Cairo University
Faculty of Engineering
Computer Engineering Department

Software Engineering
Lab1 – Introduction to Web

By / Eng Mohamed Sayed
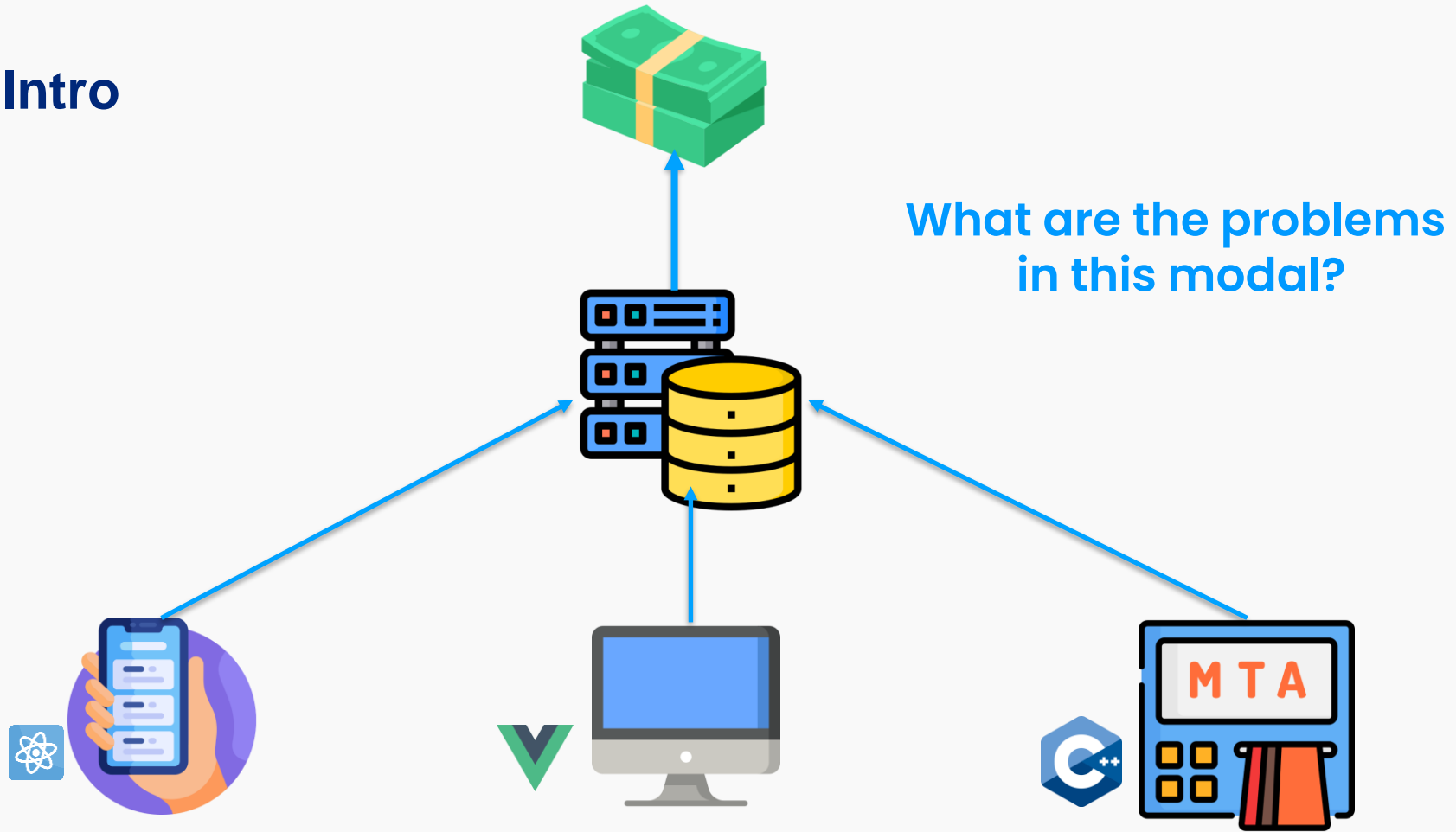
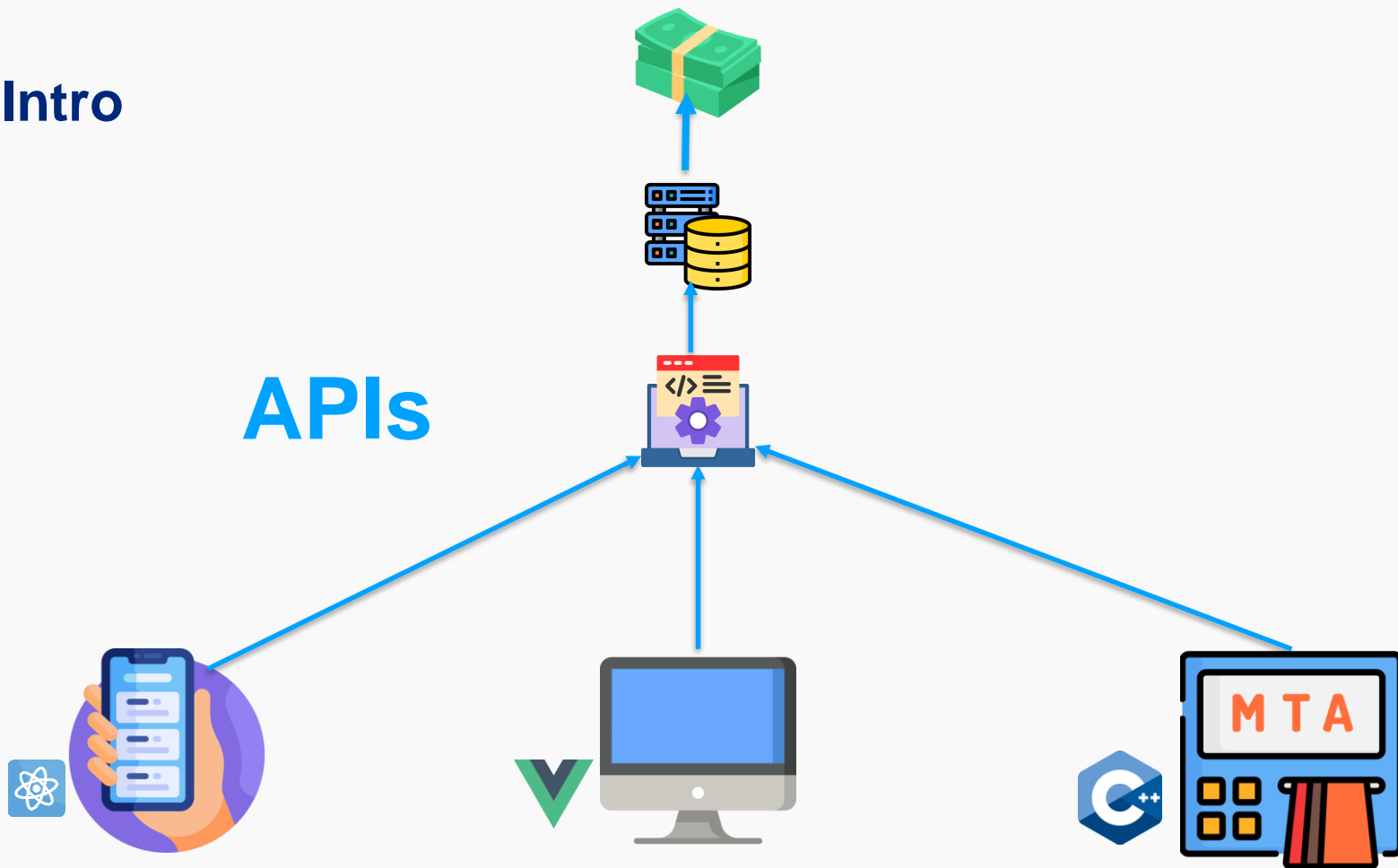# Agenda Items

- **Web Architecture**

  - **Web browser**
  - **Web service**
  - **APIs**

- **Frontend development**

  - **HTML**
  - **CSS**
  - **JS**

- **Backend development**

Intro

What are the problems in this modal?

Intro

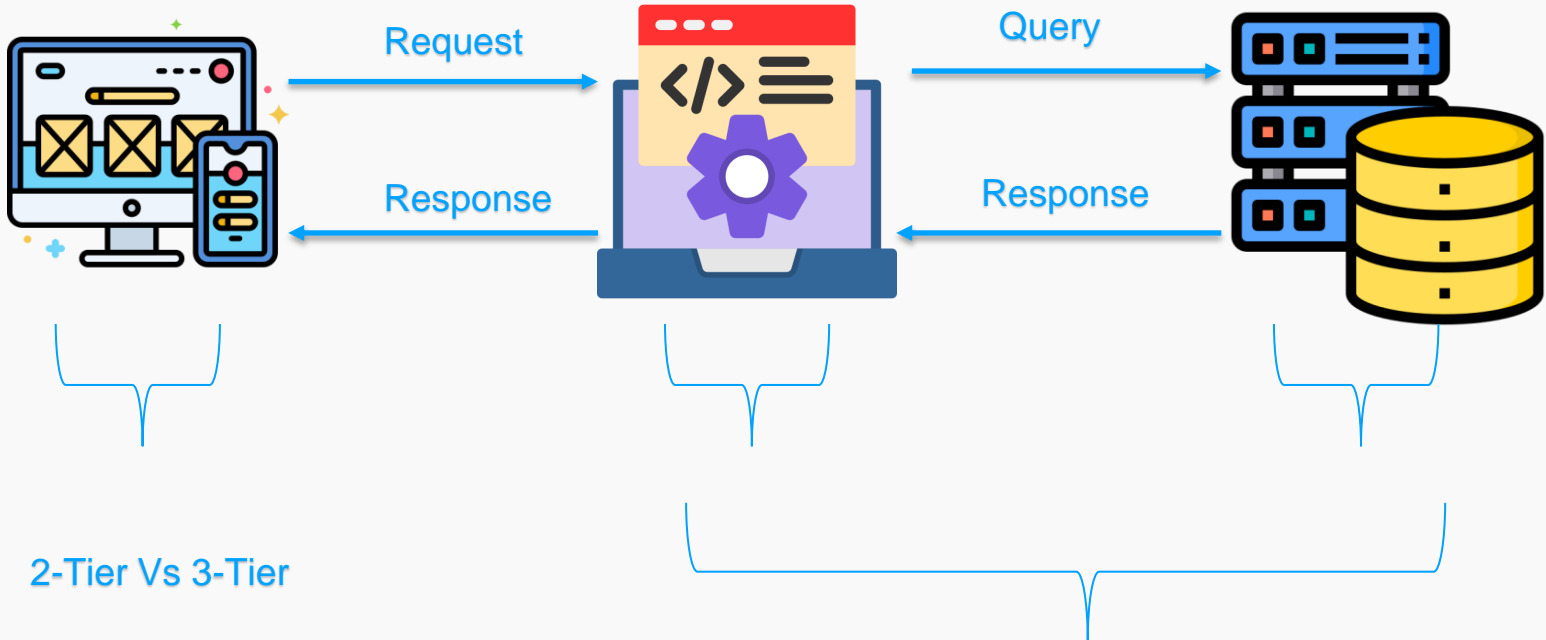APIs

Web Architecture

Request

Response

Query

Response

2-Tier Vs 3-Tier
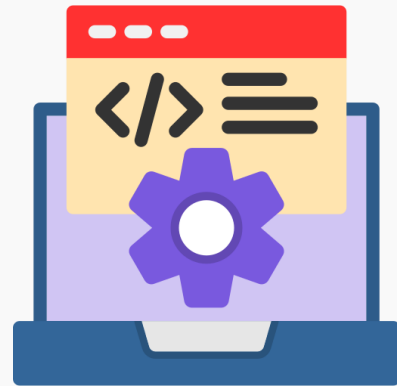
Web Architecture

What is web browser ?

- A client-side software application that connects its side to the server side.

- It is used for retrieving, presenting, and traversing.

- The major web browsers are Firefox ,Safari, Google Chrome, and Opera.

Web Architecture

What is web service ?

- Web services provide a standardized way for different software systems.

- Web services typically use standard web protocols like HTTP and data formats like JSON.

- There are different types of web services, including REST, and GraphQL.

Web Architecture - APIs
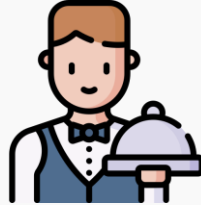
What is API?

Web Architecture - APIs

What is API?

Web Architecture - APIs
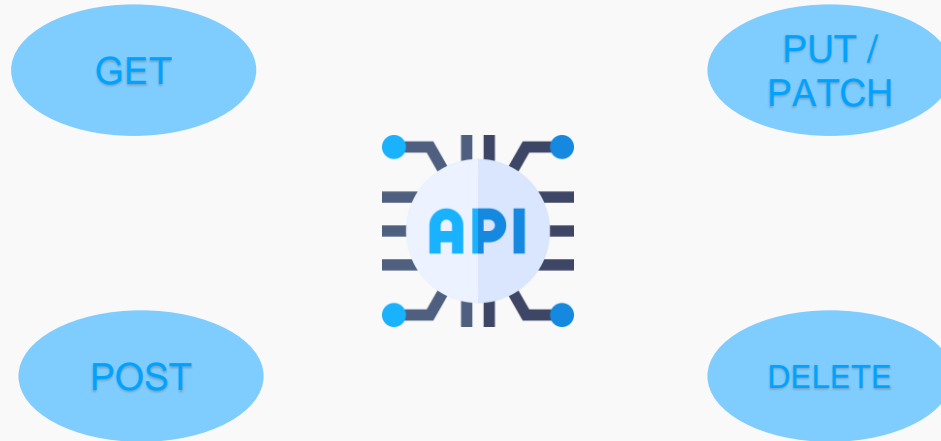
What is REST?

Web Architecture - APIs

What is REST?

Web Architecture - APIs

How Does it Work ?
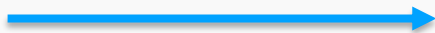

Get all burgers

GET   https://bufflo-apis.com/burgers-list
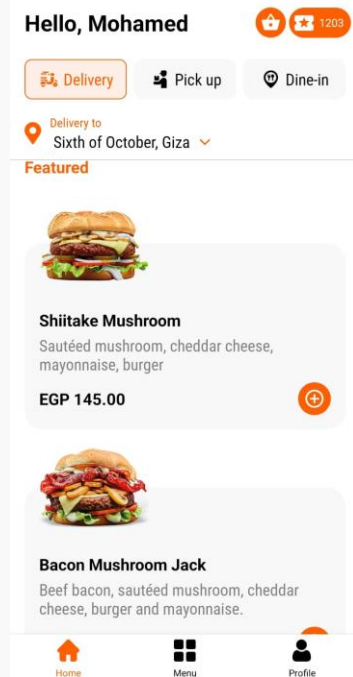
Web Architecture - APIs

How Does it Work ?

Add burger to cart

POST https://bufflo-apis.com/add-to-cart/id=burger_id

Frontend development

HTML (Hypertext Markup Language):
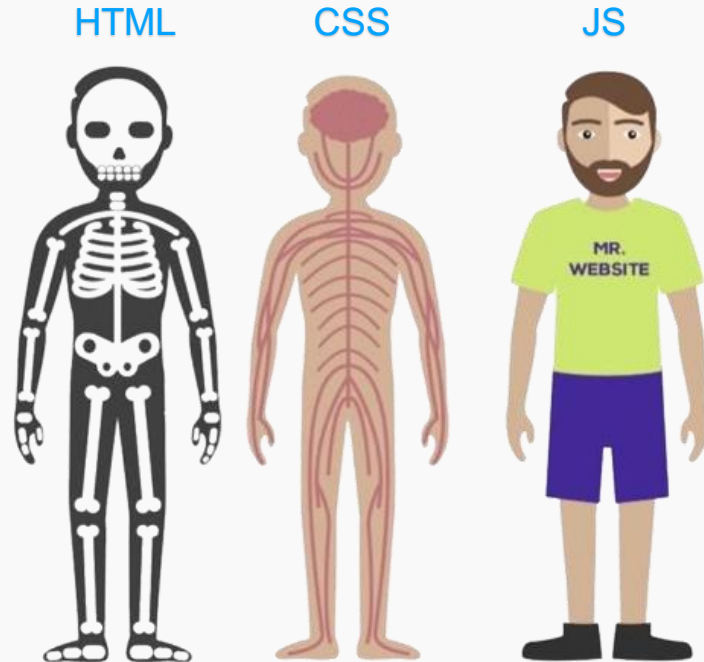
Markup Language: HTML is the standard markup
language used to create the structure and content
of web pages

CSS (Cascading Style Sheets):

Styling Language: CSS is used to style the presentation and
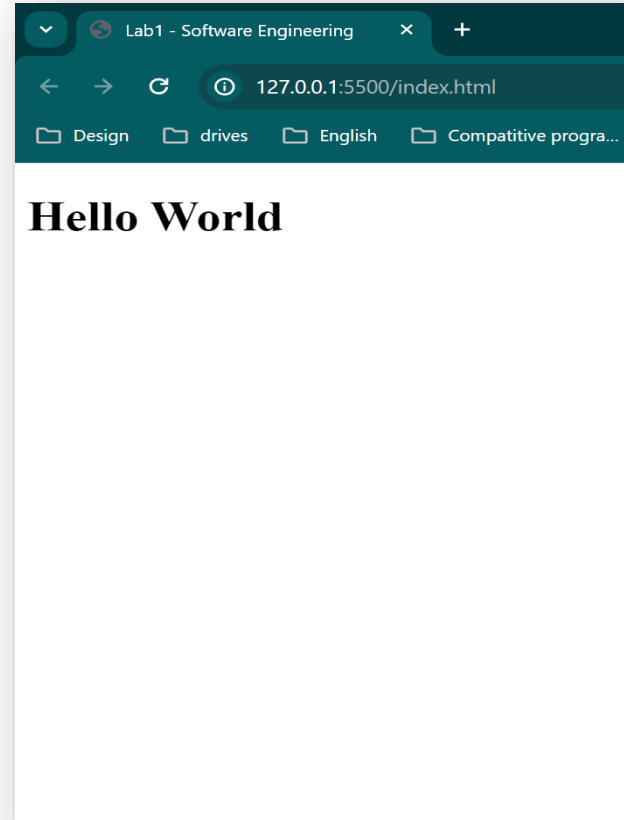layout of HTML elements on a webpage

JavaScript (JS):

Programming Language: adds interactivity and
dynamic behaviour to web pages

HTML   CSS   JS

MR.
WEBSITE

HTML Basic structure


```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta
    name="viewport"
    content="width=device-width, initial-scale=1.0"
  >
  <title>Lab1 - Software Engineering</title>
  <style>
    /* add your css code here */
  </style>
</head>

<body>
  <h1>Hello World</h1>
  <script>
    // add your JS code here
  </script>
</body>

</html>
```

HTML Basic structure

<!DOCTYPE html> :

It informs the web browser about the version of HTML being used in the document.
In this case, it signifies the use of HTML5.

<html>:

The **<html>** tag is the root element of an HTML document and wraps
all the content on the entire web page

It contains two main sections: the **<head>** section
and the **<body>** section.

HTML Basic structure

<head> :

- The <head> tag is a container for metadata and other information about the web page.

- It contains elements such as **<title>, <meta>, <link>, <style>,** and **<script>** tags.

- Content placed within the **<head>** tag is not displayed on the web page itself but is used by the browser and search engines.

- One of the most important sections should be scanned while doing performance auditing

  - Removing unneeded styles and scripts enhance web page performance.

HTML Basic structure

<body> :

- The <body> tag contains all the content that is visible on the web page.

- It includes elements such as headings, paragraphs, images, links, forms and other HTML elements that make up the actual content of the page.

- Placing <script> tags before the closing **</body>** is a best practice. Why?

HTML

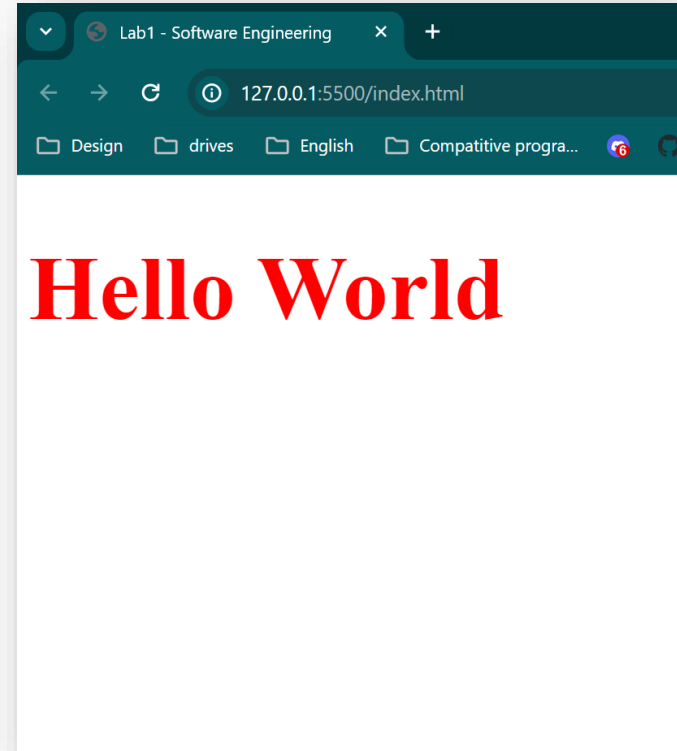What should you know after studying HTML ?

1. **Basic Structure**

2. **Tags and Elements**

3. **Attributes**

4. **Form**

5. **Links and Images**

6. **Lists**

7. **Tables**
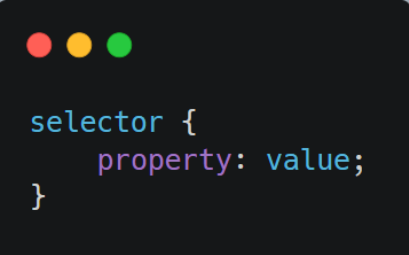
CSS

Basic Syntax and Selectors:

CSS

Basic Syntax and Selectors:

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

```css
h1 {
    font-size: 64px;
    color: red;
}
```

Element selector

```css
.main-header {
    color: red;
}
```

Class selector

```css
#main-header {
    color: red;
}
```

ID selector

CSS

What should you know after studying CSS ?

1. **Basic syntax and selectors**

2. **Box Modal**

3. **Typography**

4. **Colors and Backgrounds**

5. **Layouts**

6. **Responsive Design**

7. **Flexibility and Grid Systems**

Optional

1. **CSS Preprocessors**

2. **Transitions and Animations**

3. **CSS Frameworks**

4. **CSS Architecture**

5. **Accessibility**

JS

Basic Syntax and Selectors:

```html
<body>
  <h1>Hello World</h1>
  <h3>Today Lab: <span id="current-date"></span></h3>
  <script>

    const currentDate = new Date();
    const day = currentDate.getDate()
    const month = currentDate.getMonth() + 1
    const year = currentDate.getFullYear()

    const weekdays =
    ['Sunday', 'Monday', 'Tuesday', 'Wednesday',
     'Thursday', 'Friday', 'Saturday'];

    const currentWeekday =
     weekdays[currentDate.getDay()];

    document.getElementById('current-date').innerText =
    `${currentWeekday} - ${day}/${month}/${year}`
  </script>
</body>
```

JS

What should you know after studying JS ?

- **Basic Concepts**:
    - Data types (strings, numbers, booleans, etc.)
    - Variables and constants
    - Operators (arithmetic, assignment, comparison, logical)
    - Control flow (if statements, switch statements, loops)
- **Functions:**
    - Declaring functions
    - Parameters and arguments
    - Return statements
    - Function expressions vs. function declarations
    - Arrow functions
- **Arrays and Objects:**
    - Creating and manipulating arrays
    - Array methods (map, filter, reduce, etc.)
    - Working with objects (properties, methods)

JS

What should you know after studying JS ?

- **Scope and Closures**:
    - Understanding scope (global scope, function scope, block scope)
    - Closure concept
- **Asynchronous JavaScript:**
    - setTimeout and setInterval
    - Async/Await syntax and Promises
    - Fetch API for making HTTP requests
- **DOM Manipulation:**
    - Accessing and modifying HTML elements
    - Event handling (click events, form submissions, etc.)
    - Manipulating CSS styles
    - Creating and removing HTML elements dynamically
- **ES6+ Features:**
    - Let and Const keywords
    - Destructuring assignment
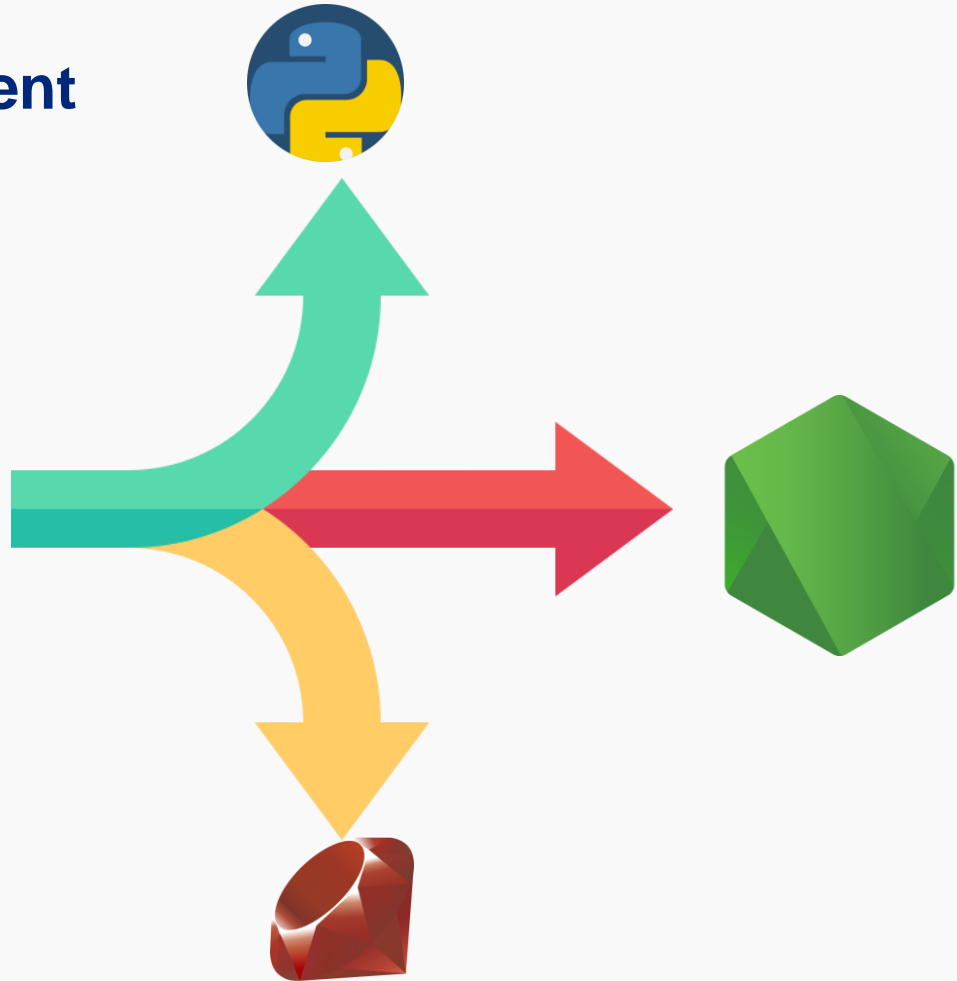   - Spread and rest operators
    - Modules (import/export)

Backend development

Database

Operating Systems

Distributed Systems

Backend development

- **Programming Languages**:
  - Choose a backend programming language
  Examples: Python, JavaScript (Node.js), Java, Ruby,GO, PHP, etc.

- **Web Frameworks:**
  - Learn popular backend frameworks for your chosen language
  Examples: Flask/Django (Python), Express.js (Node.js), Spring Boot (Java), Ruby on Rails (Ruby), Laravel (PHP), etc.

- **Databases**:
  - Understand different types of databases:
  Relational (SQL): MySQL, PostgreSQL, SQLite, etc. VS NoSQL: MongoDB, Firebase, Cassandra, etc.

- **API Development:**
  - Design and build RESTful APIs (Application Programming Interfaces) to communicate with frontend

Questions

Thank You 😇