Welcome to
CS1315 Intro to Media Computation

"Our greatest weakness lies in giving up.

The most certain way to succeed

is always to try just one more time."

– Thomas Edison

Day 31

What is needed to get started:

- Download the popular and powerful free text editor VS Code

  https://code.visualstudio.com/download

- Download from CANVAS Files- Class Notes- the folder

  day31_HTML

  - save it to your folder CS1315

Using VS Code:

1. Open VS Code.
2. Close the Welcome tab
3. Click on the Extensions icon in the side bar 
4. Install the Extension **Live server** to view live html pages in VS Code
   - https://youtu.be/ZfCi0Is9gLU?si=Qaf3OCUrKWQ96rAi
5. Install the Extension **vscode-pdf** to view PDF files in VS Code

Using VS Code to create a blank html file:

In VS Code, you can quickly create a sample HTML template:

1. Go to "File" > " Open folder ", select your downloaded folder **`day31_html`**
2. Go to "File" > "**New File**"
3. Go to "File" > "Save As..." , name the file **`index.html`**
4. On line 1 of the file type press the **<!>** key then “Enter”.
5. VS Code will create a new file with a basic HTML template, including the `<!DOCTYPE html>` declaration, `<html>`, `<head>`, and `<body>` tags.

Using VS Code to view your html file in your default browser:

1. Start Live Server: Right-click on the HTML file and select “Open with Live Server”. This will open your default web browser to a URL of your file. Copy the **Live Server URL.**

2. Use the Simple Browser Extension by pressing **`Ctrl + Shift + P`** to open the Command Palette, then type “**Simple Browser: Show**”.

3. Paste the **Live Server URL** in the dialog box, press Enter, and your default browser will open displaying your html file.

4. Set Up a Side-by-Side View: click **View -> Editor Layout -> Two-columns**

5. Right-click on the HTML file and select “**Open to the side**”

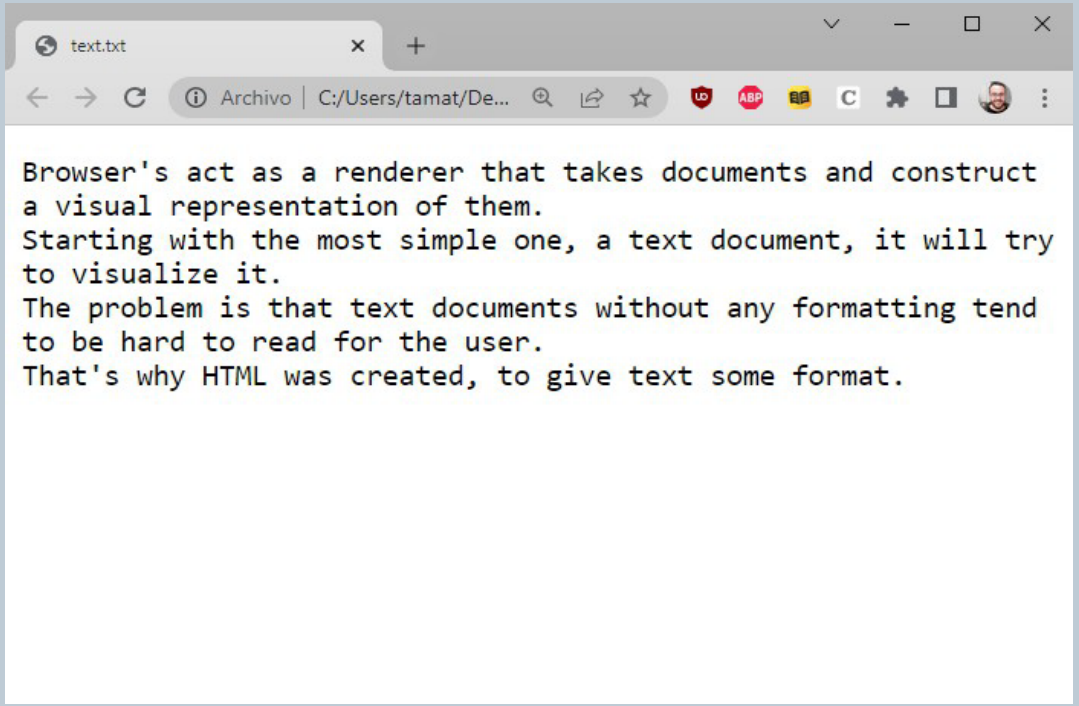
6. This setup allows you to see your code and the live preview side by side, making it easier to develop and test your web pages.

References

- W3 Schools HTML Tutorial:
  - a description of all HTML tags.
- codecademy: Learn HTML Code:
  - a list with information of the more common tags.

Browsers as a renderer

- Browser's act as a renderer that takes documents and construct a visual representation of them.
- Starting with the simplest kind, a text document, it will try to visualize it.
- You can try it: open any `.txt` file into your browser to visualize it.
- The problem is that text documents without any formatting tend to be hard to read for the user (and quite boring).
- That's why HTML was created, to give text some format.



HTML

HTML means Hyper Text Markup Language.

The HTML allow us to define the structure of a document or a website.

HTML is **NOT** a programming language, it's a markup language, which means its purpose is to give structure to the content of the website, not to define an algorithm.

It is a series of nested tags (it is a subset of XML) that contain all the website information (like texts, images and videos). Here is an example of tags:

```
<title>This is a title</title>
```

The HTML defines the page structure. A website can have several HTMLs to different pages.

```
<html>
<head>
</head>
    <body>
      <div>
      <p>Hi</p>
      </div>
    </body>
</html>
```

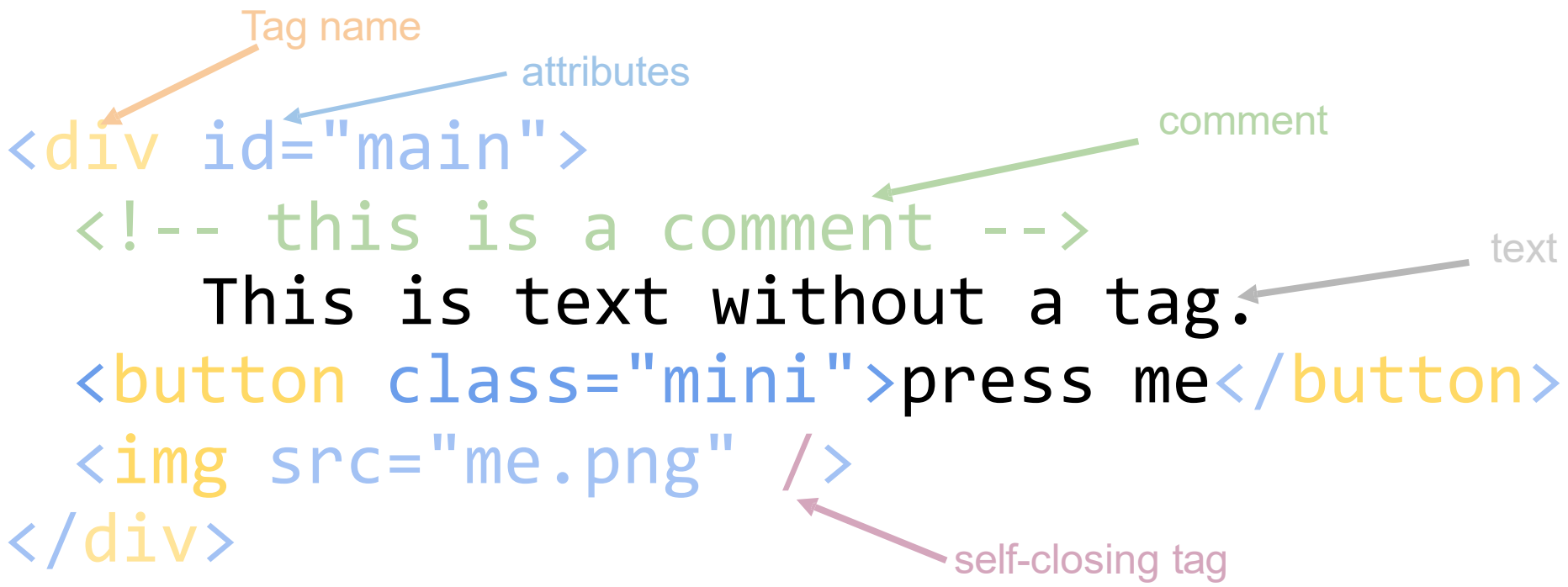
HTML: basic rules

- It uses XML syntax (tags with attributes, can contain other tags).

```
<tag_name attribute="value"> content </tag_name>
```

- It stores all the information that must be shown to the user.
- There are different HTML elements for different types of information and behaviour.
- The information is stored in a tree-like structure (nodes that contain nodes inside) called the DOM (Document Object Model).
- It gives the document some semantic structure (ie. this is a title, this is a section, this is a form) which is helpful for computers to understand websites content.
- It must not contain information related to how it should be displayed (that information belongs to the CSS), so no color information, font size, position, etc.

HTML: syntax example



HTML: main tags

Although there are lots of tags in the HTML specification, 99% of websites use a subset of HTML tags with less that 10 tags, the most important are:

- `<div></div>` : a block-level container for grouping content.
- `<img sre="url" alt="description" />` : Embeds an image (self-closing tag).
- `<a> </a>` : Creates a hyperlink to another URL.
- `<p> </p>` : Defines a paragraph of text.
- `<h1></h1>` : A top-level heading (with <h2>, <h3>, etc. for subheadings).
- `<input type="text" />` : A form element for user input (self-closing tag).
- `<style></style>` : Contains CSS rules for styling the document.
- `<script></script>` : Embeds JavaScript code for interactivity.
- `<span></span>` : An inline container for elements that doesn't do anything.

HTML: other interesting tags

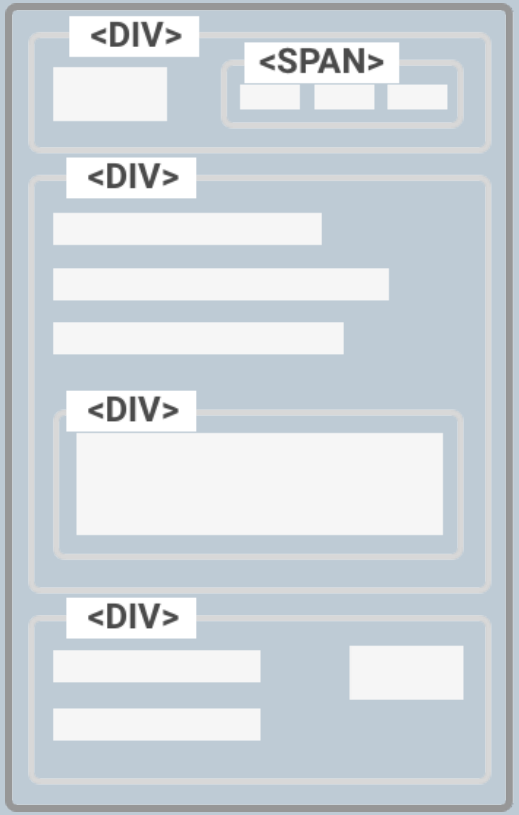
There are some tags that could be useful sometimes:

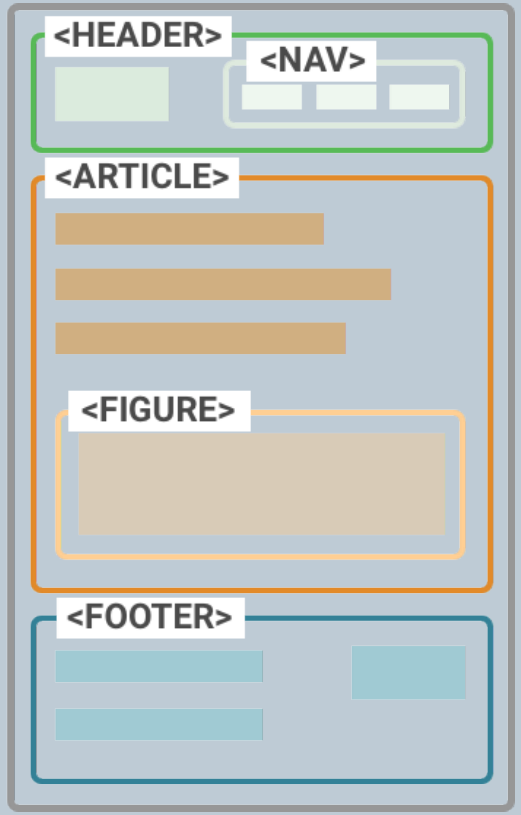
- <button> : to create a button
- <audio> : for playing audio
- <video> : to play video
- <canvas> : to draw graphics from javascript
- <iframe> : to put another website inside ours

HTML: wrapping the info

We use HTML tags to wrap different information on our site.

- The more structure the information has, the easier it will be to access it and present it.
- We can change the way the information is represented on the screen depending on the tags where it is contained, so we shouldn't be worried about using too many tags.
- HTML ignores additional spaces you place in the .html file





HTML: tagging correctly

Try to avoid doing this:

```
<div>
Title

Here is some content
Here is more content
</div>
```

Do this instead:

```
<div>
   <h1>Title</h1>
   <p>Here is content.</p>
   <p>Here is more content</p>
</div>
```

HTML good use

It is good to have all the information properly wrapped in tags that give it some semantics.

We also can extend the code semantics by adding extra attributes to the tags:

- id: is a unique identifier for this tag
- class: is a generic identifier for this tag

```
<div id="profile-picture" class="mini-image">...</div>
```