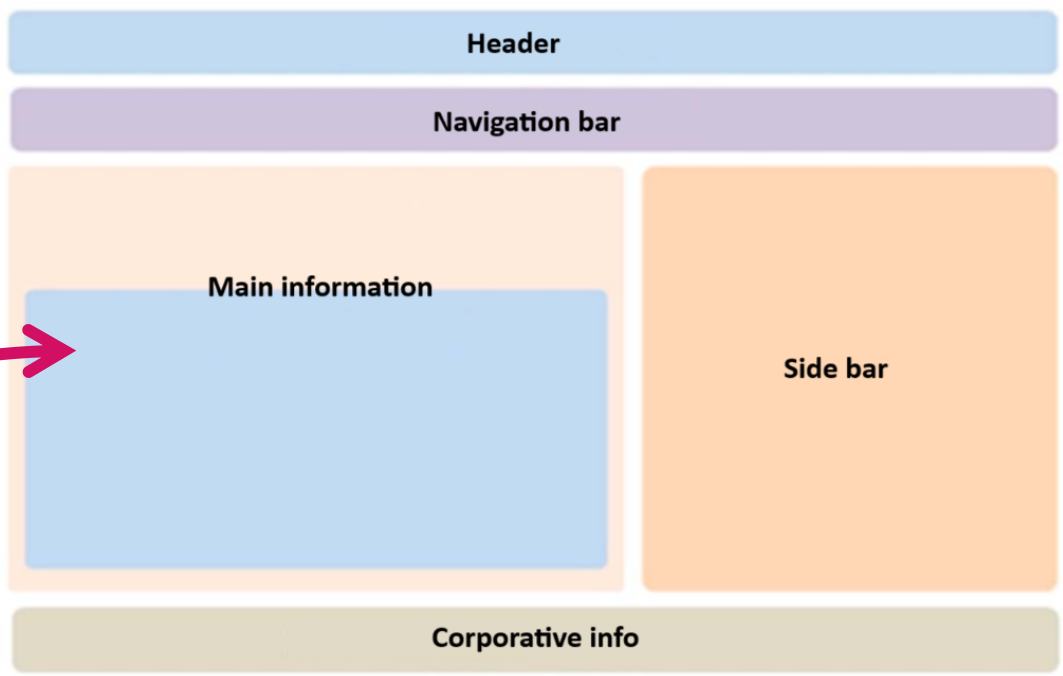


BODY

Structure

The central area:

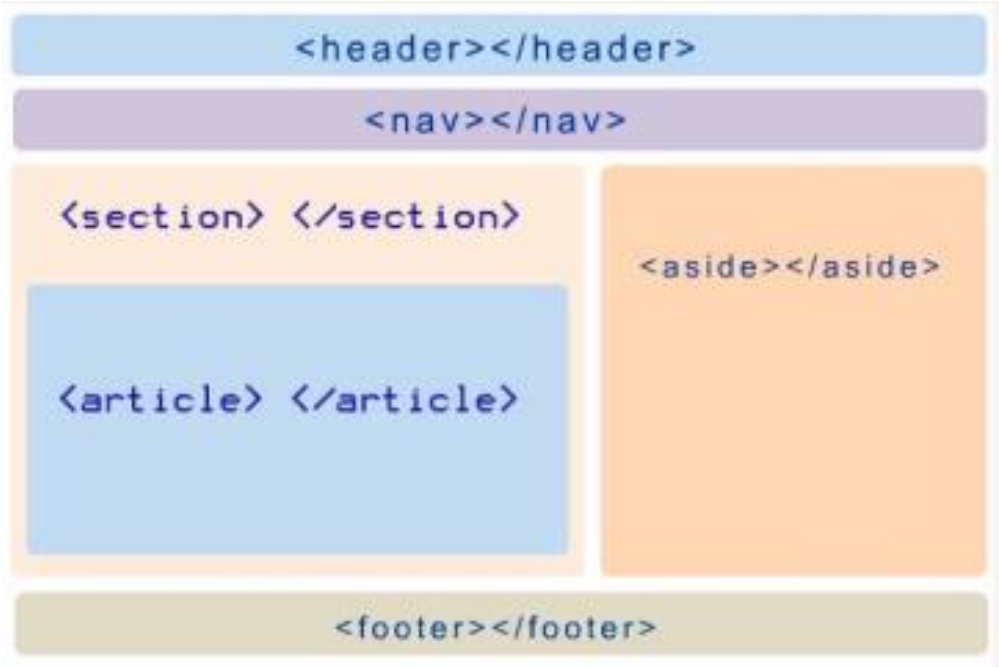
- Is the most relevant block for content.
- Somehow divided into rows and columns.
- Side bar.
- Great flexibility.



BODY

Structure

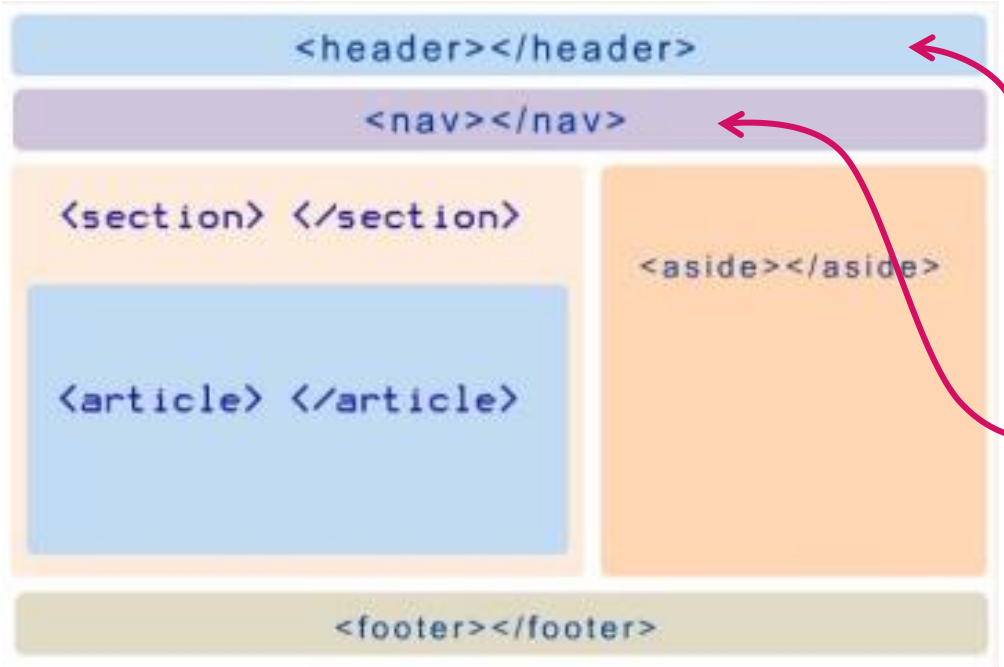
BODY



- Up to HTML4, structuring such a design was complex.
  - Use of `<div>` tags (all-purpose block element).
- New tags introduced in HTML5
  - Virtually the same as `<div>`, but with meaningful names.

Structure

BODY



<header>

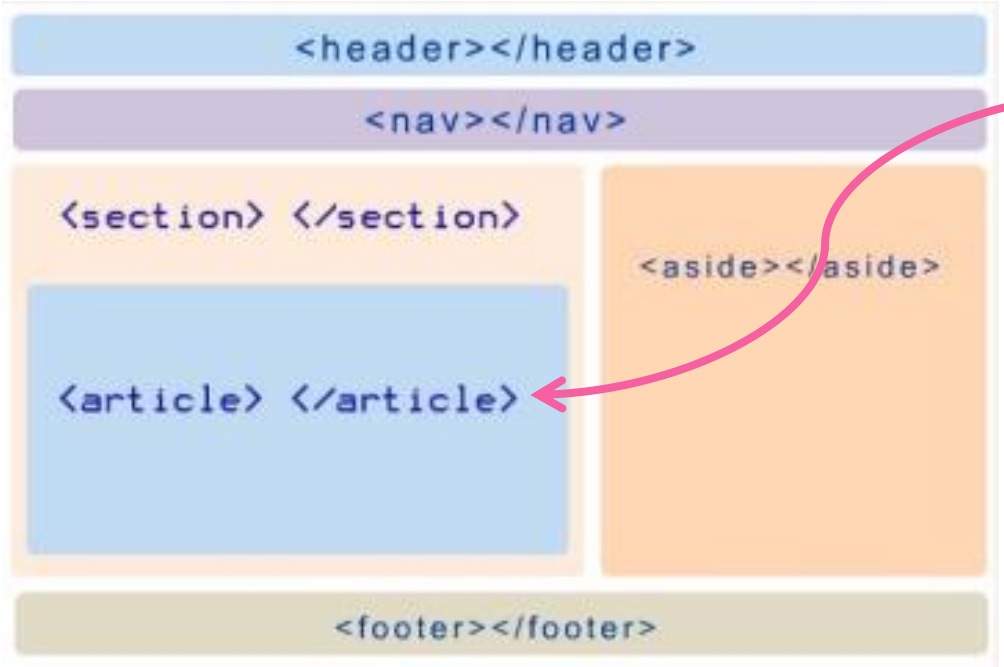
- For introductory items

<nav>

- For navigation links within the site

Structure

BODY

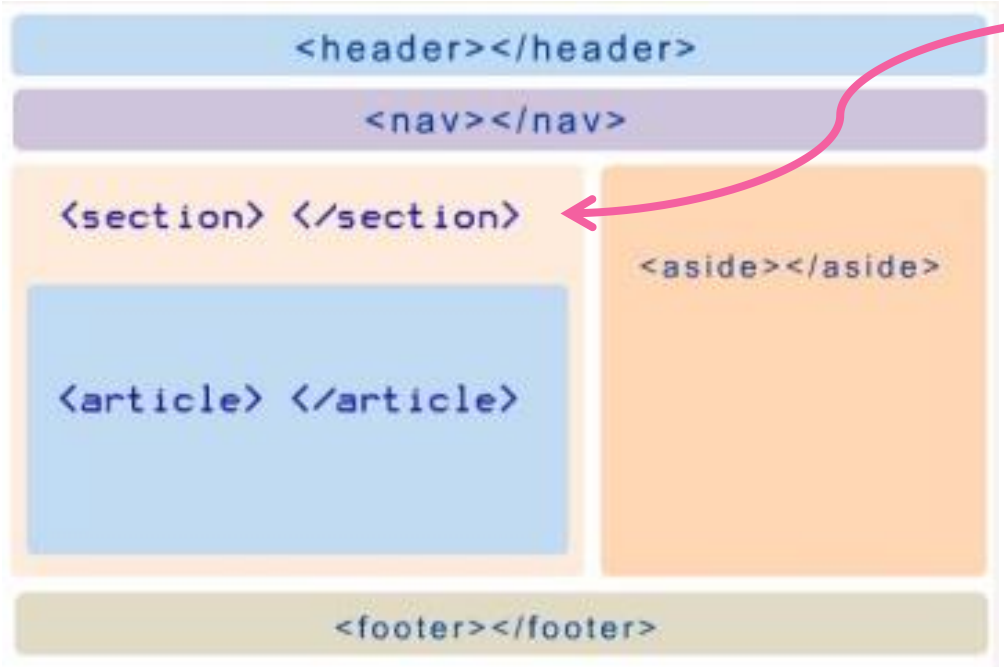


<article>

- Page component that consists of a self-contained composition.
  - Article in a forum
  - Article in a magazine or newspaper
  - Blog entry
  - User comment

Structure

BODY



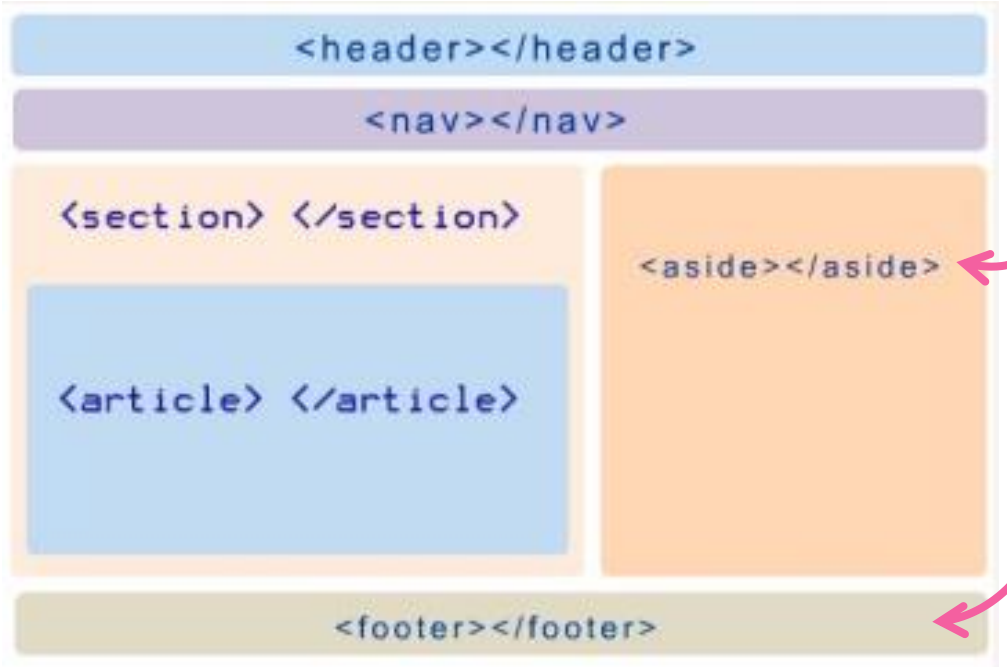
<section>

- Represents a section within a document.
  - Like the chapter of a book.
  - Can contain subsections.
  - Used to establish a hierarchy with the content.

It favors SEO
(search engine optimization).

Structure

BODY



<aside>

- Section with related content
  - Side bars
  - Advertisement

<footer>

- Lower section of the page
  - "About" information
  - Has usually little to do with the content of the page.

4. TEXT AND COMMENTS

UNIT 6 HTML + CSS, PART I

Working guidelines

If this is the first time you face HTML, here are some guidelines and advice to practice with the code fragments found in the slides:

- Open your favorite plain text editor and a web browser.
- Write the code or paste it into the editor.
- Save the file using `.html` or `.htm` extension.
- Open the file in the browser to see the result.
- Please remember that an HTML file must have some mandatory tags: `<!DOCTYPE>`, `<html>`, `<head>` and `<body>`.
- If the given code does not have `<html>`, `<head>` or `<body>`, you should write (paste) it within the `<body>` element.
- Do not forget to save changes in the file after modifying it.
- You can reload the page in the browser by pressing the refresh button or the F5 key.

Text

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> …… </title>
      ……
</head>
<body>
      Hello world
</body>
</html>
```

- To make the browser display some text, just write the text within the body of the HTML document.
  - Untagged

Text

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> …… </title>
      ……
</head>
<body>
      Hello  world
</body>
</html>
```

- HTML ignores all white spaces after the first one.
  - That is, if there are more than one space between two words, only one space will be displayed.
- Use the ` ` entity to forcefully display an extra space.

CONSIDERATIONS

Text

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> …… </title>
      ……
</head>
<body>
      Hello<br>world
</body>
</html>
```

- HTML also ignores line breaks within the body.
- To force them, use `<br>`
  - Does not have a closing tag.

CONSIDERATIONS

Text

- The content of a `<pre>` element is displayed in the web browser as is, keeping all white spaces, tabs and line breaks.
  - Uses a monospace font

```
<body>
  <pre>
This    text
        will be displayed with
the tabs, white spaces and
line   breaks right, as is.
  </pre>
</body>
```

```
This    text
        will be displayed with
the tabs, white spaces and
line   breaks right, as is.
```

Text

- The <p> tag defines paragraphs by inserting a special line break.
  - Makes a bigger line break than <br>.

This code is written in one line. Why not?

```
<body>
  First line<br>Second line
  (preceded by &lt;br&gt;)<p>
  Third line (preceded by
  &lt;p&gt;)
</body>
```

Do you remember character entities from XML?

First line
Second line (preceded by <br>)

Third line (preceded by <p>)

Text

- We will learn how to modify the typography of a text, the font size, color… later with CSS.
- There are some font formatting tags that are commonly used in HTML:
  - Bold <b>
  - Italics <i>
  - Underlined <u>
  - Subindex <sub> and superindex <sup>
  - Code <code>
- They are *inline* elements.
- They need to be closed to delimit the affected text.

```
<body>
  <b>bold</b>, <i>italics</i>,
  <u>underlined</u>, m<sup>2</sup>,
  DNA<sub>2</sub><br><br>
  <code>
    for (int i=0; i<3; i++) {<br>
      System.out.println("Hello
world");<br>
    }
  </code>
</body>
```

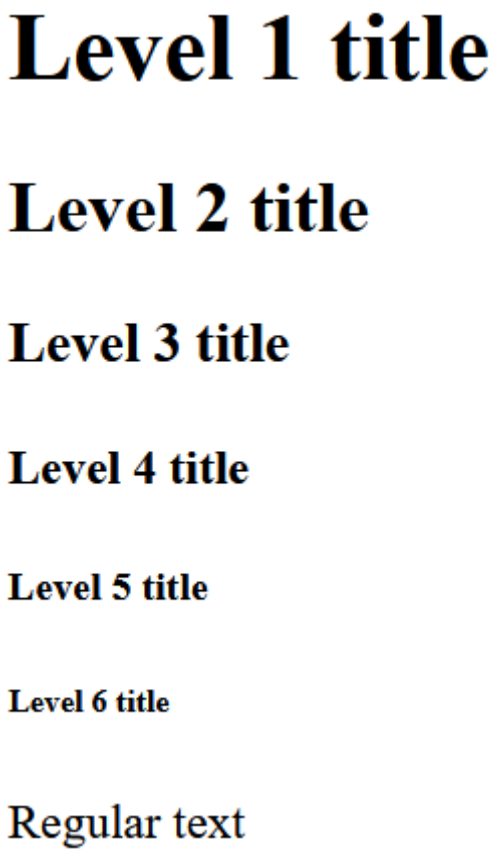
bold, *italics*, <u>underlined</u>, m^2, DNA_2

```
for (int i=0; i<3; i++) {
System.out.println("Hello world");
}
```

Text

- There are some default styles for section titles.
  - Up to six hierarchical levels.
  - The actual style depends on the web browser.
- Elements <h1>, <h2>, <h3>, <h4>, <h5> and <h6>
  - <h1> has the biggest size and should be reserved for the page title.
  - The rest have a decreasing size order.

```
<body>
  <h1>Level 1 title</h1>
  <h2>Level 2 title</h2>
  <h3>Level 3 title</h3>
  <h4>Level 4 title</h4>
  <h5>Level 5 title</h5>
  <h6>Level 6 title</h6>
  Regular text
</body>
```



Text

- HTML has some forbidden characters because its tags use them.

- HTML entities must be used instead, if required.
  - Entities begin with & and end with semicolon ;
  - Entity names and entity numbers.
    - Numbers correspond to Unicode characters and can be both decimal or hexadecimal.
  - Decimal entity numbers are preceded by #
  - Hexadecimal entity numbers are preceded by #x
- Example: the less-than character (<) can be escaped with:
  - `<`
  - `<`
  - `<`

See the full list

Text

- In the past, characters not included in the English alphabet caused display problems in browsers.

- Nowadays, these problems are inexistent when the `<head>` includes this `<meta>` tag:

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

Comments

- Useful for:
  - Including warnings and notes within the code.
  - Delimiting start and end of code sections.
  - *Hiding* code from the user-agent.
  - ...
- Comments are not displayed on screen.
  - But are kept with the code *(watch out for sensible info)*
- Cannot be nested
- Can have multiple lines
- Can be found both in <head> and <body>

```
<!-- THIS IS A COMMENT-->
```

Text

PRACTICAL EXERCISE 6.1

5. CSS SELECTORS

UNIT 6 HTML + CSS, PART I

CSS Selectors

- "Stylesheet": set of style rules that tell the browser how to display the document.

- CSS *(Cascading StyleSheets):* markup language composed of display rules (selectors and declarations).

CSS Selectors

CSS rule = Selector + declaration

- Selector: selects which HTML elements the rule applies to.
- Style or declaration:
  - Specifies the styles applied to the selector.
  - Contains one or more CSS properties.

CSS Selectors

Rule syntax:

```
selector {property: value;}
```

- Property: allows modification of one of the selector's features.
- Value: states the new value for that feature

```
adding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
der: 12px
```

CSS Selectors

SELECTORS

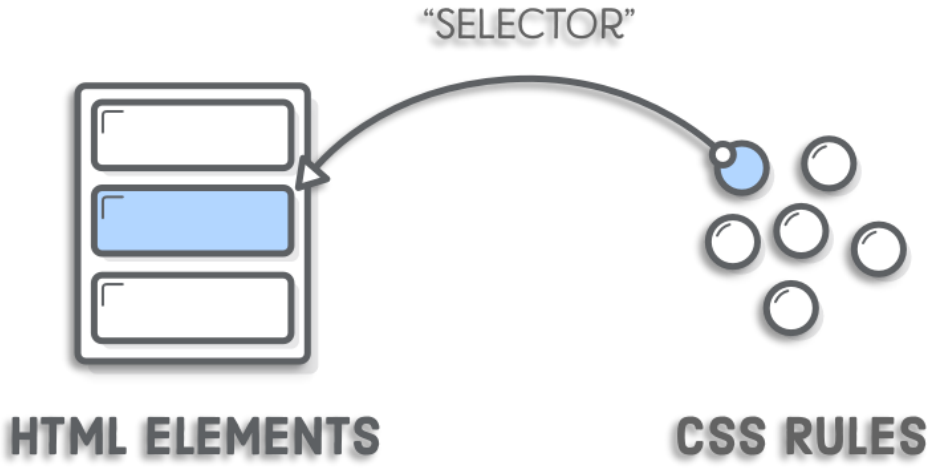
- CSS code is usually written in .css files, separated from the HTML code.
- CSS files do not require any prologue; just the rules.
- The HTML code must link to the CSS file for the styles to apply.
  - `<link>` tag

```
<!DOCTYPE HTML>
<HTML lang="es">
  <HEAD>
    <meta charset="UTF-8">
    <title>CSS TEST</title>
    <link rel="stylesheet" href="css/styles.css">
  </HEAD>
  <BODY> … </BODY>
</HTML>
```

CSS Selectors

Selectors: basic types

- Universal
- Element
- Combinator
- Class
- Identifier



SELECTORS

CSS Selectors

```
*{
  margin:0;
  padding:0;
  }
```

- Universal selector:
  - Asterisk.
  - Selects all the elements on the page.

SELECTORS

CSS Selectors

```
h1, h2, h4 {
  font-size: 12em;
  color: #093;
}
```

- Element selector:
  - Selects all elements with matching tag.
  - The same rule can be applied to different elements.
    - Separated by commas in the selector

SELECTORS

CSS Selectors

```
p span {
  font-weight: bold;
}
```

This fragment means: "To every element descending from a <p> element, apply bold font style."

- Combinator selector:
  - There are several subtypes.
  - You will learn now the descendant subtype, and the rest later on.
- Descendant combinator selector:
  - Selects elements contained within other elements.
    - Any level of ancestry (not just parent-child).
  - Could be a longer descendant chain.

SELECTORS

CSS Selectors

SELECTORS

Class selector:

- Very frequently used.
- Allows to select every element on the page with a matching class attribute.
- The selector consists of a dot and the name of the desired class attribute.
- Can be combined with other selectors.

- The value of the class attribute can be shared by multiple elements in the document.

CSS Selectors

SELECTORS

```
<BODY>
  <p>First paragraph</p>
  <p class="special">Second paragraph</p>
  <p>Third paragraph</p>
</BODY>
```

HTML code

```
.special {
  color:red;
}
```

CSS code

First paragraph

Second paragraph

Third paragraph

CSS Selectors

SELECTORS

```
<BODY>
  <h1 class="special">PAGE TITLE</h1>
  <p>First paragraph</p>
  <p class="special">Second paragraph</p>
  <p>Third paragraph</p>
</BODY>
```

HTML code

```
.special {
  color:red;
}
```

CSS code

PAGE TITLE

First paragraph

Second paragraph

Third paragraph

CSS Selectors

SELECTORS

```
<BODY>
  <h1 class="special">PAGE TITLE</h1>
  <p>First paragraph</p>
  <p class="special">Second paragraph</p>
  <p>Third paragraph</p>
</BODY>
```

HTML code

```
p.special {
  color:red;
}
```

CSS code

This fragment means: "To every 'special' class <p> element, apply red font color."

PAGE TITLE

First paragraph

Second paragraph

Third paragraph

CSS Selectors

SELECTORS

Identifier selector:

- Very frequently used.
- Allows to select elements by the value of its id attribute.
- The selector consists of a hash (#) and the value of the desired id attribute.
- Can be combined with other selectors.

- The value of the id attribute must be unique within the document.

CSS Selectors

SELECTORS

```
<BODY>
  <h1 class="special">PAGE TITLE</h1>
  <p>First paragraph</p>
  <p id="special">Second paragraph</p>
  <p>Third paragraph</p>
</BODY>
```

HTML code

```
#special{
  color:red;
}
```

CSS code

PAGE TITLE

First paragraph

Second paragraph

Third paragraph

CSS Selectors

SELECTORS

Selector combination

- Multiple selectors, even of different types, can be combined to restrict the scope.

```
.important .special { color: red; }
```

Means *"special" class elements within "important" class elements.*

```
div.important span.special { color: red; }
```

Means *"special" class elements within "important" class elements.*

CSS Selectors

Combinator selector: advanced types

- Child combinator
- Next-sibling combinator
- Subsequent-siblings combinator
- Attribute combinator

- Note: incompatible with very old browsers

SELECTORS

CSS Selectors

SELECTORS

Child combinator selector (>):

- The rule only applies to direct children.

```
div > p {
  color:red;
}
```

CSS code

```
<BODY>
<div>
  <p>Paragraph in the div.</p>
  <section>
    <p>Paragraph in the section in the div.</p>
  </section>
  <p>Paragraph in the div.</p>
</div>
</BODY>
```

HTML code

Paragraph in the div.

Paragraph in the section in the div.

Paragraph in the div.

CSS Selectors

SELECTORS

Next-sibling combinator selector (+):

- The rule only applies to the next sibling (the next element with the same parent, declared below, adjacent).

```
<BODY>
  <h1>PAGE TITLE</h1>
  <p>First paragraph.</p>
  <p>Second paragraph.</p>
  <p>Third paragraph.</p>
</BODY>
```

HTML code

```
h1 + p {
  color:red;
}
```

CSS code

PAGE TITLE

First paragraph.

Second paragraph.

Third paragraph.

CSS Selectors

SELECTORS

Press AltGr + 4 on your keyboard to type this tilde.

Subsequent-siblings combinator selector (~):

- The rule applies to all the subsequent siblings (elements with the same parent, declared below).

```
<BODY>
<ul>
  <li>The Secret of Monkey Island</li>
  <li>Monkey Island 2: LeChuck's Revenge</li>
  <li id="mi3">The Curse of Monkey Island</li>
  <li>Escape from Monkey Island</li>
  <li>Tales from Monkey Island</li>
  <li>Return to Monkey Island</li>
</ul>
</BODY>
```

HTML code

```
#mi3 ~ li {
  background: gold;
  color: maroon;
}
```

CSS code

- The Secret of Monkey Island
- Monkey Island 2: LeChuck's Revenge
- The Curse of Monkey Island
- Escape from Monkey Island
- Tales from Monkey Island
- Return to Monkey Island

CSS Selectors

SELECTORS

Attribute combinator selector ([ ]):

- Allows to select elements based on their attributes and, optionally, the values they take.
- The name of the attribute is written in square brackets.
- Relational operators can be used to specify attribute values:
  - = means "equals"
  - ~= means "any of its values equals"
  - ^= means "begins with"
  - $= means "ends with"
  - *= means "contains"

CSS Selectors

SELECTORS

```
<BODY>
  <h1 class="potato">PAGE TITLE</h1>
  <p>First paragraph.</p>
  <p class="potato">Second paragraph.</p>
  <p class="onion">Third paragraph.</p>
</BODY>
```

HTML code

```
[class] {
  background: black;
  color: white;
}
```

CSS code

PAGE TITLE

First paragraph.

Second paragraph.

Third paragraph.

```
p[class] {
  background: black;
  color: white;
}
```

CSS code

PAGE TITLE

First paragraph.

Second paragraph.

Third paragraph.

CSS Selectors

SELECTORS

```
<BODY>
  <h1 class="potato">PAGE TITLE</h1>
  <p>First paragraph.</p>
  <p class="potato">Second paragraph.</p>
  <p class="onion">Third paragraph.</p>
</BODY>
```

HTML code

PAGE TITLE

First paragraph.

Second paragraph.

Third paragraph.

```
[class="potato"] {
  background: black;
  color: white;
}
```

CSS code

This selector is equivalent to `.potato`
The attribute combinator is more useful when used with attributes other than `class` *and* `id`

Quotation marks are optional

CSS Selectors

SELECTORS

```
<BODY>
  <h1 class="potato">PAGE TITLE</h1>
  <p>First paragraph.</p>
  <p class="potato, broccoli">Second paragraph.</p>
  <p class="onion">Third paragraph.</p>
</BODY>
```

HTML code

```
[class~="broccoli"] {
  background: black;
  color: white;
}
```

CSS code

Try this example again but replacing '~=' for '=': none of the elements would get the style applied.

PAGE TITLE

First paragraph.

Second paragraph.

Third paragraph.

6. COLORS

UNIT 6 HTML + CSS, PART I

Colors

- Colors are essential on any web page.
  - They are applied to background, elements, text...

- "Additive Color Synthesis": getting color by projecting light of three basic colors (red, green and blue) → RGB
- Using 8 bits to code the intensity of each color:

$$2^8 = 256 \text{ levels}$$

- Possible RGB combinations:

$$256 \times 256 \times 256 = 16777216 \text{ colors}$$

Commonly referred to as "16 million colors"

Colors

Hexadecimal encoding

- Two HEX digits are needed to encode 8 bits
  - Two for red, two for green, two for blue → 6 HEX
- Pure red: #FF0000
- Pure green: #00FF00
- Pure blue: #0000FF
- Other colors: #RRGGBB combinations
  - Black: **#000000**
  - White: #FFFFFF
- Two more tailing HEX digits can be used to indicate transparency / opacity (alpha) → #RRGGBBAA
  - Values between 00 (transparent) and **FF** (opaque)

When alpha is omitted, it is opaque by default.

Colors

Aliases:

- 16 color aliases were introduced in HTML version 3.2.
- The list was later expanded to several dozens.
- Full CSS Color list

Color	Nombre	Código RGB	Color	Nombre	Código RGB
	blue	#0000FF		navy	#000080
	fuchsia	#FF00FF		purple	#800080
	red	#FF0000		maroon	#800000
	yellow	#FFFF00		olive	#808000
	lime	#00FF00		green	#008000
	aqua	#00FFFF		teal	#008080
	gray	#808080		black	#000000
	white	#FFFFFF		silver	#C0C0C0

Colors

Decimal RGB/RGBA encoding:

- To use decimal color values, `rgb()` function must be used.
  - `rgba()` to also include alpha.

```
rgb(redValue, greenValue, blueValue)
rgba(redValue, greenValue, blueValue, alphaValue)
```

- Color values, between [0, 255] or [0%, 100%]
- Alpha value, between 0.0 (transparent) and 1.0 (opaque).

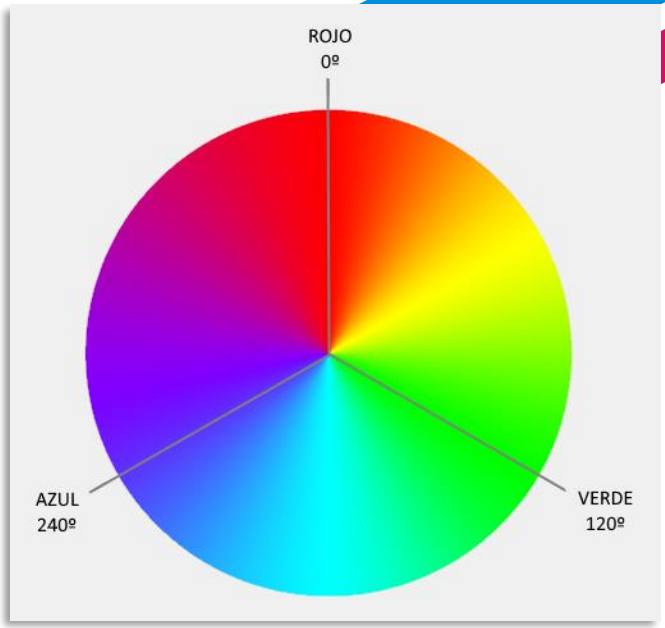
Colors

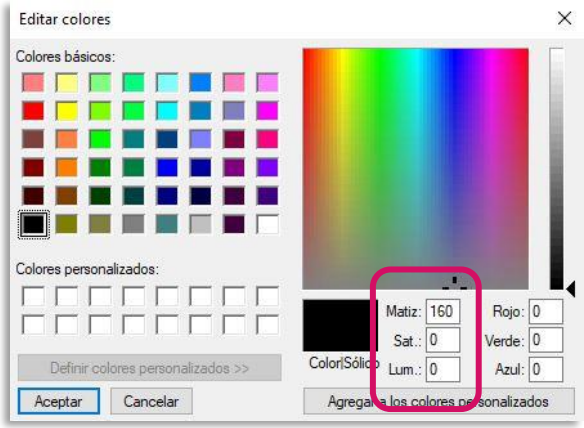
HSL/HSLA encoding:

- Alternative form to express colors. Two functions serve this purpose:

```
hsl(hueAngle, satPercent, lightPercent)
hsla(hueAngle, satPercent, lightPercent, alphaValue)
```

  - Hue: angle in the wheel
    - 0 = rojo; 120 = green; 240 = blue
  - Saturation: % of purity or vividness
    - 0% = grey; 100% = pure color
  - Lightness: % of clarity
    - 0% = black; 100% = white
  - Alpha: transparency / opacity
    - Values between [0.0, 1.0]





HSL Calculator

Colors

```
<BODY>
<ul>
  <li id=love1>I love oranges</li>
  <li id=love2>I love carrots</li>
  <li id=love3>I love pumpkins</li>
  <li id=love4>I love persimmons</li>
</ul>
</BODY>
```

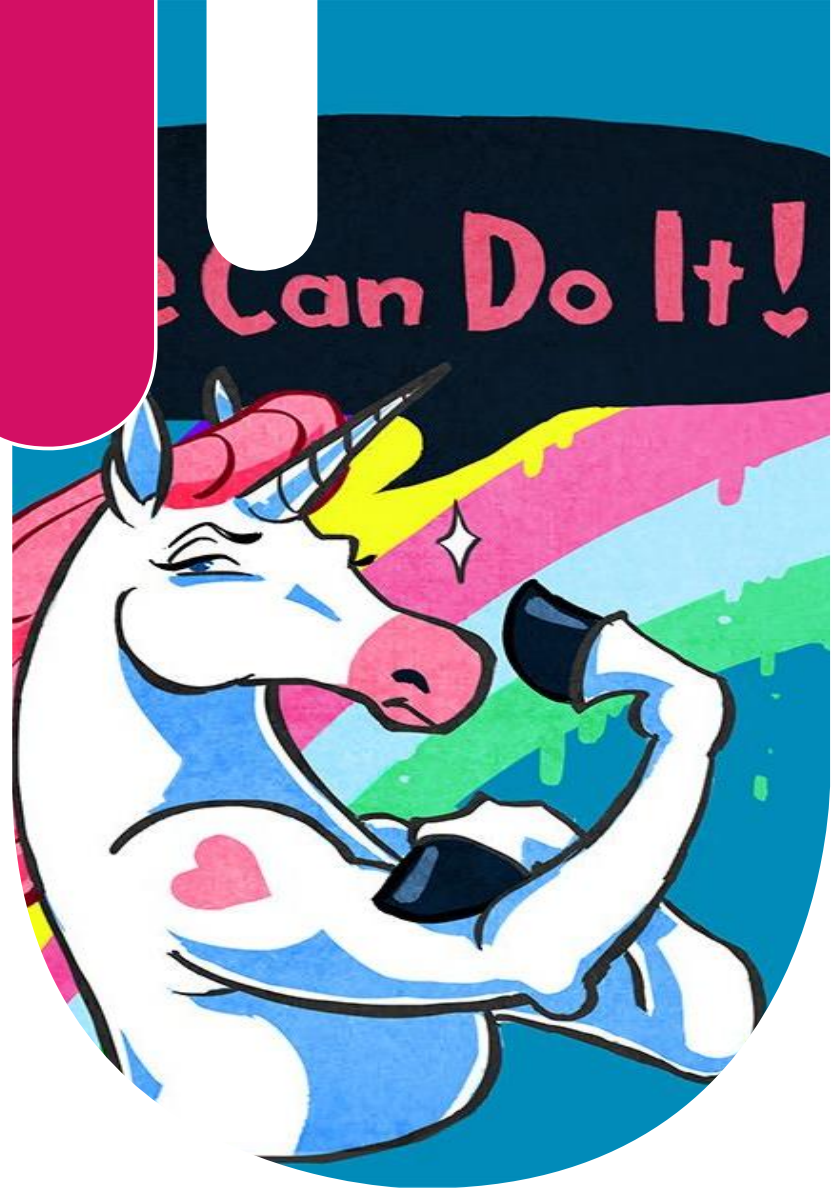
HTML code

- I love oranges
- I love carrots
- I love pumpkins
- I love persimmons

```
#love1 {
  background: orange;
}
#love2 {
  background: #ffa500;
}
#love3 {
  background: rgb(255, 165, 0);
}
#love4 {
  background: hsl(39, 100%, 50%);
}
```

CSS code

Color picker



Colors

PRACTICAL EXERCISE 6.2

7. LISTS

UNIT 6 HTML + CSS, PART I

Lists

There are three types of lists in HTML:

- Unordered
- Ordered
- Definition

Lists

UNORDERED

```
<BODY>
<h3><span>Unordered</span> lists</h3>
<ul>
  <li>The <span>order</span> of its items is <span>not important</span>.</li>
  <li><span>&lt;ul&gt;</span> and <span>&lt;/ul&gt;</span> delimit the list.</li>
  <li><span>&lt;li&gt;</span> and <span>&lt;/li&gt;</span> delimit each list item.</li>
  <li>Can be nested</li>
  <ul>
    <li>Even within ordered lists</li>
    <li>And vice versa</li>
  </ul>
</ul>
</BODY>
```

HTML code

Unordered lists

- The order of its items is not important.
- and delimit the list.
- and delimit each list item.
- Can be nested
  - Even within ordered lists
  - And vice versa

Lists

UNORDERED

Browsers display this list indented and bulleted by default.

- Can be modified using the `type` attribute
  - For the whole list
  - For some items
- The values `type` can take are:
  - disc *(default)*
  - circle
  - square
- We will rather use CSS for this.

```
<BODY>
<ul type=square>
  <li>Joffrey</li>
  <li type=disc>Cersei</li>
  <li type=circle>The Tickler</li>
  <li type=circle>Polliver</li>
  <li>Ser Amory Lorch</li>
  <li>Walder Frey</li>
  <li>Meryn Trant</li>
  <li>Tywin Lannister</li>
  <li type=disc>The Red Woman</li>
  <li>Berric Dondarrion</li>
  <li>Thoros of Myr</li>
  <li>Ser Ilyn Payne</li>
  <li>The Mountain</li>
  <li>The Hound</li>
</ul>
</BODY>
```

HTML code

- Joffrey
- Cersei
  - The Tickler
  - Polliver
- Ser Amory Lorch
- Walder Frey
- Meryn Trant
- Tywin Lannister
- The Red Woman
- Berric Dondarrion
- Thoros of Myr
- Ser Ilyn Payne
- The Mountain
- The Hound

Lists

ORDERED

Ordered lists

- Items order is important, which are numbered.
- `<ol>` and `</ol>` delimit the list.
- `<li>` and `</li>` delimit each item.
- They can be nested...

Browsers display this list indented and using Arabic numerals by default.

- Can be modified using the `type` attribute
  - For the whole list
  - For some items
- The values `type` can take are:
  - `1`: Arabic numerals *(default)*
  - `a`: lowercase letters
  - `A`: uppercase letters
  - `i`: lowercase Roman numerals
  - `I`: uppercase Roman numerals
- The `start` attribute is used to set the start value.
- We will rather use CSS for this.

Lists

ORDERED

```
<h4>Unit content:</h4>
<ol>
  <li>Word</li>
  <li>Excel</li>
  <ol start=4>
    <li type=i>Formulas</li>
    <li type=a>Charts</li>
  </ol>
  <li>Access</li>
  <ol type=A>
    <li>Tables</li>
    <li>Forms</li>
    <li>Reports</li>
  </ol>
</ol>
```

HTML code

Unit content:

1. Word
2. Excel
   iv. Formulas
   e. Charts
3. Access
   A. Tables
   B. Forms
   C. Reports

Lists

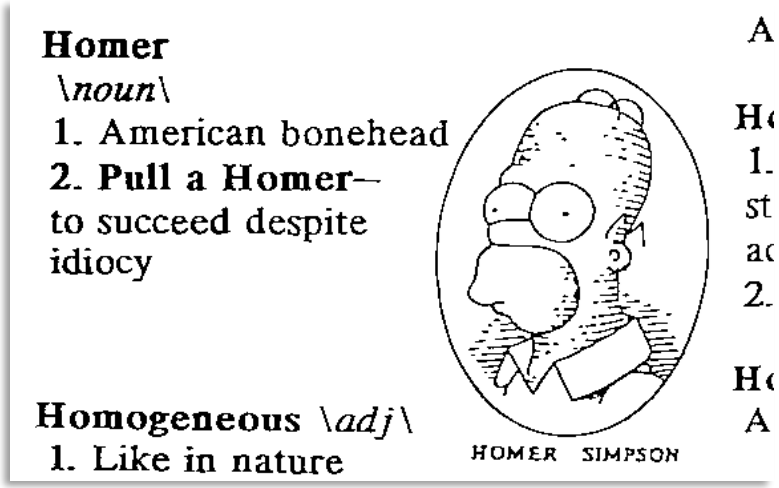
DEFINITION

Definition lists:

- Sets of terms and definitions.
  - Like a dictionary

- `<dl>` and `</dl>` delimit the list.
- `<dt>` and `</dt>` delimit each term.
- `<dd>` and `</dd>` delimit each definition.

- Browsers display this list with just the definitions indented.



Lists

DEFINITION

```
<dl>
  <dt>Coffee</dt>
  <dd>A beverage made by infusing the beans of
the coffee plant in hot water.</dd>
  <dt>Tea</dt>
  <dd>A drink made by infusing the dried leaves
or buds of the tea plant in hot water.</dd>
</dl>
```

HTML code

Coffee
: A beverage made by infusing the beans of the coffee plant in hot water.

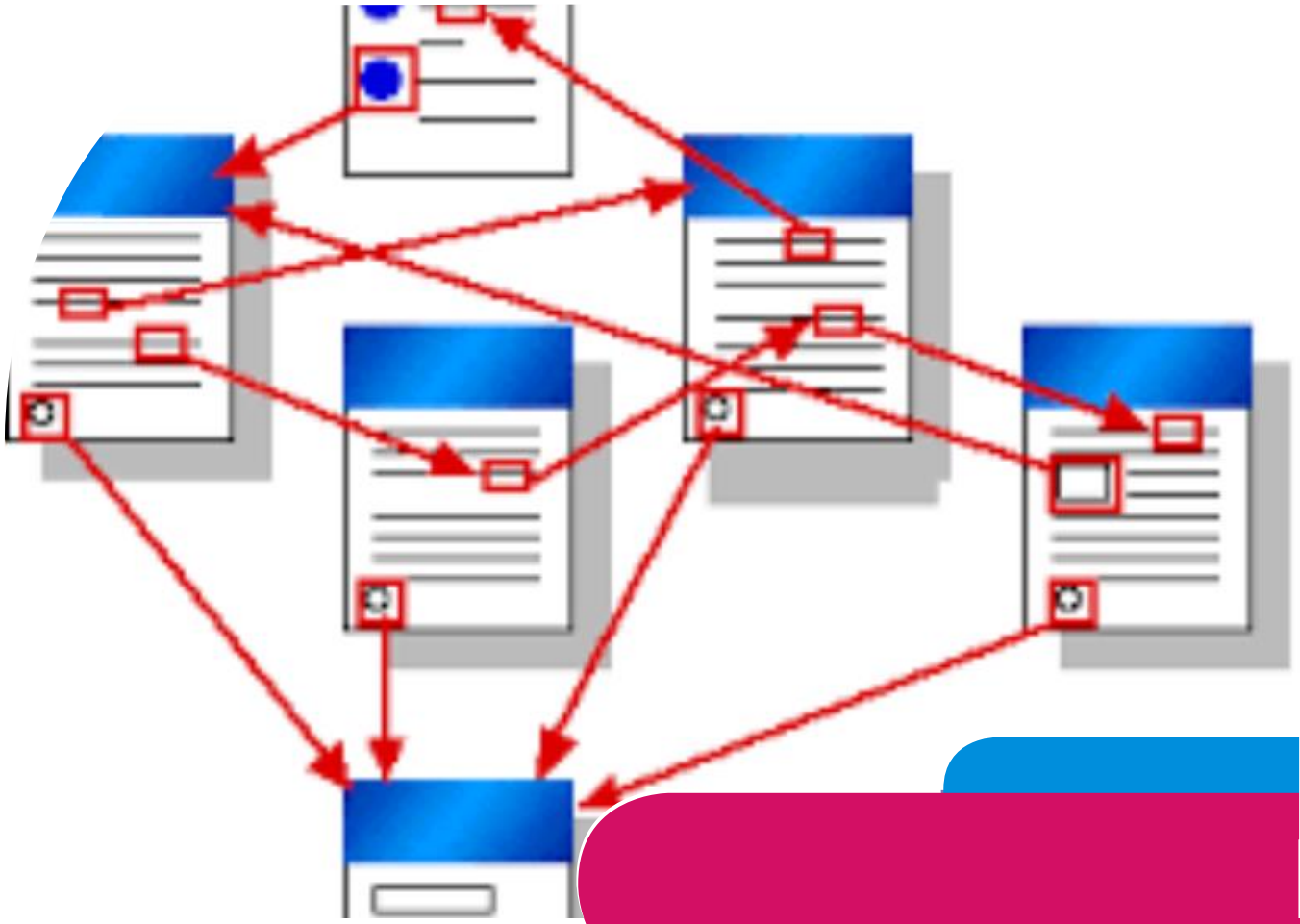
Tea
: A drink made by infusing the dried leaves or buds of the tea plant in hot water.

8. LINKS AND REDIRECTION

UNIT 6 HTML + CSS, PART I

Links

- *"Hypertext":* the information can be followed in a non-sequential manner through the links that the various elements don.
- HTML documents connect their content with other pages using references in the form of links.
- Links allow you to navigate through one site or can take you to other sites.



Links

Uniform Resource Identifier: the path to follow to reach a resource

- To create a link, the `<a>` element is needed:

```
<a href="target_URI">CONTENT</a>
```

- `<a>` is an inline element
  - The `CONTENT` can be text, an image, a video clip...
- The value the `href` attribute takes is the destination.
  - It could be a folder, a page or any file; local or remote.
    - If folder, how browsers behave depends on the server configuration (open some file, allow or deny accessing and listing).
- Include `target="_blank"` in the opening tag to open the link in a new tab.

Links

- URL *(Uniform Resource Locator)*: "Internet address", path to a remote resource (outside the local computer).

- As with local paths to resources, URLs can be:
  - Absolute: the name of the server and the full path to the resource are included.
  - Relative: the name of the server and, occasionally, the full path to the resource are omitted.

```
<body>
  <a href="https://portal.edu.gva.es/iesserpis/">Link to folder: server
configured to open some HTML file</a>
  <br>
  <a href="about.html" target=_blank>Link to file; relative URL; new tab</a>
</body>
```

HTML code

Links

URLs do not directly accept reserved characters *(blank space, line break, braces, spellings not contained in the English alphabet...)*.

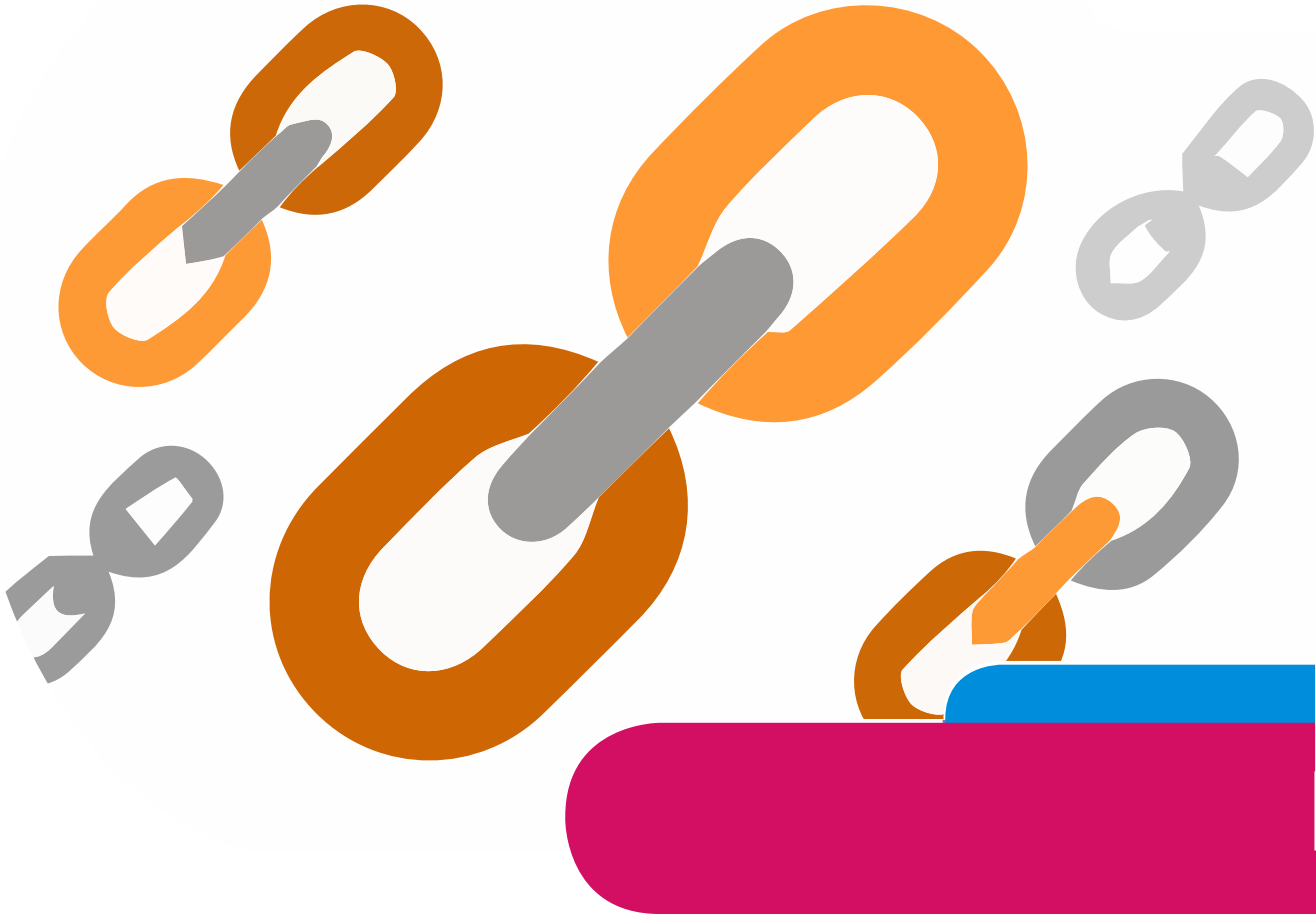
- The names of your documents and resources should never use these characters to avoid the problem.
- If there is no other choice, they can be escaped as %XX
  - XX is the hexadecimal ASCII code of the troublesome character.
  - Example: %20 is the blank space.

```
http://www.ejemplo.com/ruta/pagina inicio.html
http://www.ejemplo.com/ruta/pagina%20inicio.html
```

Links

Classification according to destination:

- Internal links
- Local links
- Remote links
- Mail links
- File links



Links

Internal links

- Take the user to a bookmark on the same page.
  - Useful in large pages.
- Bookmarks (AKA *anchors*) must be placed first.
  - `id` attribute of any element can be used for this.

```
<h3 id="ch3">Chapter 3: Some Text Here</h3>
```

- Then, link to the bookmark prefixing with #:

```
<a href="#ch3">Go to Chapter 3</a>
```

- `href="#"` takes the user to the start of the page
  - Commonly used in early stages of the development

Links

Local links

- Take the user to another page on the same site
- The link reference is the file name of the destination page.
  - With its path, if needed.
- Relative paths to the origin document should be used, whenever possible.

```
<a href="page2.html">Next page</a>
```

Links

Remote links

- Take the user to a page on another site (usually, another server).
- You need its URL.
  - It usually starts with `http://` or `https://`
  - Other uncommon protocols exist: `ftp://`, `news://`

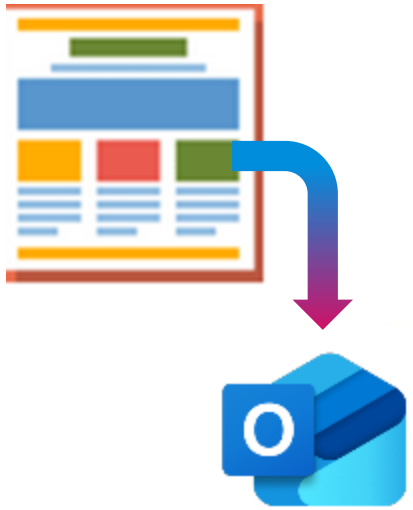
```
<a href="https://www.iesserpis.org">IES Serpis</a>
```

Links

Mail links

- If the browser is properly configured, open the default email application to compose a message to the specified recipient.

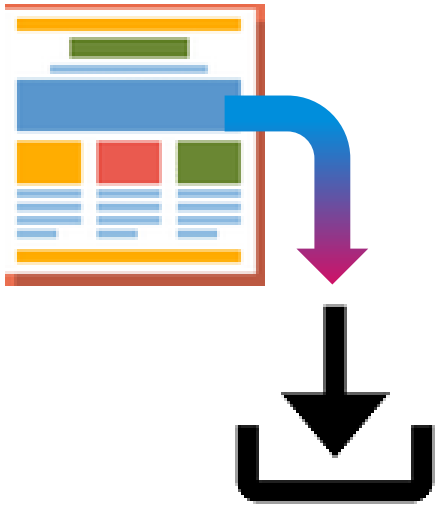
```
<a href="mailto:user@example.com">Contact us</a>
```



Links

File links

- Usually download a file.
  - It depends on the file type and how the browser behaves with each type.
- Browsers usually ask the user whether to open or save the file.
- The reference syntax is like those of HTML documents (local and remote).

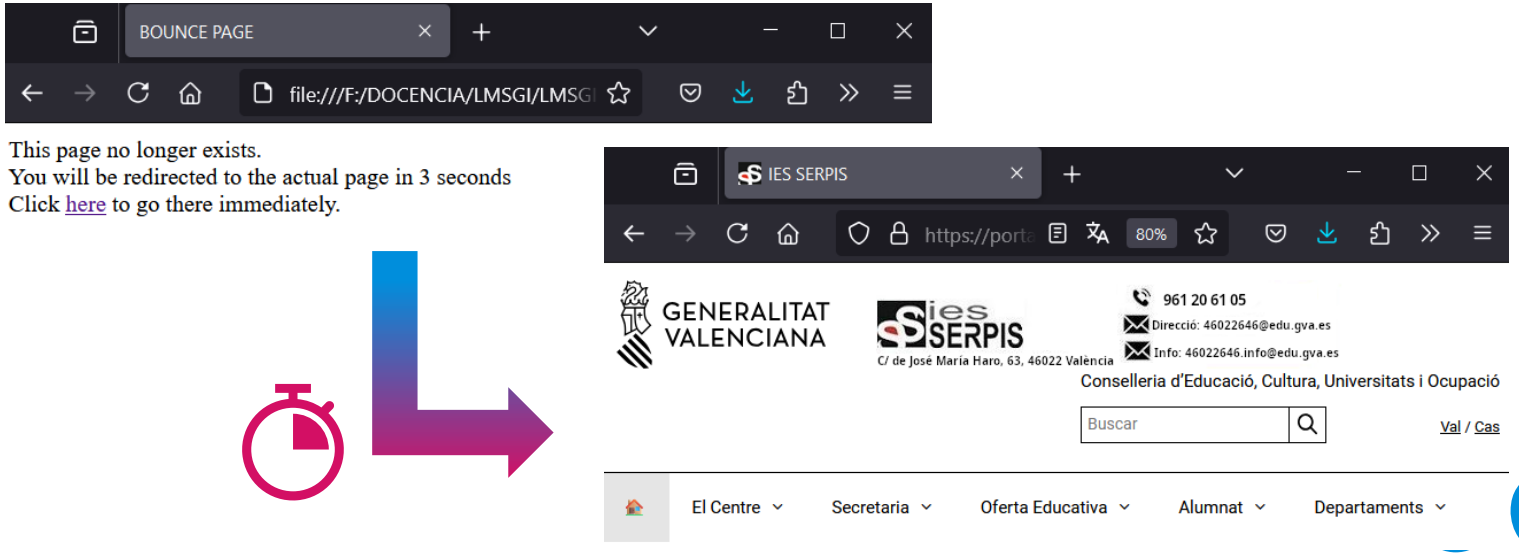


Redirection (forwarding)

- To make the page automatically redirect to another one on load, include a <meta> tag like this:

```
<meta http-equiv="refresh" content="3; https://www.iesserpis.org">
```

- `content` attribute:
  - time, in seconds
  - semicolon (`;`)
  - target URL



9. IMAGES

UNIT 6 HTML + CSS, PART I

Images

Two main types of graphics:

Raster or bitmap

- Can display images without defined shapes.
- Bitmap means that the image is considered a grid where each cell houses one color.
- The coordinates of each cell and its assigned color are stored.
- Lose quality when zoomed in.
- Heavier.
- Various compression algorithms.

Many formats

pixel
(picture element)

Vector

- Can only display defined shapes: lines, arcs, polygons
- Do not lose quality when zoomed in.
- Lighter.

Images

VECTOR

SVG

Scalable Vector Graphics

- Displays elemental shapes using tags
  - Like a markup language
- SVG files can be edited in two ways:
  - Graphically, using special editors (such as Inkscape)
  - Editing its code with a plain text editor
- (Usually) very light files.

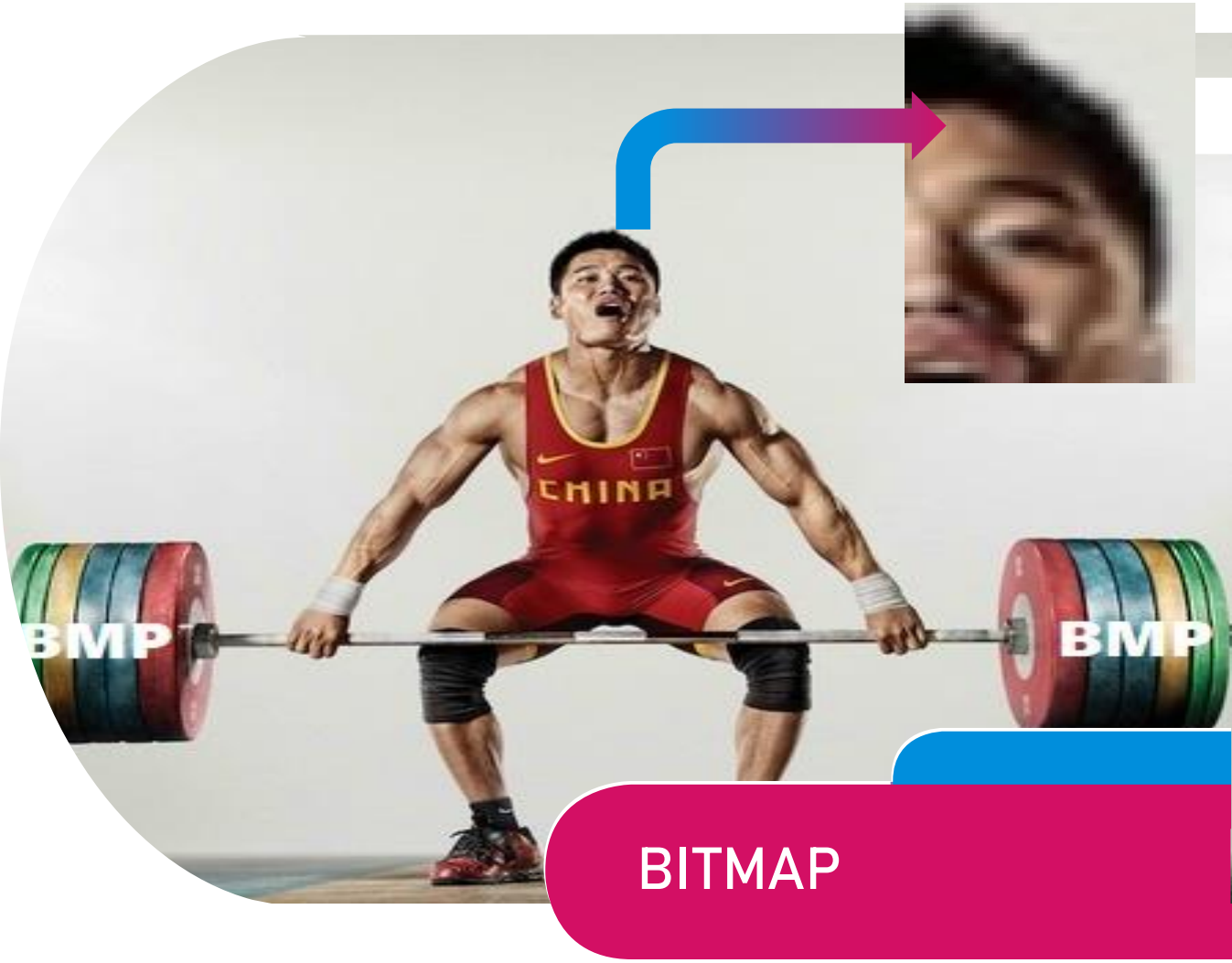
```
<path fill="#7D3A16" opacity="1.000000" stroke="none"
    d="
M173.468658,740.000000
    C172.904938,722.510132 173.755432,705.064697 175.714966,687.349609
    C181.116943,696.247559 186.608414,705.283325 191.305618,714.714722
    C195.412659,722.961182 201.291840,730.288086 203.996979,739.623413
    C193.979111,740.000000 183.958206,740.000000 173.468658,740.000000
z"/>
<path fill="#42868F" opacity="1.000000" stroke="none"
    d="
M1001.000000,720.531372
    C999.537598,719.469971 1000.021851,717.544556 1000.021729,715.764
    C1000.003601,481.862976 1000.005127,247.961472 1000.025818,14.0599
    C1000.025940,12.582335 1000.304749,11.104711 1000.726685,9.313538
    C1001.000000,246.020889 1001.000000,483.041779 1001.000000,720.531372
z"/>
<path fill="#1A1009" opacity="1.000000" stroke="none"
    d="
M741.468384,740.000000
```

Images

BMP

Bitmap

- No compression
- Very heavy
- Not recommended for web pages



Images

GIF

Graphics Interchange Format

- Up to 256 colors (8 bits per pixel).
  - Unsuitable for photographs.
- Light
- Lossless compression
- Can house animations and transparencies.

The original data can be perfectly restored from the compressed data without losing information

```
<!DOCTYPE html>
  <html>
    <body>
      <p>KEEP</p>
      <p>CALM</p>
      <p>AND</p>
      <p>CODE</p>
      <p>ON</p>
    </body>
  </html>
```

BITMAP

Images

BITMAP

JPEG

Joint Photographic Experts Group

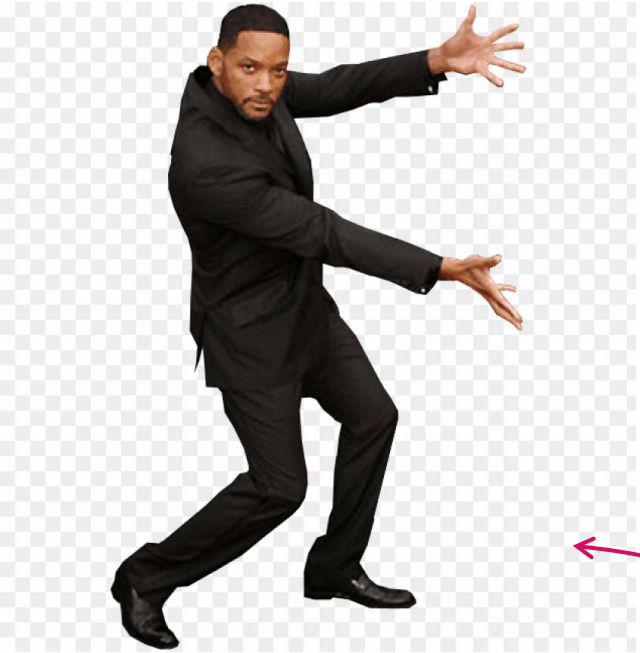
- Most photographs on the Internet.
- When creating the files, compression can be adjusted.

Weight / Quality ratio

- Supports interlacing
- Lossy compression:
  - Original data cannot be fully restored
  - Loses quality with each edition

Encoding technique that allows partially received images to be displayed, degraded

Images



The grey and white mosaic means
that the background is transparent

PNG

Portable Network Graphics

- Lossless and best compression than GIF.
- *Alpha* (transparencies).
  - Allows "background removing".
- *Gamma* (adjustable bright).
- Interlaced.

BITMAP

Images

BITMAP

WebP

Web Picture

- Google's proposal to replace JPEG, GIF and PNG.
- Supports both lossy and lossless compression.
- Supports animation and alpha.

- Recent format.
- Worse quality encoding than that of JPEG.
  - Blurrier images.

Images

- Using images is critical for page loading speed.
- Sometimes it will be convenient to make touch-ups to the images.
  - Decrease size.
  - Decrease quality (colors and/or resolution).
- Use *thumbnails* that link to the original images.
- Reuse *assets*
  - Images are stored in cache in the device.
  - When used on other pages, they are not downloaded.
  - Buttons, icons, delimiter bars...

Images

SYNTAX

```
<img src="image_URI" alt="Alternative text">
```

- Void element (does not have a closing tag).
- The `alt` attribute is mandatory in HTML5.
  - Accessibility: required for screen readers.
- Optional attributes:
  - `title="popup text"`
    - Displayed next to the mouse pointer after one second leaving it idle over the image.
  - `width` and `height` to adjust size.
    - Bitmaps are deformed if not used properly, vector images auto-adjust.
    - We will use CSS for this.

Images

PRACTICAL EXERCISE 6.3

10. FAVICON

UNIT 6 HTML + CSS, PART I

つづく

THANKS!

Do you have any questions?

g.domingomartinez@edu.gva.es