

# Introduction to web development

By Alex Robic

# Main objective of this course

- Make YOUR website about things YOU like
  - Don't make it too simple but also don't over complicate it
- Ex: I like Pokemon so my site is all about pokemon
- **The real objective is to give you enough basics for you to make a simple page, if you want to make it more complicated, I can give you references**
- How I like to structure my courses:
  - Tutorial part (1h-1h30)
  - Independent work (30min-1h) → during this time, you can consult me for help (debugging) while you work on your website

By the end of the week you should have a simple website that you can then edit **as you wish**

*Legal disclaimer: Alex does not condone nor condemn the content on your website. You have total freedom to put whatever you want but do keep in mind that your parents may want to look at it*

# Class rules

Important note: demonstrations are important, not everything is easily understandable in theory so sometimes, it's best to see it live and understand why it's doing such a thing

- Name format:
  - Instructor: Alex Robic
  - Student:
    - Alex R
    - Alex
- Chat use: unlimited but DO NOT:
  - Share contact info (phone #, email, social media etc)
  - You can share your website links with others (after all you *own* it)**
- **FEEL FREE TO ASK AS MANY QUESTIONS AS YOU WANT AND/OR REQUEST A DEMONSTRATION (this kind of stuff is not always easy to understand and some of you are more visual learners like me)**

# Good things to have

- A web browser (I think everyone has one) → software that lets you explore the web
- Visual studio code → free software that lets you edit (with colors 😊)
- A github account (that's how you'll be hosting your website)

# By the end of the course...

You should be able to:

1. Place elements on a webpage using HTML (hypertext markup language)
2. Style these elements (placement, alignment, color, border etc)
3. Add interaction with these elements using javascript (+ use a programming language as opposed to a scripting one)

# Objective 1: build a simple web page

- Build a simple web page with HTML
  - Header types → (<h> and <p>)
  - Title
  - Page structure
  - Div / span
  - Links
  - Images
  - Videos
  - Iframe
  - Br / hr
  - Table / tr / td
  - Ol / ul
  - Favicon

## Objective 2: give it style with CSS

- Give it style with CSS
  - Color
  - Font
  - Placement
  - Highlight
  - Decoration
  - Float
  - Border / background
  - Style markers VS external file

## Objective 3: make it interactive with javascript

- Click, double click, drag and mouse over
- Alert and confirm
- Variables, loops etc
- Animations
- Keyboard events
- Mouse events

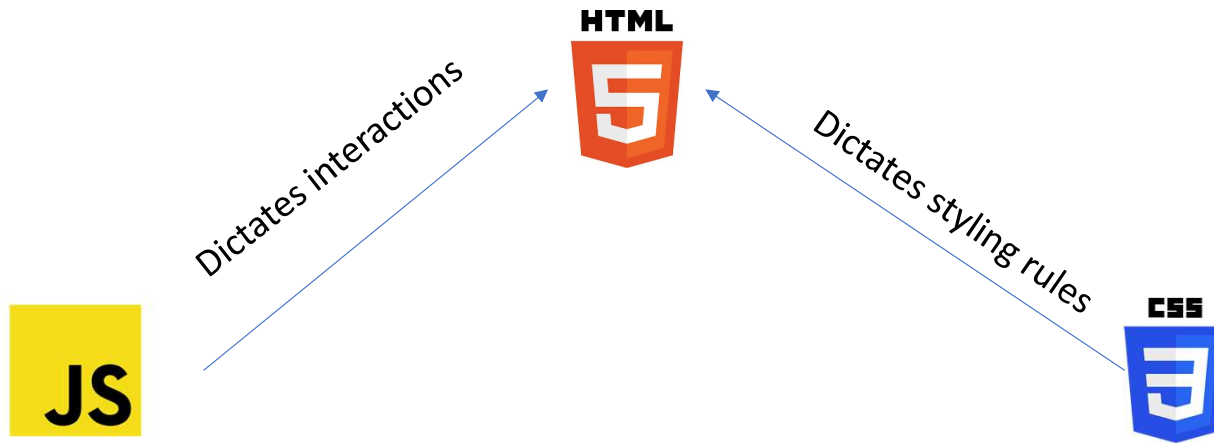


MONDAY



# Objective 1: a simple page with a few things on it

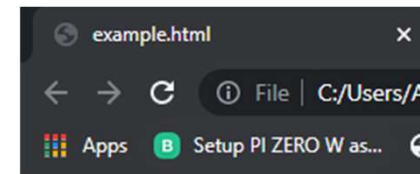
- File type: .html → HTML = HyperText Markup Language
  - Other file extensions exist (.php, .xml) but those will not be covered in this course



# Objective 1: a simple page with a few things on it

- Ex:

```
example.html x script.js style.css
example.html > html > p#paragraph
1 <html>
2   <link rel="stylesheet" href="style.css">
3   <script type="text/javascript" src="script.js"></script>
4   <p id="paragraph" onclick="disappear(this)"> This is a paragraph </p>
5 </html>
```

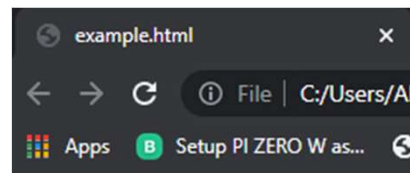


This is a paragraph

- HTML: `<p>This is a paragraph </p>` →

- CSS: `p { color:red; }` →

- Javascript:

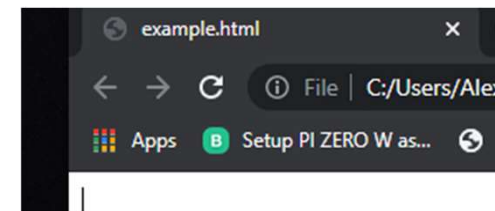


This is a paragraph

```
example.html x script.js style.css
script.js > ...
1 function disappear (x) {
2   x.style.display = "none";
3 }
4
5 var p = document.getElementById("paragraph");
```

AR1

The code



After clicking on the paragraph

## Slide 11

---

**AR1**

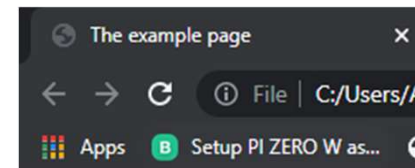
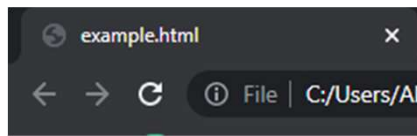
Alexandre Robic, 6/20/2021

# Objective 1: a simple page with a few things on it

- Today: HTML
- **What is written first, gets loaded first**
- Rule: in most cases, if you open a tag, you have to close it
  - Ex: `<p>` opens a paragraph and `</p>` closes it
  - Exceptions: `<img>`
- See it as a big sandwich but you have to have 2 of each thing (hope that makes sense)
- Your document starts with a `<html>` tag so which should it end with?
- Page structured like a human (header, body, footer)

# Objective 1: a simple page with a few things on it

- Title: by default uses the file name (here, example.html)

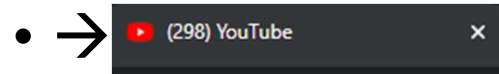


```
1 <html>
2   <header>
3     <link rel="stylesheet" href="style.css">
4     <script type="text/javascript" src="script.js"></script>
5     <title> The example page </title>
6   </header>
7   <body>
8
9   </body>
10  <footer>
11
12  </footer>
13
14 </html>
```

<title> Your page title here </title>  
→ Put this in your header

# Objective 1: a simple page with a few things on it

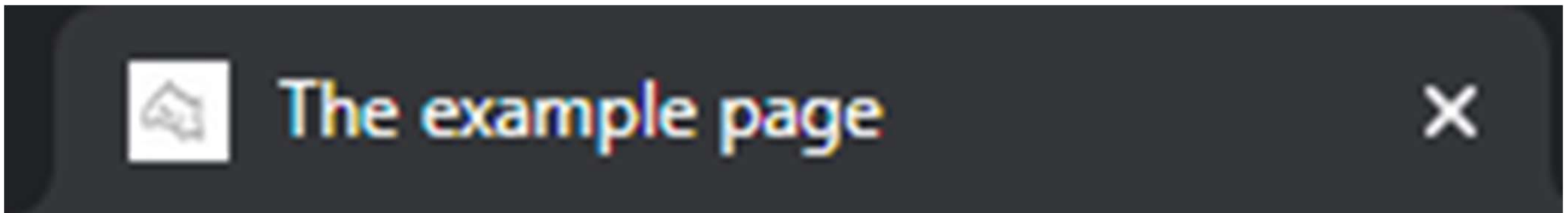
- Favicon: the little icon next to the page title



- You can set your own!

- `<link rel="shortcut icon" href="bongo.jpg" type="image/jpg">`

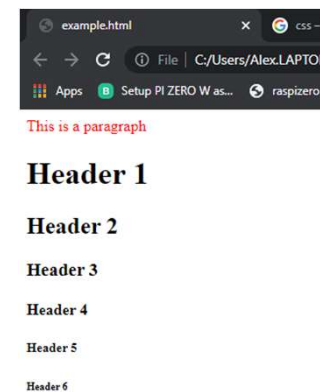
Opens a link tag   Define the type of link   Give it the link to your image   In this case, it's a good thing to mention the type of image



# Objective 1: a simple page with a few things on it

- Header types:
- `<h>` tags are used for titles and `<p>` for paragraphs
- `<h>` tags can be accompanied by a number to give a certain order of importance
  - Ex: `<h1>` is your most important (page title perhaps) and `<h6>` is your least important (end credits?)

```
1 <html>
2   <link rel="stylesheet" href="style.css">
3   <script type="text/javascript" src="script.js"></script>
4   <p> This is a paragraph </p>
5   <h1>Header 1</h1>
6   <h2>Header 2</h2>
7   <h3>Header 3</h3>
8   <h4>Header 4</h4>
9   <h5>Header 5</h5>
10  <h6>Header 6</h6>
11 </html>
```





# Objective 1: a simple page with a few things on it

- Images: One of the only tags that does not have a closing tag
- `<img src='path/to/image' alt='alt text' width='chosen width' height='chosen height'>`
- Height and width can be written in px or % (although other units exist, these are the 2 you will most frequently find, pt is another story)
  - % is % of window
  - Px is fixed (when you change the size of the window, you DO NOT change the size of the image)
- Demonstration with pikachu image

# Objective 1: a simple page with a few things on it

- `<br>` and `<hr>`
- `<br>` used to break line (like pressing enter when you're typing)
  - Use case: You have content you want on the next line or you want to space content out
- `<hr>` used to draw a horizontal line on page (useful to separate content and can be customized)

# Objective 1: a simple page with a few things on it

```
<video width="50%" controls muted>  
  <source src="Microwave Sugar Cookie _ Em's Kitchen.mp4" type="video/mp4">  
  Your web browser does not support videos  
</video>
```

- Video tag (closed at the end)
- Source (you can put several and if the first one does not work or is not supported, it will switch to the second one)
- The text at the bottom is displayed if none of the sources work (like if for some reason your browser does not support video playback)
- Controls, controls muted, autoplay muted (chrome does not allow autoplay)
- It's good to set a width (and/or height) to avoid flickering during page load (and avoid too large of a video)
- Poster → thumbnail

# Objective 1: a simple page with a few things on it

- Youtube: the exception + iframes
- Iframe: a special tag to see a window INSIDE another window



- Each video has a specific ID (here, zHBqUgPEa\_0), rather than use iframe on whole link (and get video suggestions, ads on side etc), use embedded youtube (youtube.com/embed/{video id})
- Ex:
- [https://www.youtube.com/embed/zHBqUgPEa\\_0](https://www.youtube.com/embed/zHBqUgPEa_0) as opposed to
- [https://www.youtube.com/watch?v=zHBqUgPEa\\_0](https://www.youtube.com/watch?v=zHBqUgPEa_0)

# Objective 1: a simple page with a few things on it

- Talking about getting stuff from other websites...
- Links or the secret of the internet's success
- `<a href="{the page you want to link to}">`

`<!-- Whatever you put in here, becomes what is clicked on to access your site -->`

`</a>`

Ex:

```
<a href="https://www.youtube.com/watch?v=hBP-NzOadL0"></a>
```

This places the bongo cat image on screen and if you click it, takes you to a youtube video

# Objective 1: a simple page with a few things on it

- Tables (starts with a <table> tag) (see it as a wall of boxes, each table is a wall of shelves, each tr is a shelf and each td is a box)
- <tr> → opens a new row
- <td> → opens a new data entry (or column)
- (stuff in tables doesn't have to be numbers btw)
- Ex:

<table>

<tr> <td> A </td> <td> L </td> </tr>

<tr> <td> F </td> <td> H </td> </tr>

</table>

A	L
F	H

Fill in the table

# Objective 1: a simple page with a few things on it

- Lists:
- Unordered (ul) → by default, displayed with disks
- Ordered (ol) → by default, displayed with numbers (Arabic)
- List item (li) → an item in the list (by default, goes to next line at the end)

<ol>

<li>Tomatoes</li>

<li>Bread</li>

<li>Cheese</li>

</ol>

What will it look like?

# Objective 1: a simple page with a few things on it

- Div / span → useful for styling (tomorrow's topic)
- Div: a certain division of the page
- Span: a certain portion of text
- We'll look at this more tomorrow when we do styling



# Objective 1: a simple page with a few things on it

- A few extra tags to start to add a bit of style:
- `<u>Text</u>` → Text
- `<i>Text</i>` → *Text*
- `<b>Text</b>` → **Text**
- `<center> </center>` → puts content in center of screen
- `1<sup>st</sup>` → 1<sup>st</sup>
- `C<sub>12</sub>H<sub>22</sub>O<sub>11</sub>` → C<sub>12</sub>H<sub>22</sub>O<sub>11</sub>

(For those of you wondering, it's the chemical formula for sugar, like your everyday table sugar)

# Objective 1: a simple page with a few things on it

- Get to work!!!!!!
- Your task: put everything (or almost everything) you want on your website on the page, we'll worry about style tomorrow (+ build the other pages to your website)
- Feel free to consult me (just give me a few minutes to get coffee while you work)
- You have total design freedom (just do note that the staff or your parents may want to look at it)
- When you're done if you don't already have a github account, create one, if you do, create a repository called *username.github.io* and upload your page(s) **with the main one being index.html**