

# CHAPTER 2 : Front-end programming language for web



*Dr. KIHAL Marouane*

University of M'sila

Faculty of Mathematics and  
Computer Science

Departement of computer  
Science

chapter 2

# Table of contents

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>I - CHAPTER 2 : Front-end programming language for web</b>          | <b>3</b>  |
| 1. Chapter 2 specific objectives .....                                 | 3         |
| 2. Introduction .....                                                  | 3         |
| 3. General information: static pages and Web applications .....        | 4         |
| 3.1. Web Applications .....                                            | 4         |
| 3.2. Front-End and Static Web Pages .....                              | 4         |
| 4. Tag languages: definition and history .....                         | 6         |
| 4.1. Defining tag languages .....                                      | 6         |
| 4.2. Historique .....                                                  | 6         |
| 5. HTML .....                                                          | 7         |
| 5.1. HTML Definition .....                                             | 7         |
| 5.2. HTML execution context .....                                      | 7         |
| 5.3. Basic HTML .....                                                  | 9         |
| 5.4. Style sheets (CSS) .....                                          | 13        |
| 6. Conclusion .....                                                    | 15        |
| 7. Exercice : Do you understand chapter2 !! lets answer the quiz. .... | 15        |
| <b>Abbreviation</b>                                                    | <b>17</b> |
| <b>References</b>                                                      | <b>18</b> |
| <b>Bibliography</b>                                                    | <b>19</b> |
| <b>Web bibliography</b>                                                | <b>20</b> |

# I CHAPTER 2 : Front-end programming language for web

## 1. Chapter 2 specific objectives

At the end of Chapter 2, the student will be able to:

- **Define** the fundamental concepts of HTML.
- **Identify** the main HTML tags and CSS properties.
- **Describe** the structure of a static web page.
- **Explain** the respective roles of HTML (structure) and CSS (style) in a web application.
- **Use** HTML to build a basic web page. Apply CSS rules to style a web page.
- **build** a complete web page with structure and style according to given requirements.

## 2. Introduction

This chapter introduces the fundamentals of front-end web development, focusing on how static web pages are built using markup and styling languages. We begin by explaining what a web application is, and then distinguish static pages from dynamic ones. A detailed look at front-end technologies such as HTML and CSS follows, showing how they structure and style web content. Through examples and code, to understand the key building blocks of static websites.

## 3. General information: static pages and Web applications

### 3.1. Web Applications

#### Definition

A web application is a computer program that can be accessed through an internet browser and combines front-end (also known as client) and back-end (also known as server) technology. These apps, which range from content consultation to real-time data management, enable users to interact. Typically, they use server-side languages like PHP\*, Python, Ruby, or Node.js to manage user interactions with a database or carry out particular treatments, as well as HTML, CSS\*, and JavaScript for user interfaces.

#### Fundamental

Primary characteristics of web applications :

- **Accessibility:**  
Web applications can be accessed from any device with an internet browser without the need for a specific installation.
- **Interactivity:**  
Thanks to contemporary frameworks and languages like JavaScript, they enable rich interactions.
- **Client-server architecture:**  
Web applications rely on communication between the client (navigator) and the server, where the server processes requests and sends the relevant responses.
- **Simplified daily management:**  
Unlike native programs, daily management occurs directly on the server without requiring user participation.

#### Note

Web applications can be divided into two distinct categories according to how they work:

1. **Static web pages:**  
fixed content, generated once and identical for all users.
2. **Dynamic web pages:**  
Content generated in real time as a function of user interaction or data from a database.

#### Reminder

In this chapter, we focus on static web pages.

### 3.2. Front-End and Static Web Pages

#### Definition

A web static page is a kind of page that is created and displayed entirely by the front-end. This indicates that everything of its content, including titles, text, graphics, and styles, is written in Amont, usually using front-end technologies like HTML, CSS, and occasionally JavaScript. This content is static and does not alter dynamically

based on user behavior or particular interactions. As a result, a static page is created once and sent to each visitor's browser without any server changes. It doesn't need data bases or dynamic processing. The entire user experience is based on what the front-end displays.

### 💡 *Fundamental*

Primary characteristics of static page:

- **Same content:** The content is predetermined in the source code and remains the same for every visitor.
- **Technologies front-end :** Mostly uses HTML for structure, CSS for presentation, and occasionally JavaScript for a few fundamental interactions.
- **Facilitated creation:** Because static pages don't require sophisticated settings or server routines, they are easy to develop.
- **Charge speed:** Because the data are sent straight from the server to the browser, they appear quite quickly.

### 🔗 *Example:concrete example*

A showcase site. This often consists of a home page presenting the business, an 'About' section, a contact page with fixed contact details, a gallery or a list of products with no interactive features. In all these cases, the content does not change automatically: only manual changes to the HTML or CSS files can update it.

### *Advantages vs disadvantages*

- **Advantages:** Easy to understand and quick to put online, low costs (no database or dynamic server) and excellent charging times.
- **Disadvantages:** No user-specific content is available, every daily update necessitates a manual code intervention, and limited interactions (systems such as user connections or dynamic searches cannot be integrated).

## 4. Tag languages: definition and history

### 4.1. Defining tag languages

#### Definition

Tag languages (or Markup languages) are computer languages designed to structure and organize digital documents. They use tags special elements enclosed in angle brackets “< >” to define the logical structure of content, such as headings, paragraphs, images, or links.

#### Fundamental

The main goal is to make content clear and well-structured for both humans and machines (like web browsers and search engines). Unlike styling languages (e.g., CSS), markup languages focus on semantic structure rather than visual appearance. The most widely used markup language is HTML<sup>\*</sup> (HyperText Markup Language), which is essential for building and organizing web pages. These languages help create standardized, accessible, and versatile digital content.

### 4.2. Historique

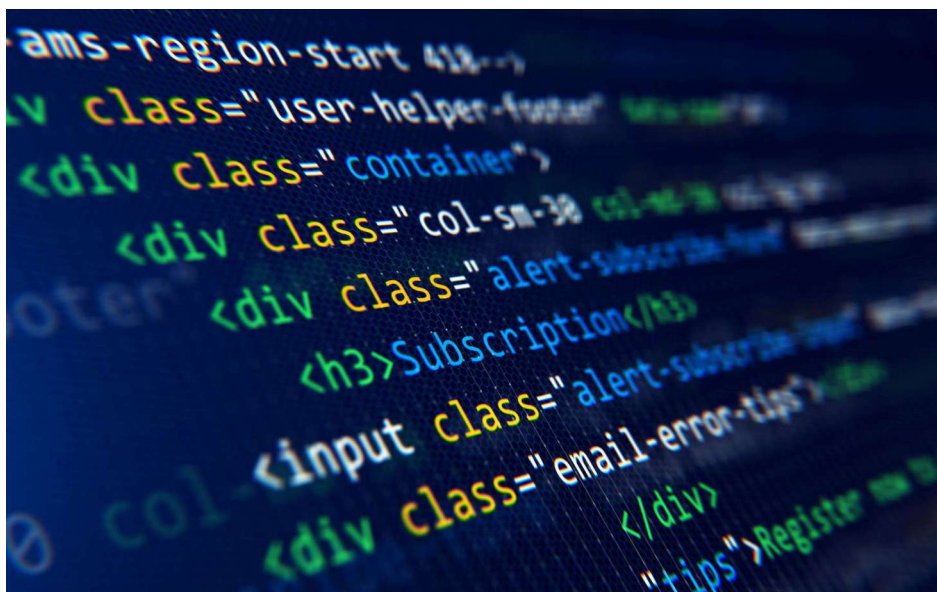
The evolution of markup languages is marked by the move from SGML (Standard Generalized Markup Language) , a complex meta-language created in the 1980s to define document structuring languages, to HTML (HyperText Markup Language) , a simplified version adapted specifically for the web. HTML, developed by Tim Berners-Lee in the early 1990s, was inspired by SGML to enable the creation of hypertext documents accessible via a browser. Over time, HTML evolved through several versions (HTML 2.0, HTML 4.01, XHTML) to incorporate more advanced functionalities, until the arrival of HTML5 , which introduces semantic elements, better support for media, and APIs for dynamic interactions. This evolution reflects the transition from a generic and rigid standard to a flexible and powerful language, adapted to the modern needs of the web.

## 5. HTML

### 5.1. HTML Definition

#### Definition

HTML (HyperText Markup Language) is used to represent and organize the content of web documents. By defining the semantic elements and their hierarchy, it contributes to a web page and enables the incorporation of media, hypertext links, and other resources through user-interpretable balises.



HTML

### 5.2. HTML execution context

The context in which HTML is executed depends on how web browsers interpret it; *they analyze HTML code* to create and display sites in a *visually* appealing manner.

When a user accesses a web page, the browser retrieves the HTML file from a server, parses its contents (including its structure and balises), and then *builds a document object model (DOM)* that serves as the foundation for graphically restoring the page. In order to provide a comprehensive and functional online experience, HTML frequently works with languages like CSS for style and JavaScript.



-HTML



## 5.3. Basic HTML

### 5.3.1. Framework of an HTML document

The backbone of an HTML document is based on a minimal structure made up of three essential elements:

#### 1) `<html>` :

The root tag which encompasses all the content of the document.

- `<head>` The content tag: `<head>` contains information that is not directly visible on the web page, but which is essential for the browser or search engines. Common elements include:
- `<title>`: Defines the title of the page, displayed in the browser tab.

```
1 <title>Page title</title>
```

`<meta>`: Provides metadata such as character encoding, page description or display settings.

```
1 <meta charset="UTF-8"> <meta name="description" content="A description of the page"> <meta name
  ="viewport" content="width=device-width, initial-scale=1.0">
```

`<link>` : Allows external resources, such as CSS style sheets, to be linked.

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

`<script>` : Includes or references JavaScript scripts.

```
1 <script src="script.js"></script>
```

`<style>` : Used to define CSS styles directly in the HTML document.

```
1 <style> body { background-color: lightblue; } </style>
```

#### 2) `<body>` :

The `<body>` contains all the visible content of the web page. Here are the essential elements commonly used:

- Headings (`<h1>` to `<h6>`) : Define the levels of titles and subtitles.

```
1 <h1>Main title</h1>
2 <h2>Subtitle</h2>
```

- Paragraphs (`<p>`): To structure text into paragraphs.

```
1 <p>This is a paragraph of text.</p>
```

- Links (`<a>`): Create hyperlinks to other pages or sections.

```
1 <a href="https://example.com">Visit Example.com</a>
```

- Images (`<img>`): Insert images into the page.

```
1 
```

- Lists (`<ul>`, `<ol>`, `<li>`): To create unordered (`<ul>`) or ordered (`<ol>`) lists.

```
1 <ul>
2 <li>Element 1</li>
3 <li>Element 2</li>
4 </ul>
5
6 <ol>
```

```
7 <li>Ordred Element 1</li>
8 <li>Ordred Element 2</li>
9 </ol>
```

- Divisions (<div>): Used to group together sections of content.

```
1 <div>
2 <p>Content grouped in a div.</p>
3 </div>
```

### 3) Semantic sections (<header>, <footer>, <article>, etc.):

Used to structure content in a logical and accessible way.

```
1 <header> <h1>Page header</h1> </header>
2 <footer> <p>Page footer</p> </footer>
3
```

## 5.3.2. Exercice : Exercise

Reorder the code HTML :

</h1>

</html>

<h1>

<html>

My Website

### 5.3.3. Tables and Forms

#### a) Tables

##### Definition

---

Tables are used to structure data in rows and columns

##### Fundamental

---

They are useful for displaying tabular information such as results, times or comparisons using tags such as:

- `<table>`: This tag defines the start and end of a table.
- `<tr>` (table row): Defines a row in the table.
- `<td>` (table data): This defines a data cell in a row. Each `<td>` represents a column in the row.

##### Syntax

---

Syntax of a table in HTML:

```
1 <table>
2 <tr>
3 <td>Data 1</td>
4 <td>Data 2</td>
5 <td>Data 3</td>
6 </tr> <tr>
7 <td>Data 4</td>
8 <td>Data 5</td>
9 <td>Data 6</td>
10 </tr>
11 </table>
```

#### b) Forms

##### Definition

---

forms are used to collect data entered by the user

##### Fundamental

---

Forms use the tags :

- `<form>`: Defines the start of a form.

```
1 <form action="/submit">...</form>
```

- `<input>`: Defines an input field (text, password, email, checkbox, etc.)

```
1 <input type="text" name="username" placeholder="Enter your username">
```

- `<textarea>`: Defines a multi-line text input field (for larger text like messages).

```
1 <textarea name="message" rows="6" cols="60" placeholder="Enter your message"></textarea>
2
```

- `<label>`: Defines a text label.

```
1 <label> Username </label>
```

- **<select>**: Defines a dropdown list, and **<option>**: defines an option inside a **<select>**.

```
1 <select name="country">
2 <option>Algeria</option>
3 <option>Canada</option>
4 <option>Italy</option>
5 </select>
```

**<button>**: Defines a clickable button (can submit or reset a form).

```
1 <button>Submit</button>
```

## Syntax

---

Syntax of a form in HTML:

```
1 <form action="/submit">
2   <label for="username">Username:</label>
3   <input type="text" id="username" name="username" placeholder="Enter your username">
4   <br><br>
5   <label for="message">Message:</label>
6   <textarea id="message" name="message" rows="6" cols="60" placeholder="Enter your message"></
  textarea>
7   <br><br>
8   <label for="country">Select your country:</label>
9   <select id="country" name="country">
10    <option>Algeria</option>
11    <option>Canada</option>
12    <option>Italy</option>
13  </select>
14  <br><br>
15  <button type="submit">Submit</button>
16 </form>
```

## 5.4. Style sheets (CSS)

### 5.4.1. CSS Definition

#### Definition

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation and formatting of documents structured in HTML or XML<sup>\*</sup>. It separates content (defined in HTML) from its visual appearance, by applying style rules (colours, typography, margins, positioning, etc.) via selectors that target DOM elements. CSS is interpreted by browsers and is part of an inheritance and cascade logic, where styles are applied according to their specificity and the order in which they are defined.

### 5.4.2. CSS Colors

#### Fundamental

The CSS properties linked to colours allow you to define the visual appearance of HTML elements in terms of tints, transparency and gradients. They are essential for customising the design of a web page. Here are the main properties and their uses:

- **Color:** Defines the text colour of an element.

```
1 color: red; /* Red text */
2 color: #336699; /* Blue text with hexadecimal code */
3 color: rgb(0, 128, 0); /* Green text in RGB */
4 color: rgba(0, 0, 0, 0.5); /* Semi-transparent black text (50%) */
5
```

- **background-color:** Defines the background colour of an element.

```
1 background-color: yellow; /* Yellow background */
2 background-color: #f0f0f0; /* Light grey background */
3 background-color: rgba(255, 0, 0, 0.3); /* Semi-transparent red background */
```

- **border-color:** Defines the border colour of an element.

```
1 border: 2px solid blue; /* Blue border */
2 border-color: green; /* Changes the border colour only */
3
```

**font-style:** Applies a specific style to the font, such as italic or oblique.

Possible values: normal, italic, oblique.

### 5.4.3. Fonts

#### Fundamental

- **font-family:** Defines the font used for the text.

```
1 font-family: 'Arial', 'Helvetica', sans-serif;
```

- **font-size:** Specifies the size of the text.

```
1 font-size: 16px; /* Fixed size in pixels */
2 font-size: 1.2rem; /* Size relative to the root */
```

- **font-weight:** Defines the thickness (fatness) of the font.  
common values: normal, bold, or numerical values such as 100, 400 (normal), 700 (bold).

```
1 font-weight: bold; /* Bold text */
2 font-weight: 500; /* Intermediate thickness */
```

- **text-align:** Horizontal text alignment.  
possible values: left, right, centre, justify.

```
1 text-align: centre; /* Centred text */
2 text-align: justify; /* Justified text */
3
```

- **text-decoration:** Add decorations to the text.  
possible values: underline, overline, line-through, none.

```
1 text-decoration: underline; /* Underline the text */
2 text-decoration: none; /* Remove all decoration */.
3
```

#### 5.4.4. Exercice : HTML or CSS ?

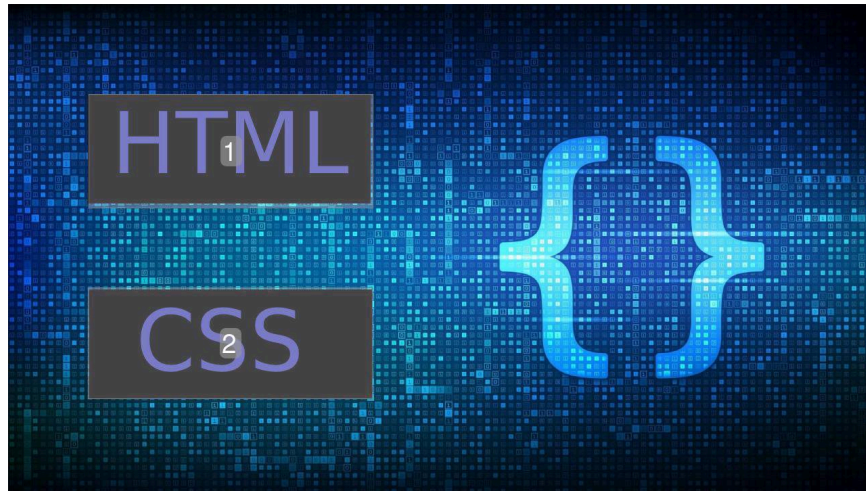
Which language is used to structure the content of a webpage?



graphique quiz

#### 5.4.5. Exercice : HTML or CSS ?

Which language is used to style the layout, colors, and fonts of a webpage?



*graphique quiz*

#### 5.4.6. Exercice : HTML or CSS ?

Which languages are needed to create a basic styled webpage with both content and visual design?



*graphique quiz*

## 6. Conclusion

In summary, front-end development relies heavily on HTML for structure and CSS for styling to deliver static content efficiently. These technologies make it possible to build clear, fast, and lightweight web pages, even without dynamic server-side processing. Students should now understand how to write simple static pages and style them using basic CSS properties. The historical progression from SGML to HTML. This knowledge forms a necessary first step toward mastering more interactive and dynamic front-end development.

## 7. Exercice : Do you understand chapter2 !! lets answer the quiz.

Exercice

---

Which property is used in CSS to change the text color of an element?

☐ font-color

Exercise : Do you understand chapter2 !! lets answer the quiz.

- ☐ text-style
- ☐ color
- ☐ text-color

Exercise

---

Is a personal portfolio website static or dynamic ?

Exercise

---

Which part handles database operations and server logic?

- ☐ Front-end
- ☐ Backend

Exercise

---

Which part of web development involves designing user interfaces and layout?

- ☐ Front-end
- ☐ Backend

Exercise

---

Which one includes HTML, CSS?

- ☐ Front-end
- ☐ Backend

Exercise

---

Which one is responsible for making an API call?

- ☐ Front-end
- ☐ Backend



# Abbreviation

**CSS:** Cascading Style Sheets

**PHP:** PHP: Hypertext Preprocessor

# References

*ref5*

5. Dean, J. (2025). Web Programming with HTML, CSS, and JavaScript. Jones & Bartlett Learning.

*ref6*

6. Berardini, E., Caminata, A., & Ravagnani, A. (2024). Structure of CSS and CSS-T quantum codes. Designs, Codes and Cryptography, 92(10), 2801-2823.

# Bibliography

Raggett, D., Lam, J., Alexander, I., & Kmieć, M. (1998). Raggett on HTML 4. Addison-Wesley Longman Publishing Co., Inc..

DUCKETT, Jon et SCHLÜTER, Jens. Html & Css. Wiley, 2011.

Faulkner, S., Eicholz, A., Leithead, T., Danilo, A., & Moon, S. (2017). HTML 5.2. W3C. Retrieved January, 17, 2018.

Heurtel, O. (2014). PHP y MySQL. Domine el desarrollo de un sitio web dinámico e interactivo. Barcelona: ENI.

Lerdorf, R., & Tatroe, K. (2002). Programming Php. " O'Reilly Media, Inc.".

# Web bibliography

<http://simplilearn.com/what-is-web-1-0-web-2-0-and-web-3-0-with-their-difference-article>

<https://www.avast.com/fr-fr/c-web-3-0>

<https://www.w3schools.com/html/>

<https://www.w3schools.com/js/>

<https://www.w3schools.com/css/>