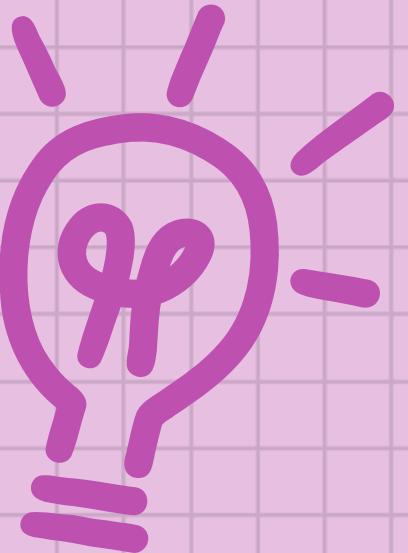


POWERPUFF GIRLS



Serine Sefardjelah



Maroua Graba

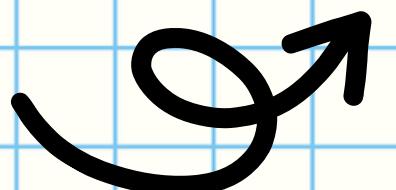
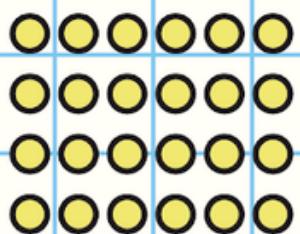


Lyna Razane Mguellati

20



BUILD A RESPONSIVE
WEB APP WITH
REACT



20

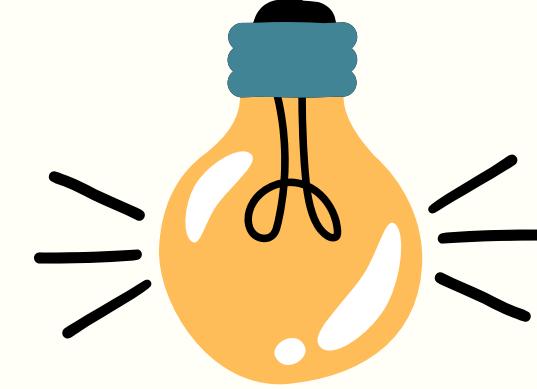




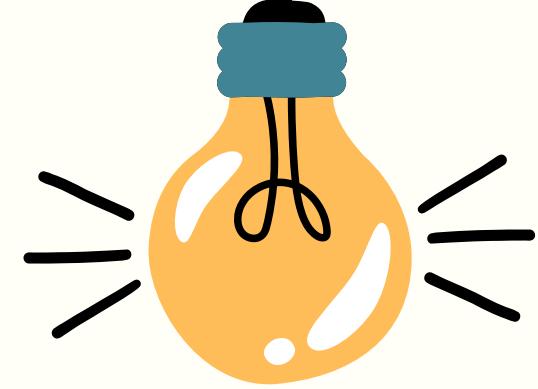
CODING EXPERIENCE

How familiar are you with web dev ?

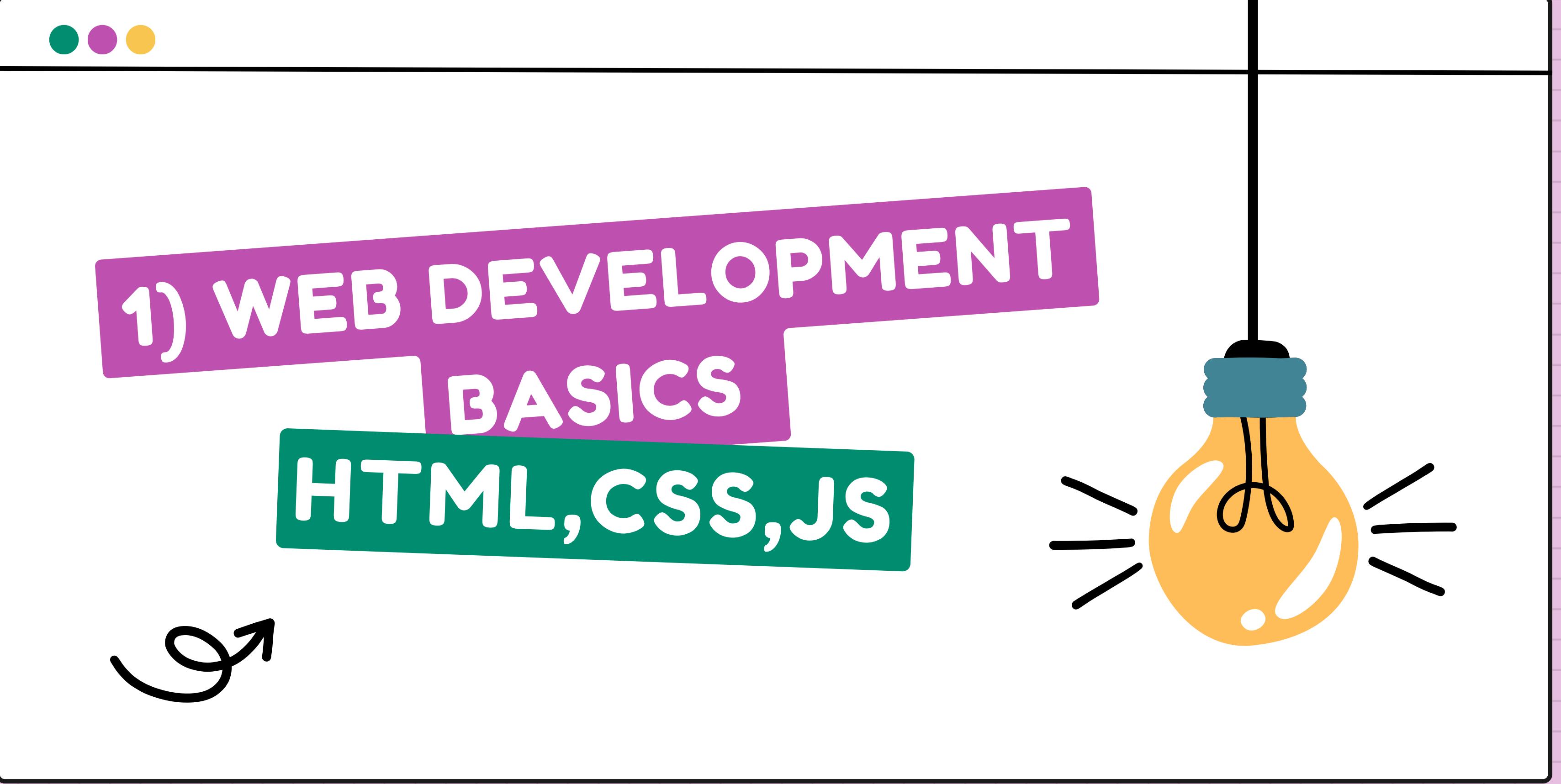




CONTENT



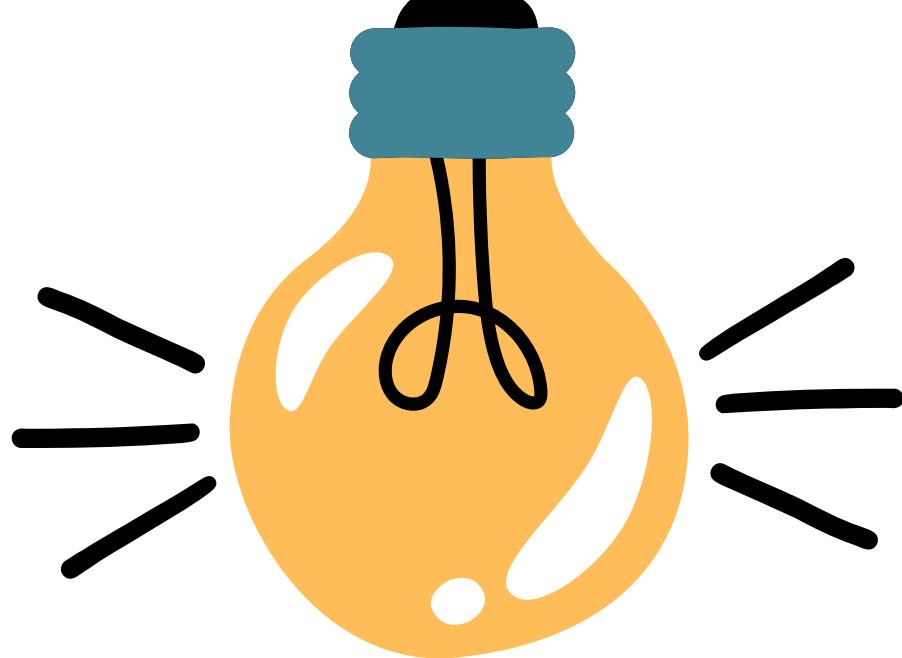
- 1 **Web Development Basics : HTML , CSS , JavaScript**
- 2 **React basic concepts**
- 3 **Hands-On Coding - TODO App**



1) WEB DEVELOPMENT

BASICS

HTML,CSS,JS



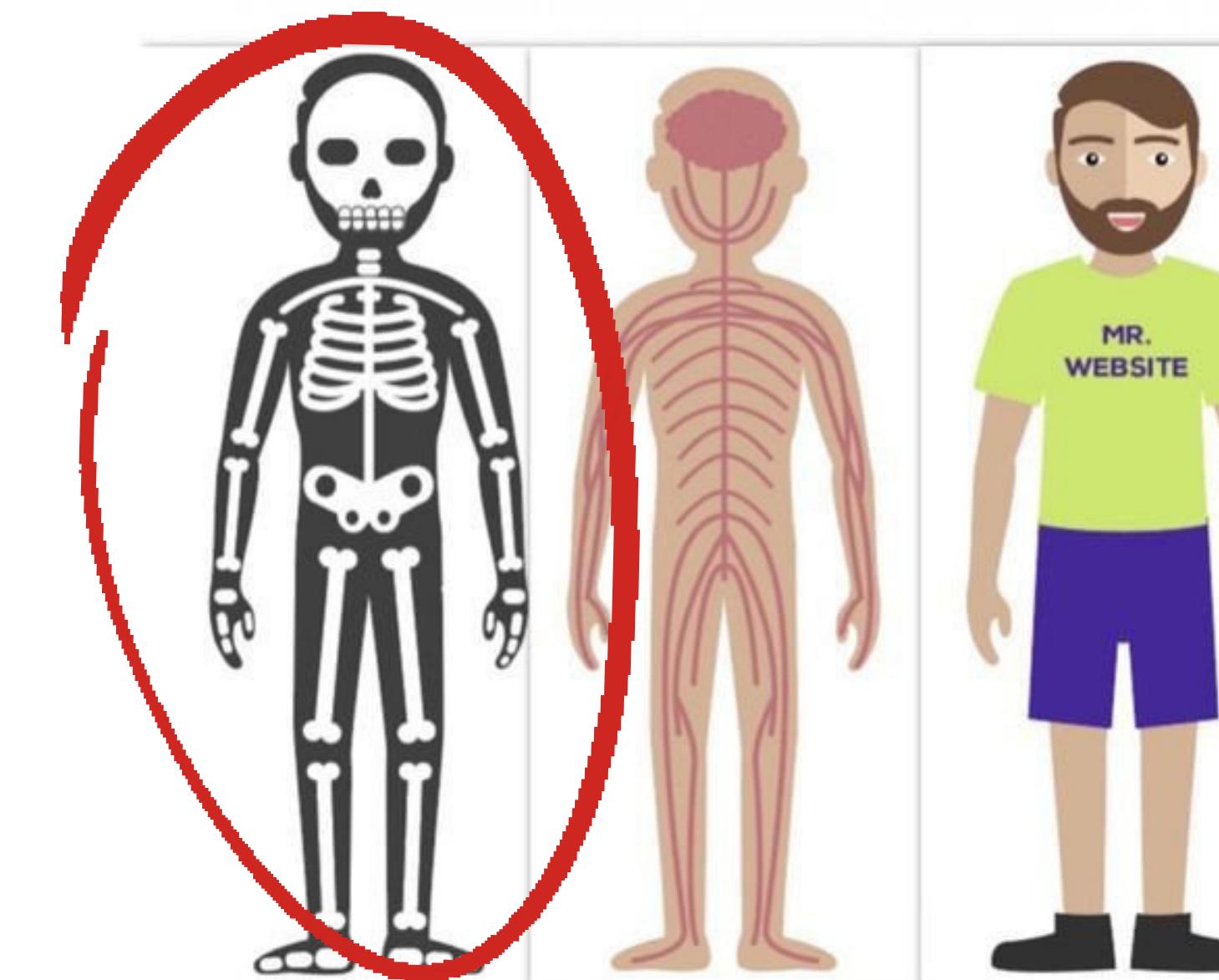


HTML

JS

CSS

HTML

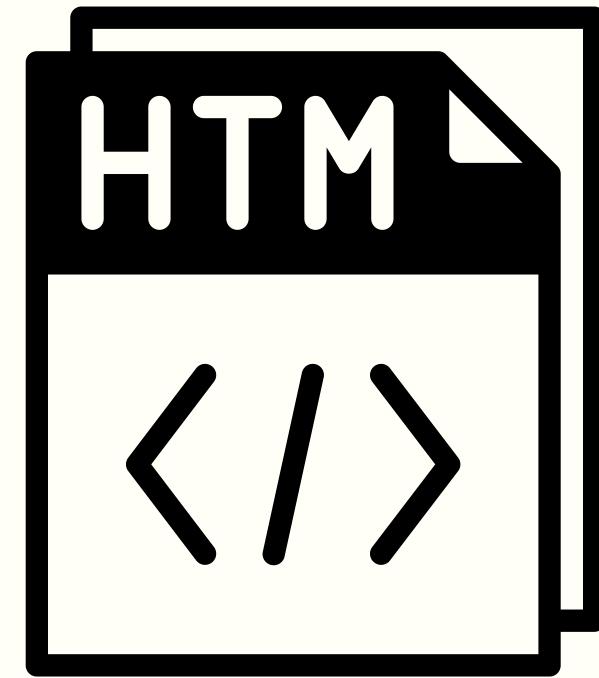


HTML



- **HTML is the standard markup language for creating Web pages**
- **A markup language is a system of codes or tags that provides instructions on how to structure and format text**
- **It's the backbone of web content, providing the structure and meaning to what we see on websites**

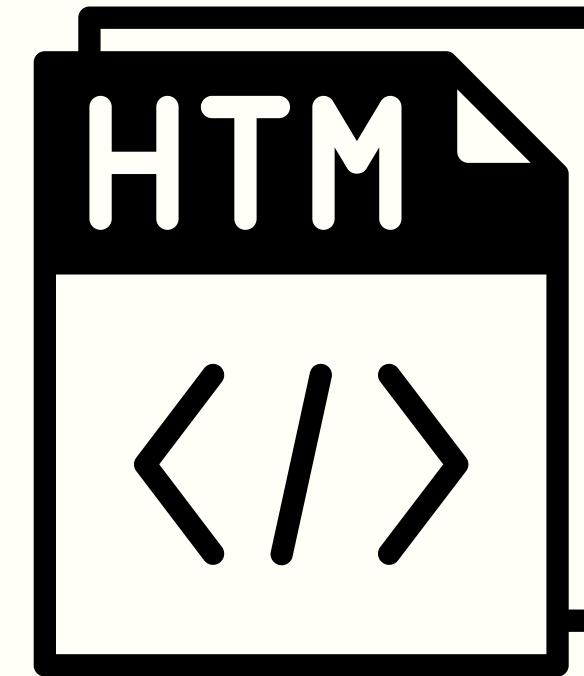
HTML



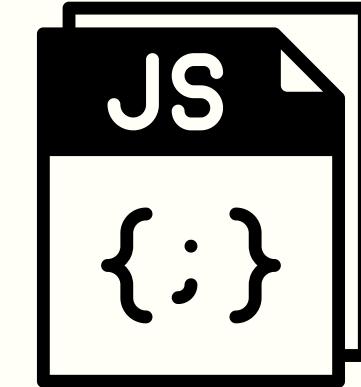
index.html

THE INSTRUCTION MANUAL

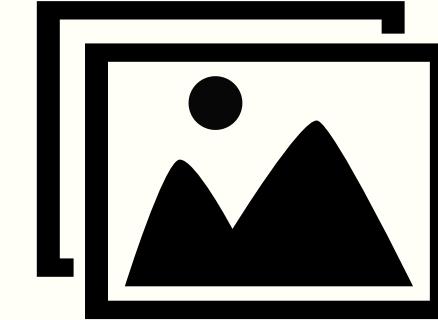
HTML



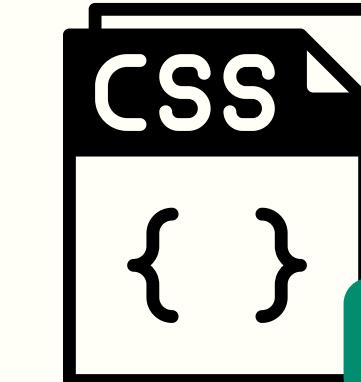
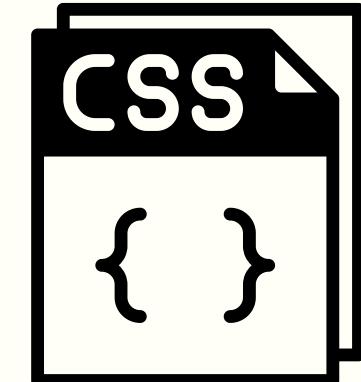
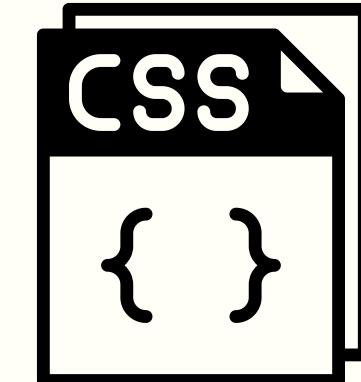
index.html



JAVASCRIPT



IMAGES



STYLE SHEETS

THE BROWSER SEES

```
5 index.html index.html\...

<html>

  <head>

    <title>My First Website</title>
    <meta name="description" content="yay!" />

  </head>

  <body>

    <h1>HTML is awesome!</h1>

    <p>Look mom, I built a website 🤗</p>

    <div>
      <a href="https://fireship.io">learn to code</a>
    </div>

  </body>

</html>
```

WHAT WE SEE

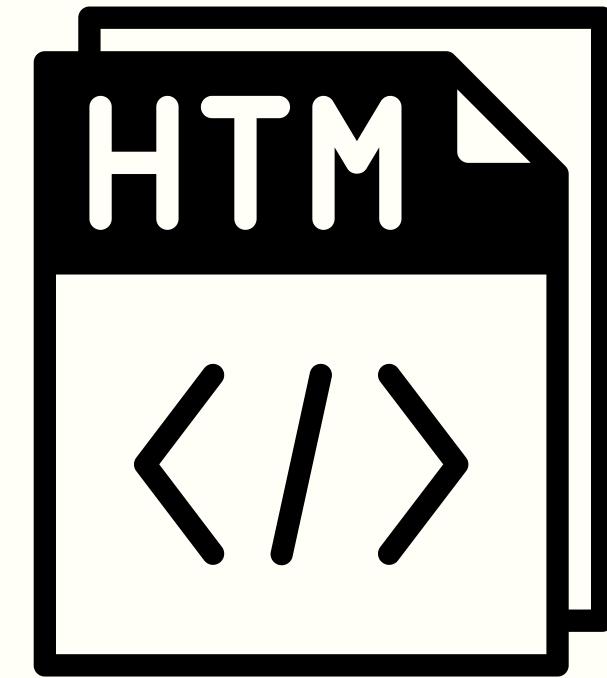
← → ⌂ ⌂ File | C:/Users/delan/apps/html-app/index.html

HTML is awesome!

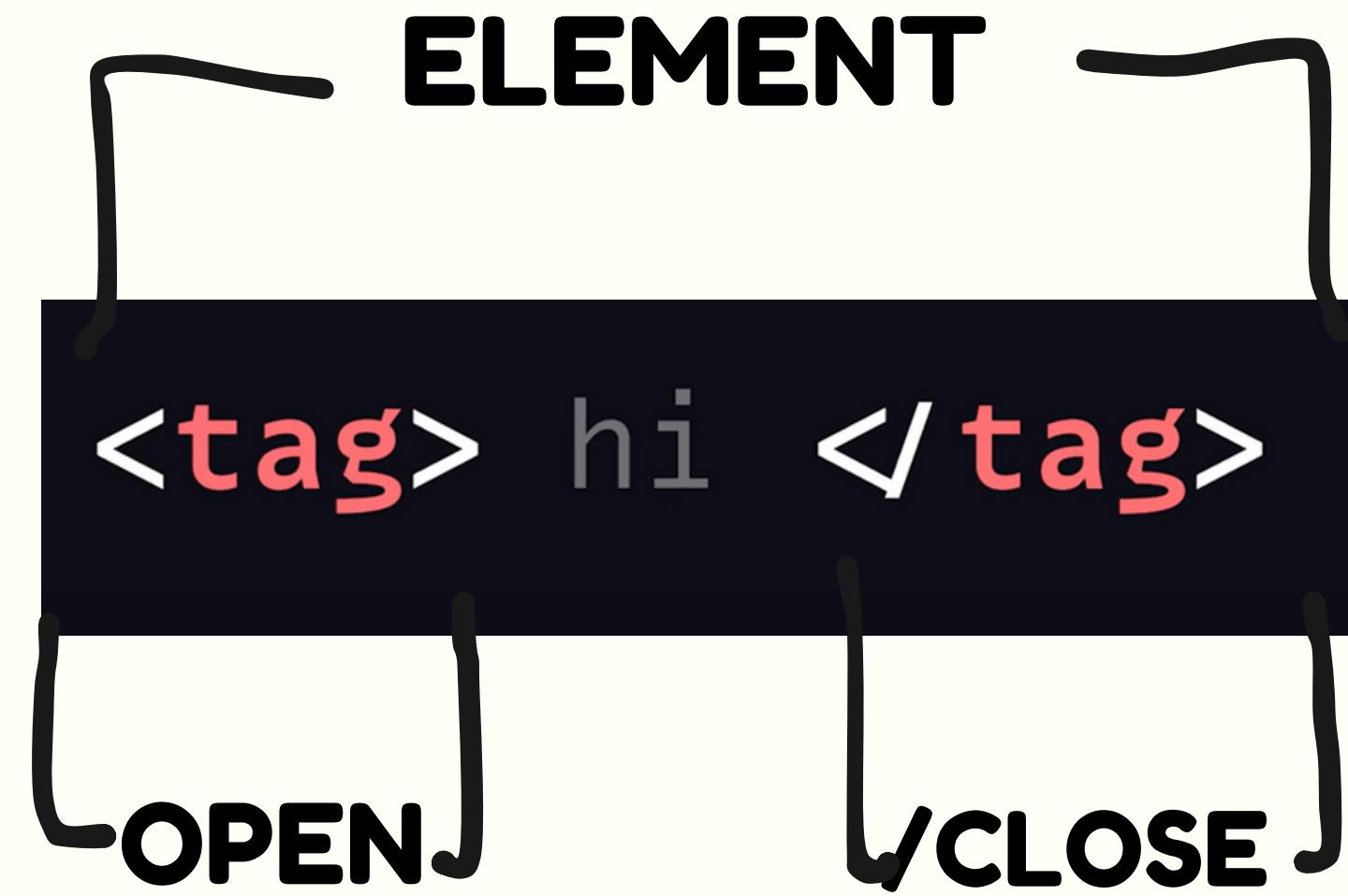
Look mom, I built a website 🤗



HTML



index.html



Surfing the web

UNORGANIZED TEXT

I love development check the first website i built with my bare hands |

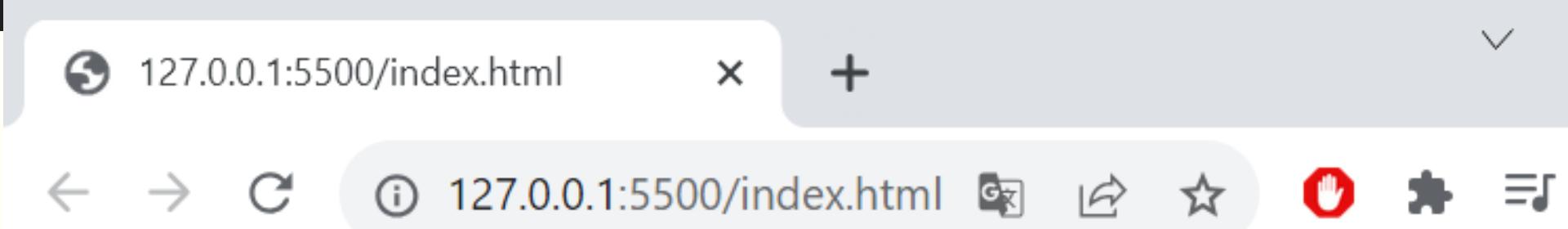
PARAGRAPH

```
<h1>Surfing the web</h1>
```

HEADING

```
<p> I love development check the first website i built with my bare hands </p>
```

ON THE BROWSER >>



Surfing the web

I love development check the first website i built with my bare hands

EXAPMLES OF SOME HTML </> ELEMENTS



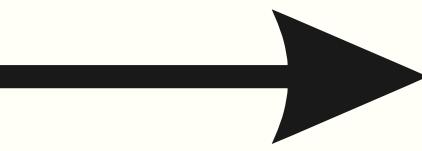
HEADINGS :

Headings are titles or subtitles that you want to display on a webpage.

<h1> defines the most important heading

<h6> defines the least important heading.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

EXAPMLES OF SOME HTML </> ELEMENTS



PARAGRAPHS:

The HTML `<p>` element defines a paragraph
A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

```
<p>This is a paragraph</p><p>This is also paragraph</p>
<h1>heading between paragraphs</h1> <p>Another paragraph</p>
```

This is a paragraph

This is also paragraph

heading between paragraphs

Another paragraph

EXAPMLES OF SOME HTML </> ELEMENTS



LISTS:

```
<h1>Unordered list</h1>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>

<h1>Ordered list</h1>
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
</ol>
```



← → ⌂ ① 127.0.0.1:5500/index.html

Unordered list

- Item 1
- Item 2

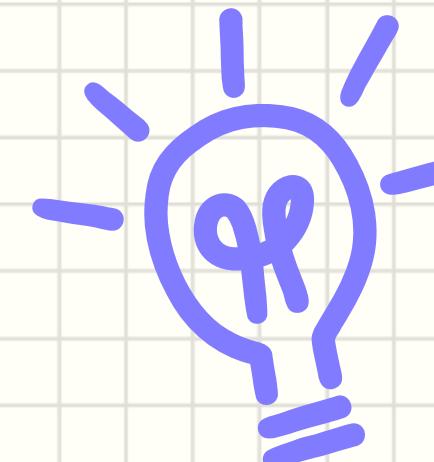
Ordered list

1. Item 1
2. Item 2

```
<h3>Paragraph without line break</h3>  
<p>This is the first line.  
This is the second line</p>
```

```
<h3>Paragraph with line break</h3>  
<p>This is the first line.<br> This is the second line</p>
```

```
<h3>paragraph with a horizontal line element </h3>  
<p>Content above<hr>Content below</p>
```



Paragraph without line break

This text has a line break And the next part is on a new line.

Paragraph with line break

This text has a line break
And the next part is on a new line.

paragraph with a horizontal line element

Content above

Content below

**
 : line break**

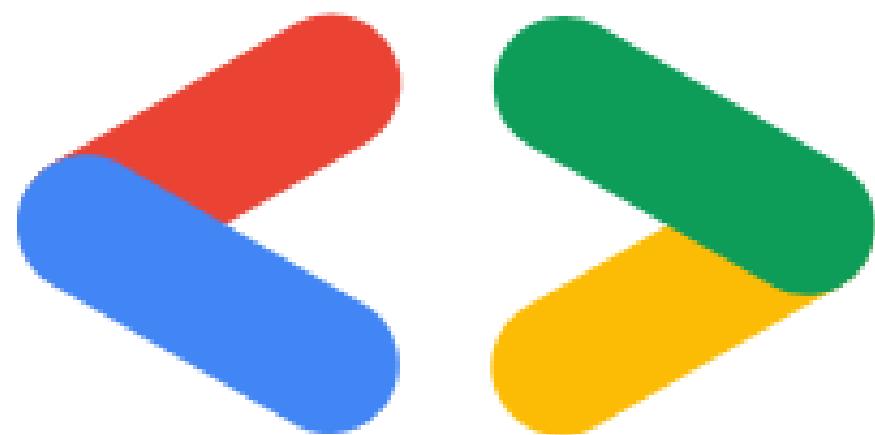
**<hr> : a horizontal rule, used
to separate content**

IMAGE ELEMENT >>


```
<h3>This is an image</h3>  
  

```

This is an image



This is an image

IF THE IMAGE CANNOT BE LOADED >>



GLOBAL ATTRIBUTES

CLASS : <p class=""></p>

```
<h2 class="city">algiers</h2>
<p>Algiers is the capital of Algeira.</p>
```

```
<h2 class="city eaurope ">Paris</h2>
<p>Paris is the capital of France.</p>
```

ID : <h1 id="">My </h1>

```
<h1 id="myHeader">Cities</h1>

<h2 class="city">algiers</h2>
<p>Algiers is the capital of Algeira.</p>
```



```
<button>click me!</button>
```

click me!

```
<form>
  <label>First name:</label> <br />
  <input type="text" /> <br />
  <label>Last name:</label> <br />
  <input type="text" />
</form>
```

First name:

Last name:

THE STRUCTURE OF AN HTML FILE



```
<!DOCTYPE html> THIS DOCUMENT IS AN HTML5 DOCUMENT
```

```
<html> THE ROOT ELEMENT OF AN HTML PAGE
```

```
<head> INCLUDES META-INFO ABOUT THE DOCUMENT
```

```
<meta charset="UTF-8" /> UTF-8 IS A CHARACTER ENCODING SYSTEM
```

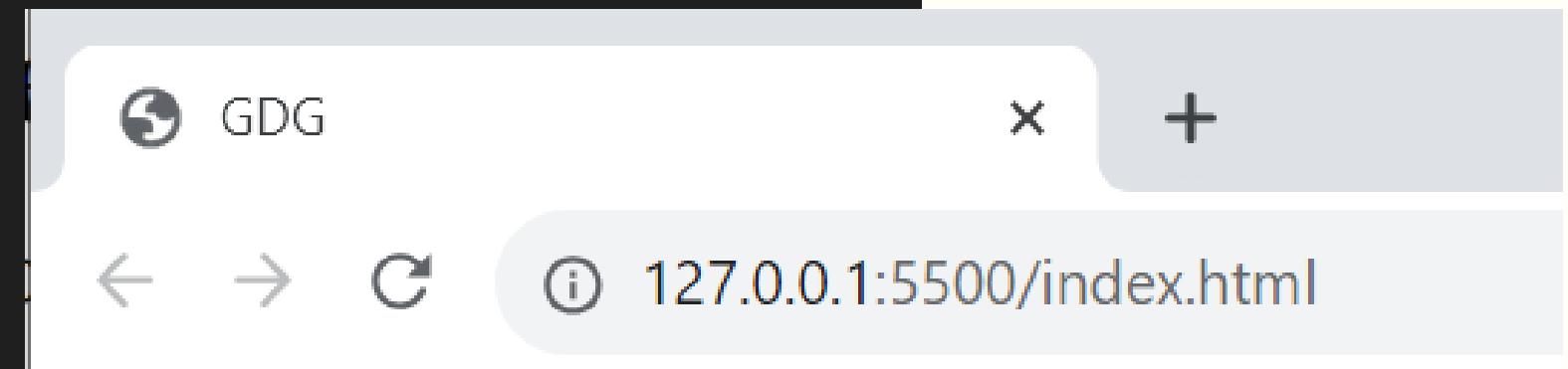
```
</head>
```

```
<body> CONTAINS THE MAIN CONTENT
```

```
<h1>welcome to GDG Algiers</h1>
<p class="name">- GDG School 2023 -</p>
```

```
</body>
```

```
</html>
```

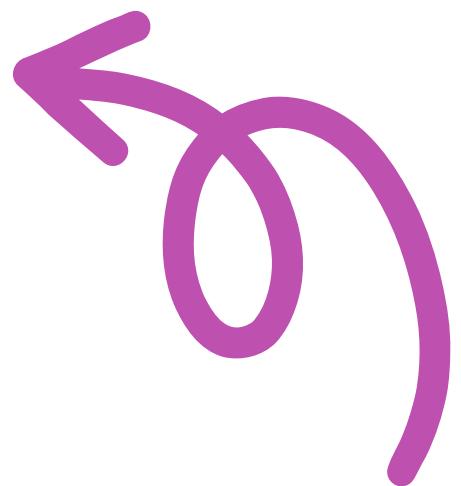


welcome to **GDG Algiers**

- GDG School 2023 -



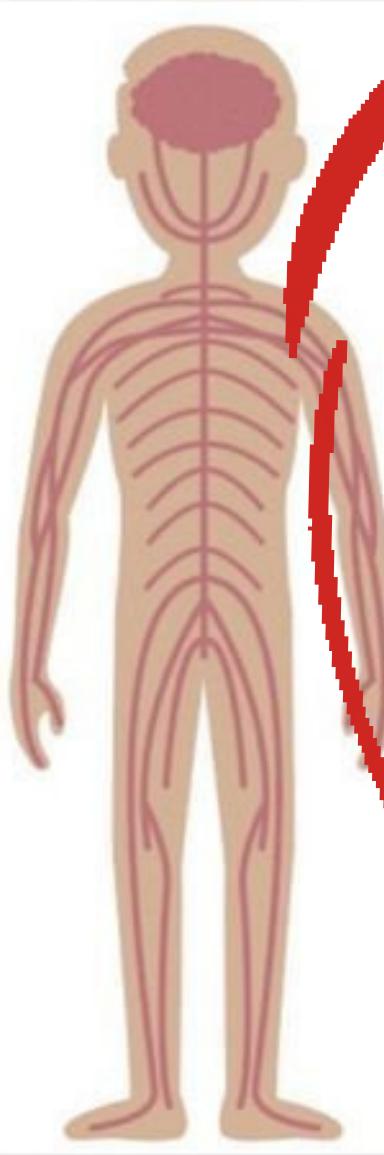
CSS



HTML



JS



CSS



- **CSS stands for Cascading Style Sheet**
- **Styles the html page : colors, fonts, shadows ...**
- **CSS describes how HTML elements are to be displayed on screen**

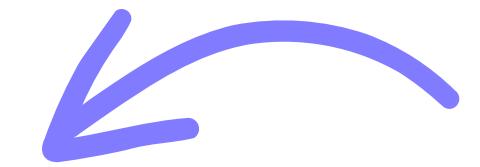
It's what makes the website PRETTY

How to use it



Inline CSS

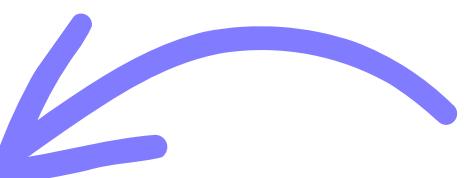
```
<p style="color: blue;">This is a paragraph.</p>
```



inline: add style as an attribute

Internal CSS

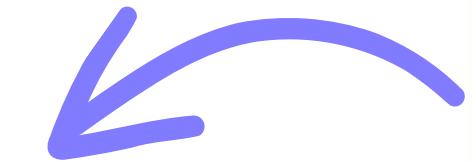
```
<head>
<style type = text/css>
  body {background-color: blue;}
  p { color: yellow;}
</style>
</head>
```



internal : add a style tag inside
the head tag of an html file

External CSS

```
<head>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
```



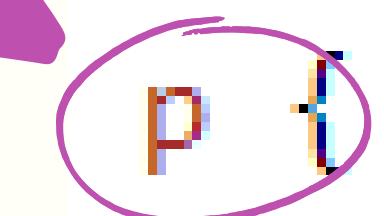
External : use css in a different file

CSS syntax



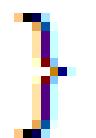
“p” is a selector

it points to the HTML
element you
want to style: <p>



**“red” is a
property value**

```
color: red;  
text-align: center;
```



“color” is a property

Css Selectors



ID SELECTORS

```
#gros{  
    font-size: 24px;  
}
```

ELEMENT SELECTORS

```
body  
{  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    flex-direction: column;  
}
```

CLASS SELECTORS

```
.bleu{  
    color: blue;  
}
```

```
h1 h2 {  
    color: blue;  
}
```

Properties



COLORS

color
background-color

POSITIONS

padding
margin

FONTS & TEXTS

font-weight

ANIMATIONS

animation

EFFECTS

:hover

Example

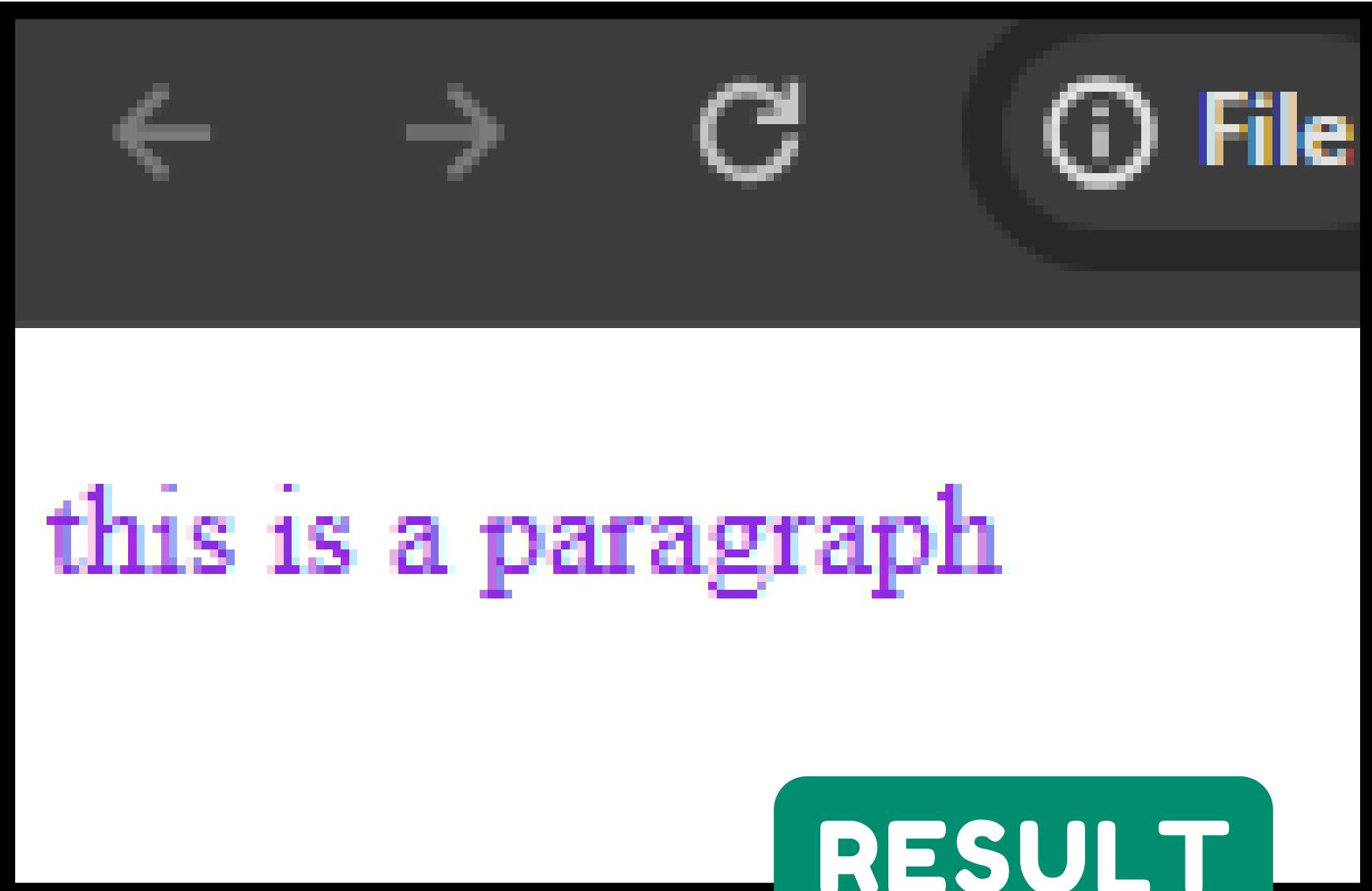


HTML

```
<body>
|   |   <p> this is a paragraph </p>
</body>
```

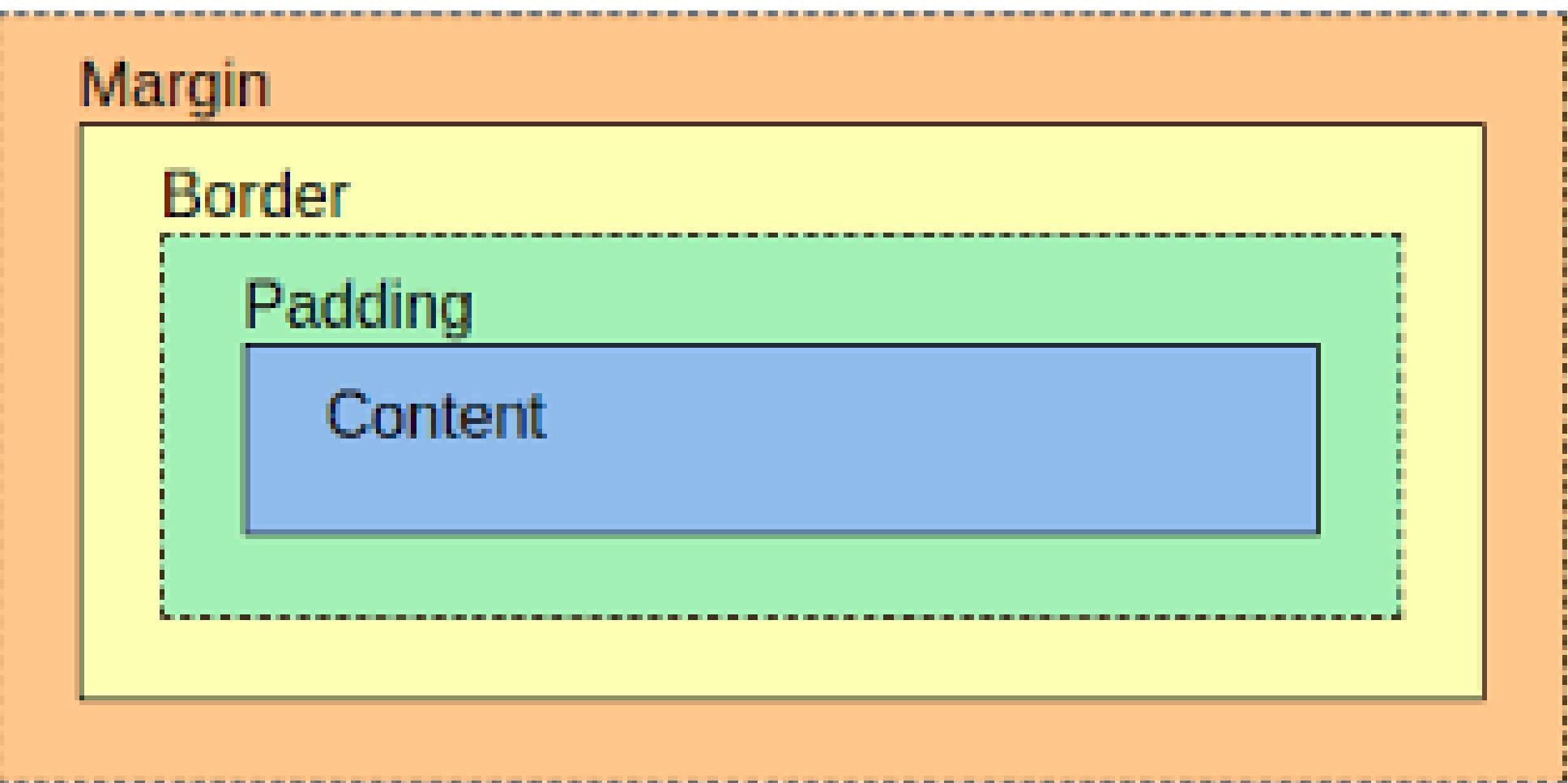
CSS

```
p{
|   color: blueviolet;
|}
```



RESULT

Css Box Model



example

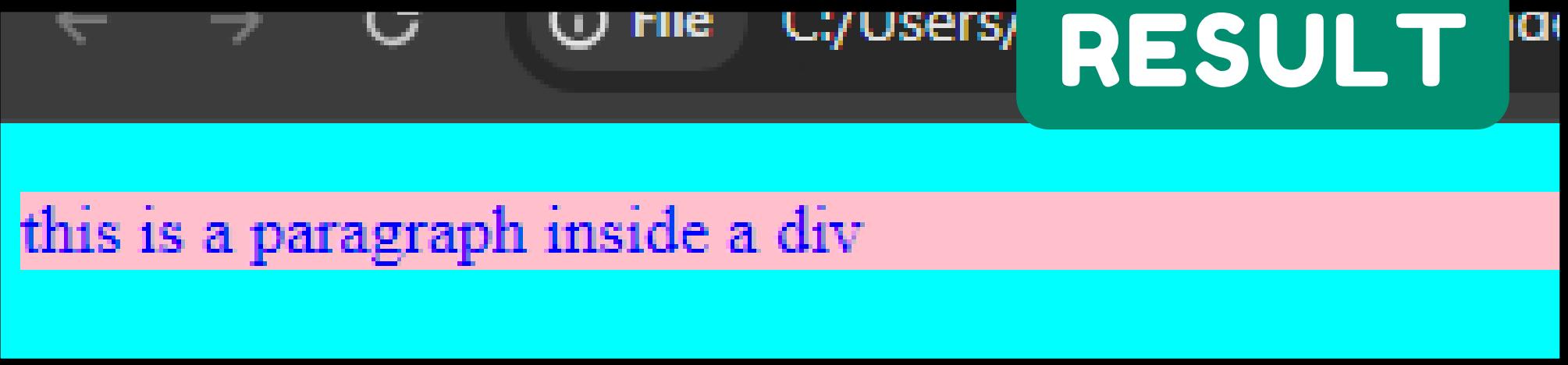
HTML

```
<body>  
    <div class="container">  
        <p id="par"> this is a paragraph inside a div </p>  
    </div>  
</body>
```

CSS

```
body{  
    background-color: aqua;  
}  
  
.container{  
    background-color: pink;  
}  
  
#par{  
    color: blue;  
}
```

RESULT



example



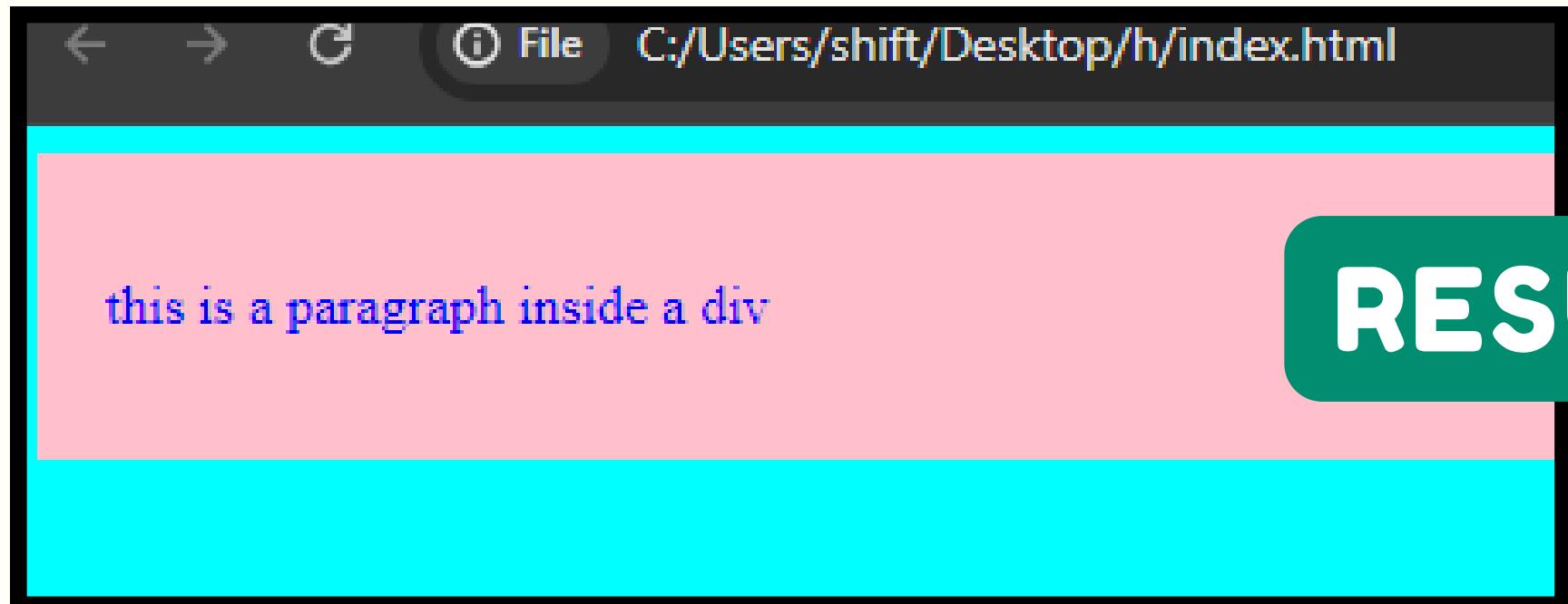
HTML

```
<body>
  <div class="container">
    <p id="par"> this is a paragraph inside a div </p>
  </div>
</body>
```

CSS

```
.container{
  background-color: #f08080;
  padding: 20px;
}
```

RESULT





HTML

```
<body>  
    <div class="container">  
        <p id="par"> this is a paragraph inside a div </p>  
    </div>  
</body>
```

```
.container{  
    background-color: # pink;  
    padding: 20px;  
    margin: 5%;  
}
```

CSS

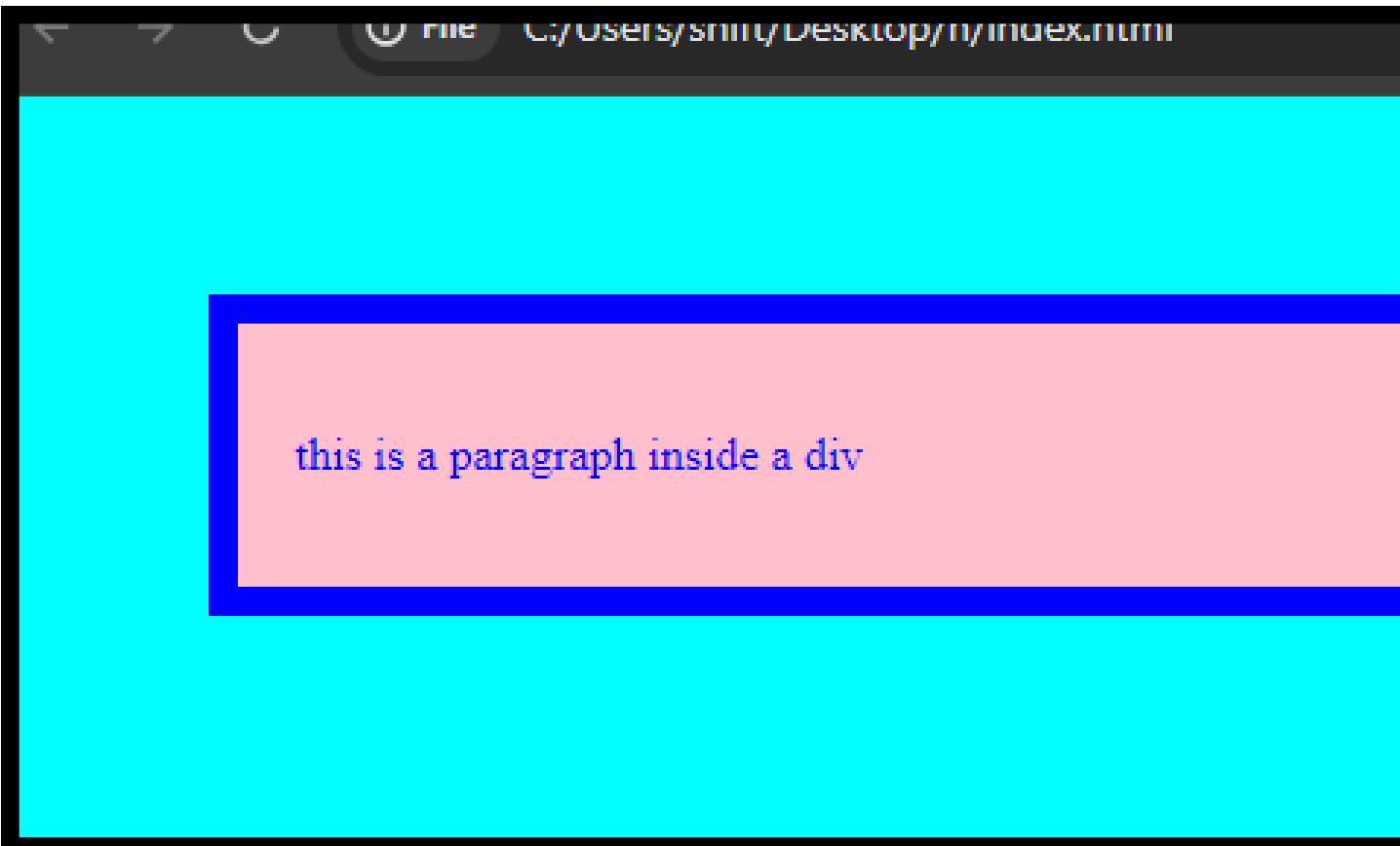


RESULT



HTML

```
<body>  
    <div class="container">  
        <p id="par"> this is a paragraph inside a div </p>  
    </div>  
</body>
```



CSS

```
.container{  
    background-color: pink;  
    padding: 20px;  
    margin: 5%;  
  
    border-width: 10px;  
    border-style: solid;  
    border-color: blue;  
}
```

RESULT

Css Media queries



- Media query is a CSS technique to make responsive web apps
- It uses the @media rule to include a block of CSS properties only if a certain condition is true.

```
✓ @media only screen and (max-width: 600px) {  
    ✓ body {  
        | background-color: lightblue;  
        | }  
    | }  
| }
```



**using Pure CSS and
writing media queries
codes**



**Using CSS
frameworks like
tailwind**



a tailwind syntax example

The image shows a code editor interface with four tabs: `index.html`, `tailwind.config.js`, `package.json`, and `build.css`. The `index.html` tab contains the following HTML code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
initial-scale=1.0">
6      <title>Document</title>
7      <link rel="stylesheet" href="/build.css">
8    </head>
9    <body>
10      <button class="flex items-center px-4 py-3 text-white
bg-blue-500 hover:bg-blue-400"></button>
11    </body>
```

The `tailwind.config.js` tab contains the configuration for Tailwind CSS.

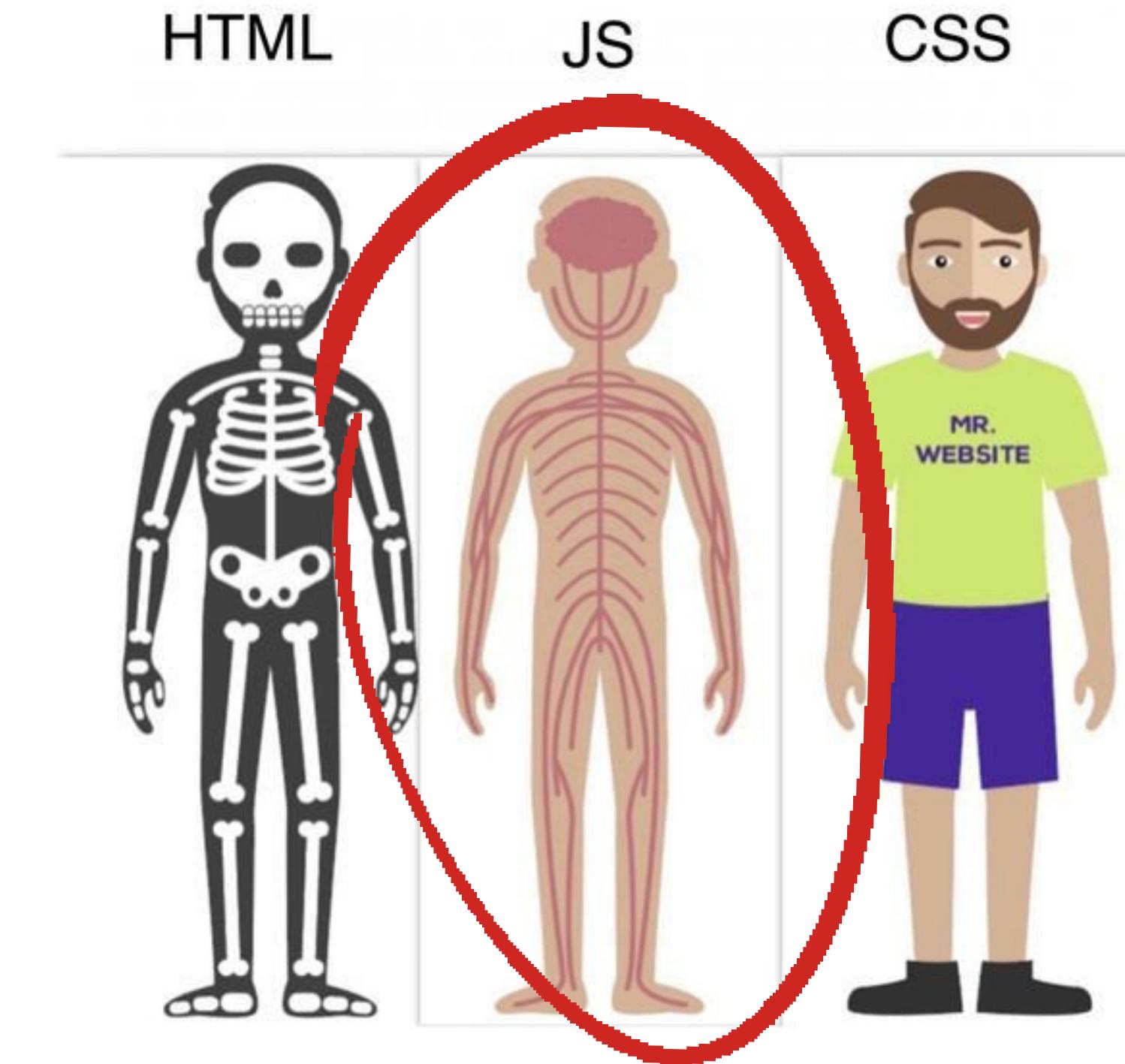
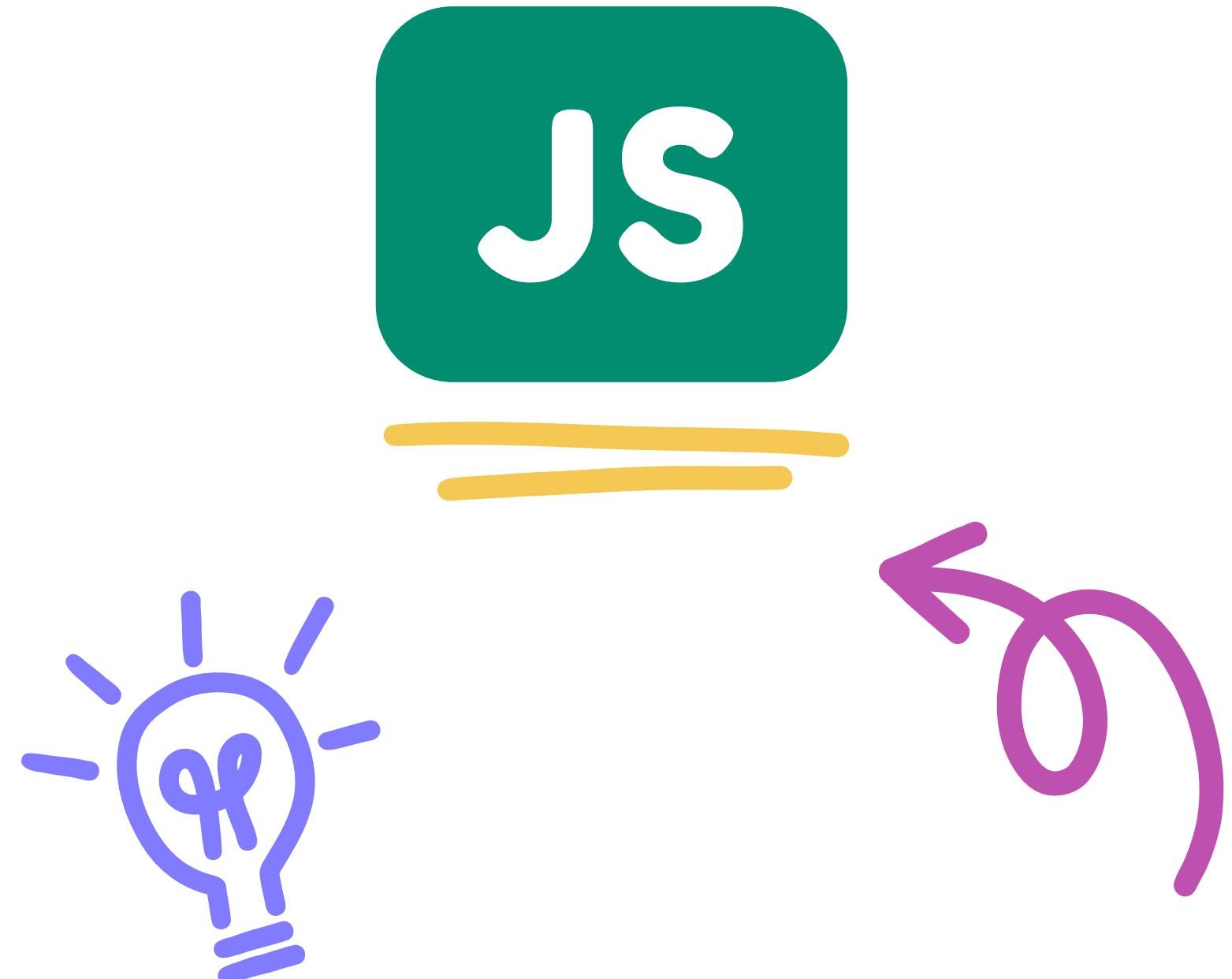
The `package.json` tab contains the project's package information.

The `build.css` tab shows the generated CSS code, which includes Tailwind utility classes:

```
14  .px-4 {
15    padding-left: 1rem;
16    padding-right: 1rem
17  }
18
19  .py-3 {
20    padding-top: 0.75rem;
21    padding-bottom: 0.75rem
22  }
23
24  .text-white {
25    --tw-text-opacity: 1;
26    color: rgb(255 255 255 / var(--tw-text-opacity))
27  }
28
29  .hover\:bg-blue-400:hover {
30    --tw-bg-opacity: 1;
31    background-color: rgb(96 165 250 / var(--tw-bg-opacity))
32  }
```

The terminal pane at the bottom shows the build process output:

```
Rebuilding... done in 3ms.
Rebuilding... Done in 7ms.
Rebuilding... Done in 4ms.
Rebuilding... Done in 5ms.
```



JS

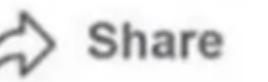


What's difference between Java and JavaScript ?

1.2k



Like



Jay Prakash

It is like "car and carpet".

Like · Reply

210



Faisal

It's like "moon and honeymoon".



Haha · Reply

2.3k

IT'S A Frontend

IT'S A Backend

IS IT... BOTH?

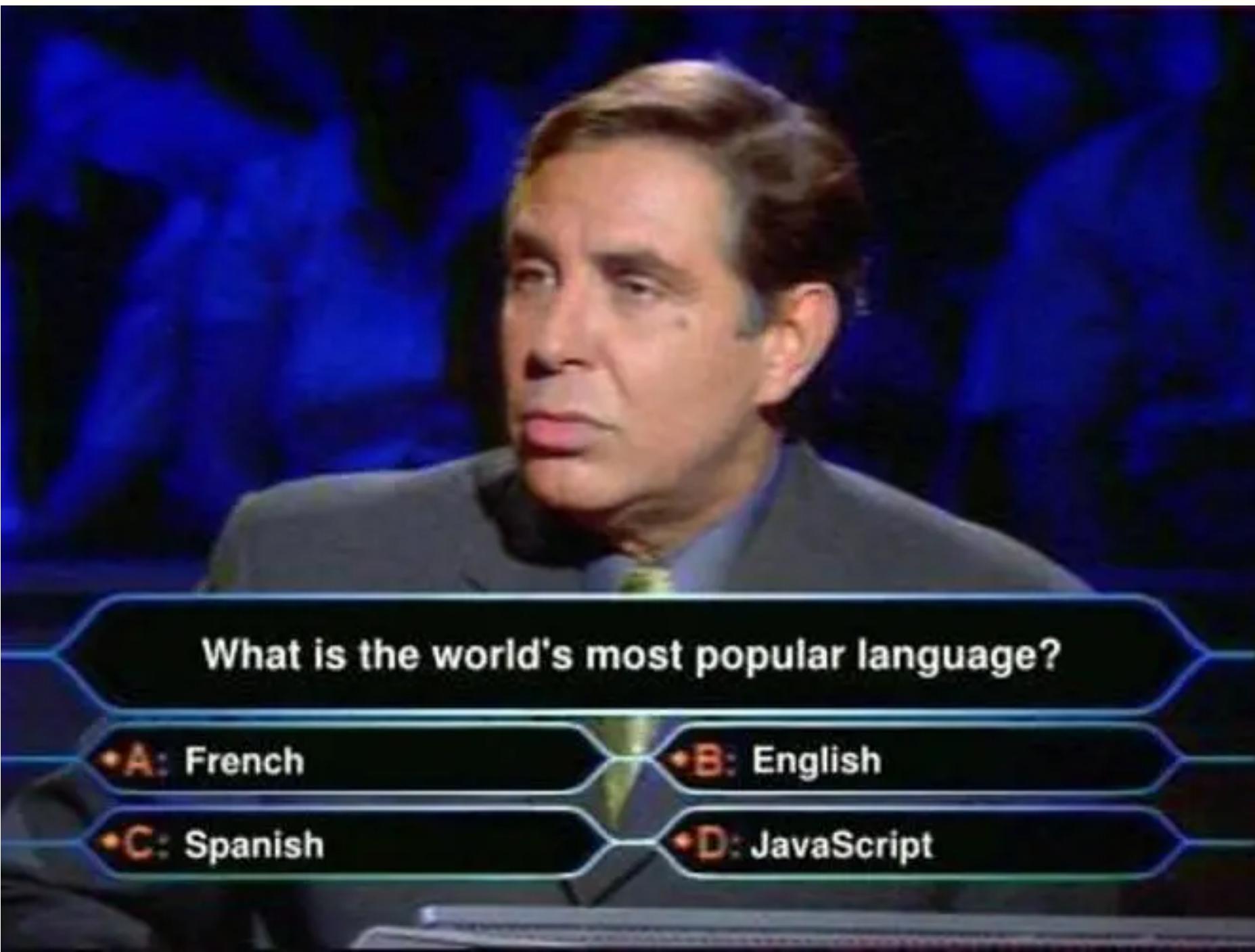
IT'S BOTH.

I HATE IT



It's a Meteor !

JS



JS



JS SYNTAX

Move to the console :}

JS OPERATORS

`==` equal to

`x == 8` false

`x == 5` true

`x == "5"` true

`====` equal value and equal type

`x === 5` true

`x === "5"` false

`!=` not equal

`x != 8` true

`!==` not equal value or not equal type

`x !== 5` false

`x !== "5"` true

`x !== 8` true

Move to the console :)

EVENTS

- are "things" that happen to HTML elements.
- When JavaScript is used in HTML pages, JavaScript can "react" on these events.
- check the example to make it clearer

JS



JS



WHAT'S ES6 ?

ECMAScript 6



- ECMAScript was created to standardize JavaScript
- ES6 is the 6th version of ECMAScript
- React uses ES6

ES6 VARIABLES

```
var x = 5.6  
let x = "GDG School"  
const x = "GDG is the Best"
```

ARRAY METHODS

The `.map()` method allows you to run a function on each item in the array, returning a new array as the result.

```
const myArray = ['apple', 'banana', 'orange'];
```

```
const myList = myArray.map((item) => console.log(item))
```

ARRAY METHODS

.filter(), as its name indicates, allows us to filter certain elements out of our array.

```
const myArray = ['apple', 'banana', 'orange'];

const myList2 = myArray.filter((item)=> !item.startsWith('o'))
console.log(myList2)

▶ (2) ['apple', 'banana']
```

SPREAD OPERATOR

[...]

```
const numbersOne = [1, 2, 3];
const numbersTwo = [4, 5, 6];
const numbersCombined = [...numbersOne, ...numbersTwo];
console.log(numbersCombined)
```

FUNCTIONS

```
function add(a, b) {  
    return a + b;  
}  
  
let result0 = add(4, 3)  
console.log(result0)
```

DECLARING A FUNCTION

CALLING IT

JS



FUNCTIONS

```
const add = function (a, b) {return a + b};  
let result1 = add(4, 3);  
console.log(result1)
```

ARROW FUNCTION

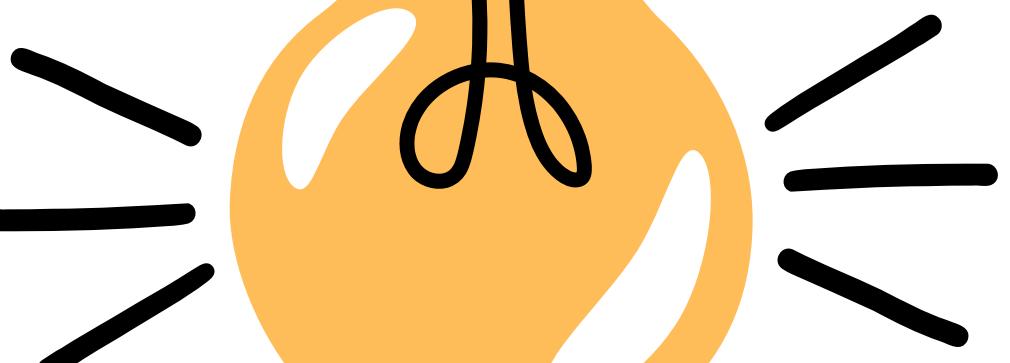
ANONYMOUS FUNCTION

```
const add = (x, y) => { return x + y };  
let result2 = add(4, 3);  
console.log(result2)
```



2) REACT BASIC CONCEPTS

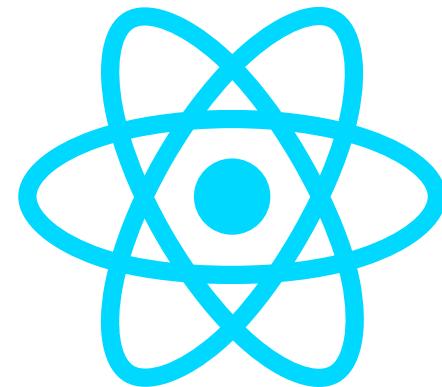
WHY IS IT SO COOL ?



ee



WHAT IS REACT?



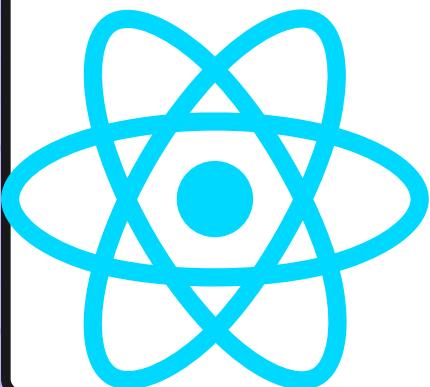
React is a JavaScript **LIBRARY** for building fast and interactive user interfaces





WHY DO WE USE REACT

- it's fast because it keeps track of only html elements that need to change
- React has a virtual DOM
- Component-Based Architecture
- JSX syntax
- SPA



SPA



Single Page Application



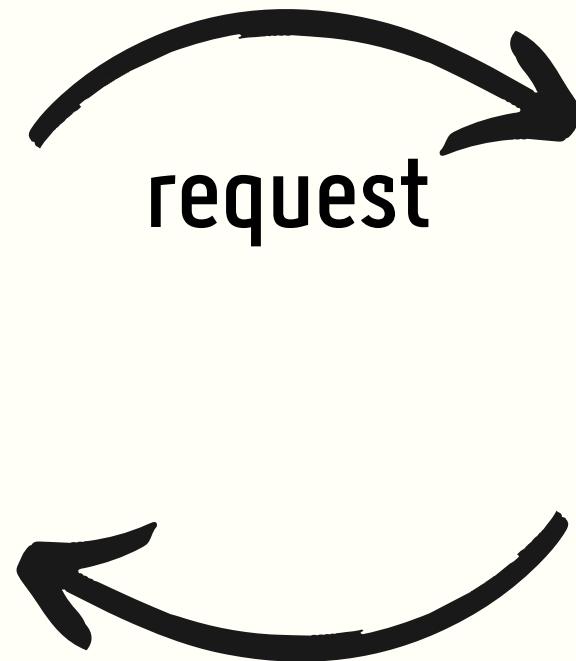
Multi Page Application



Single Page Applications (SPAs) : one html page that changes content

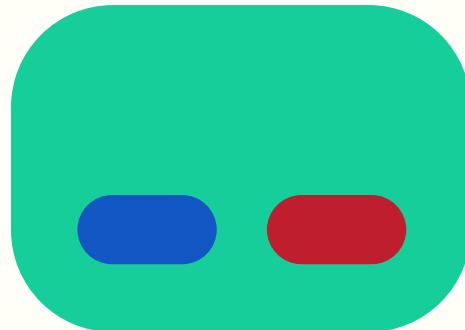


traditional



request

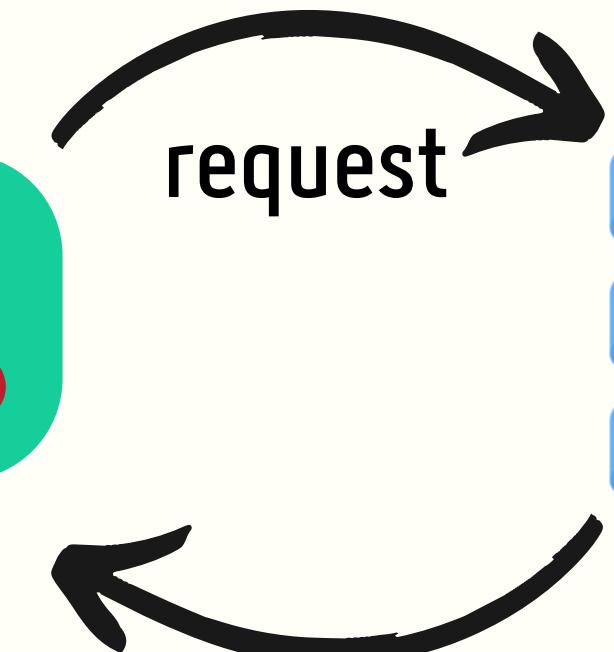
response:



entire html page with
that **MINOR change**

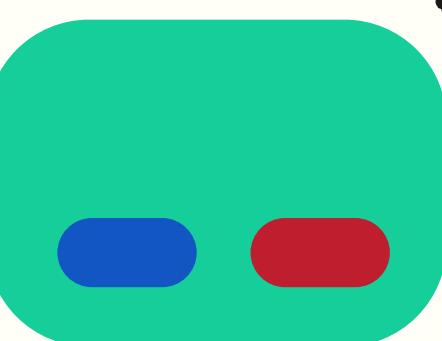


SPA

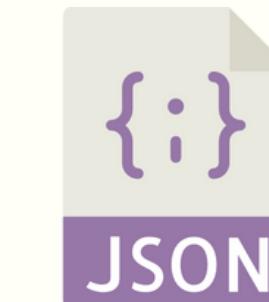


request

response:



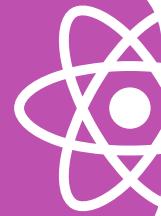
the existing page
gets updated with
the new object



Json object that represents
only the changed element



VIRTUAL DOM

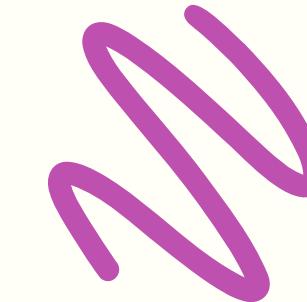


DOM

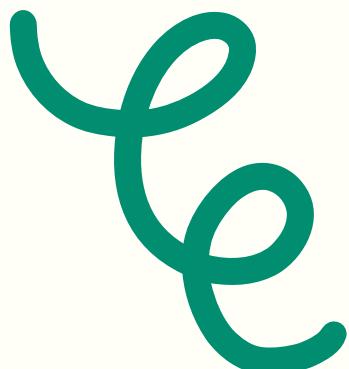


DOM : document object model

**tree structure of objects in the html file
built by the browser**

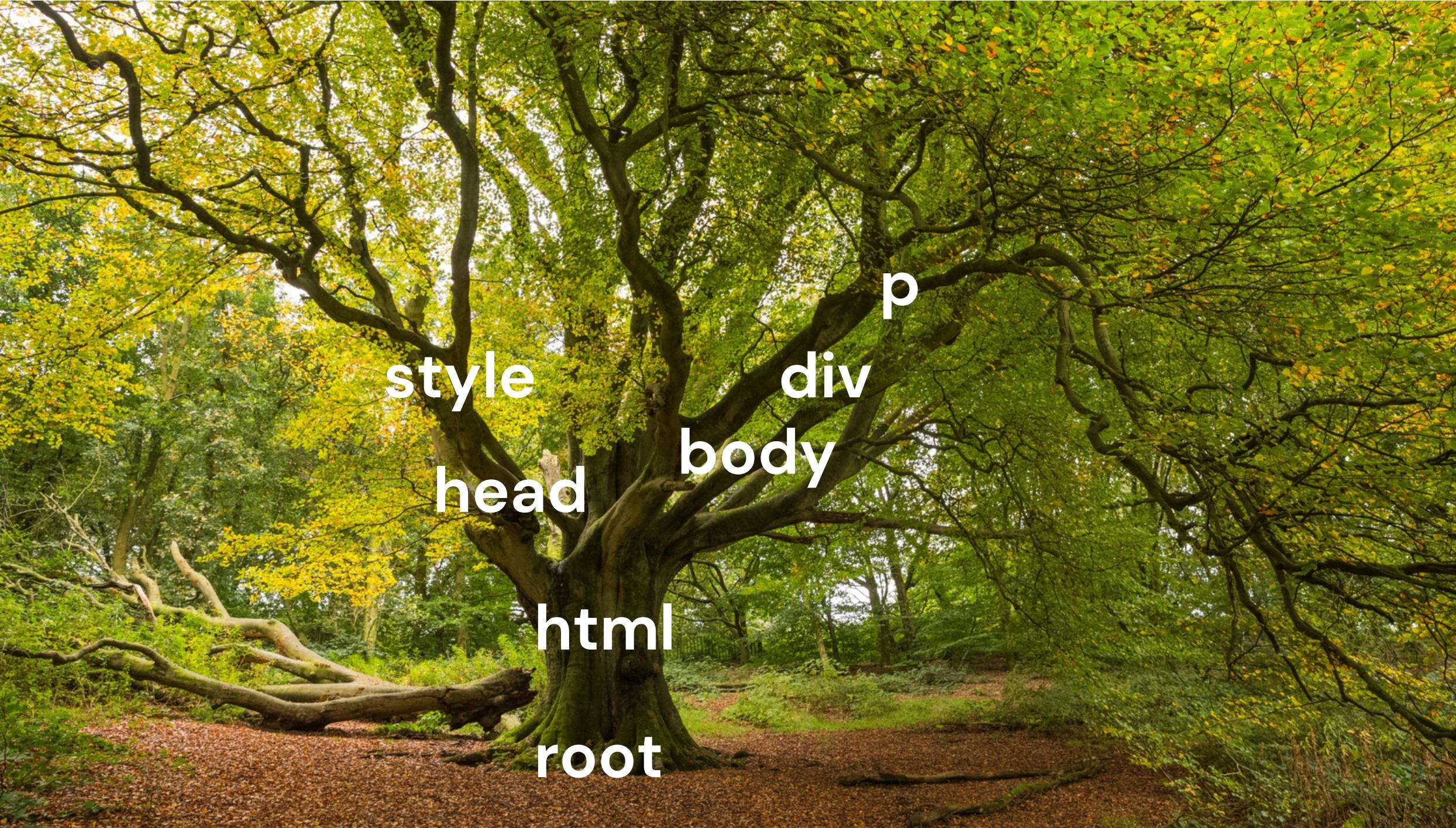


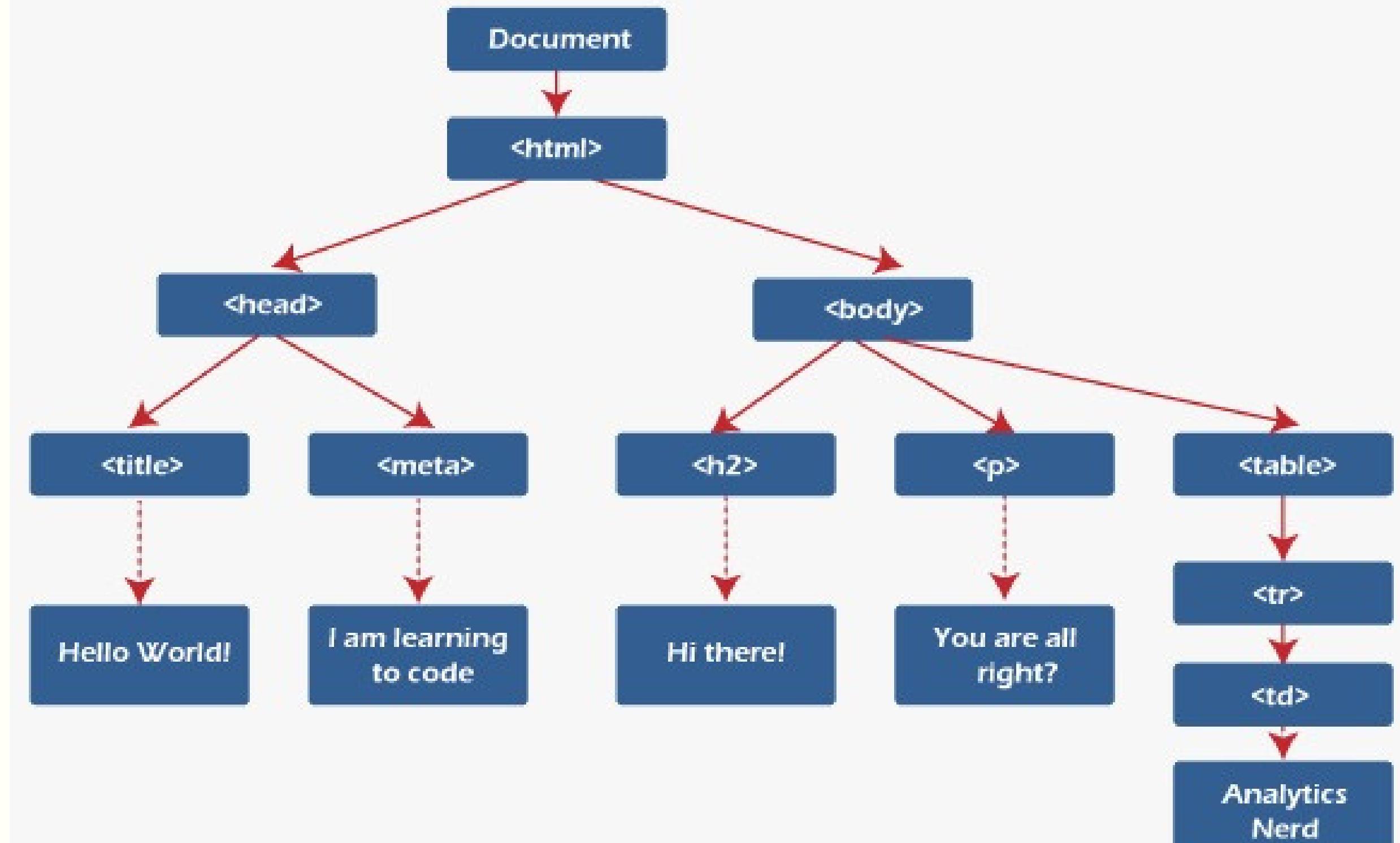
**A representation of html document that we can
interact with**



**it allows js to access it so that our page becomes
dynamic**

DOM



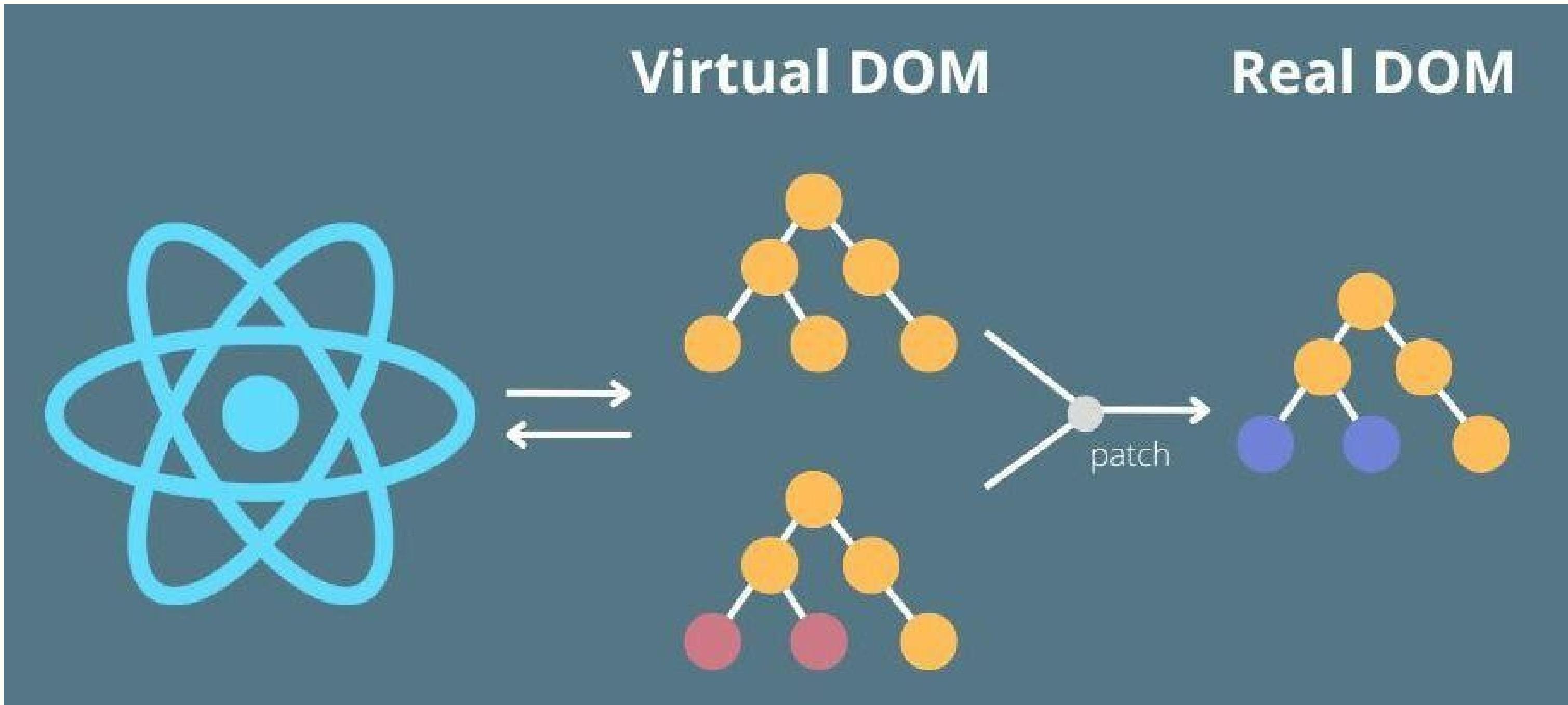


virtual DOM



Virtual DOM

Real DOM

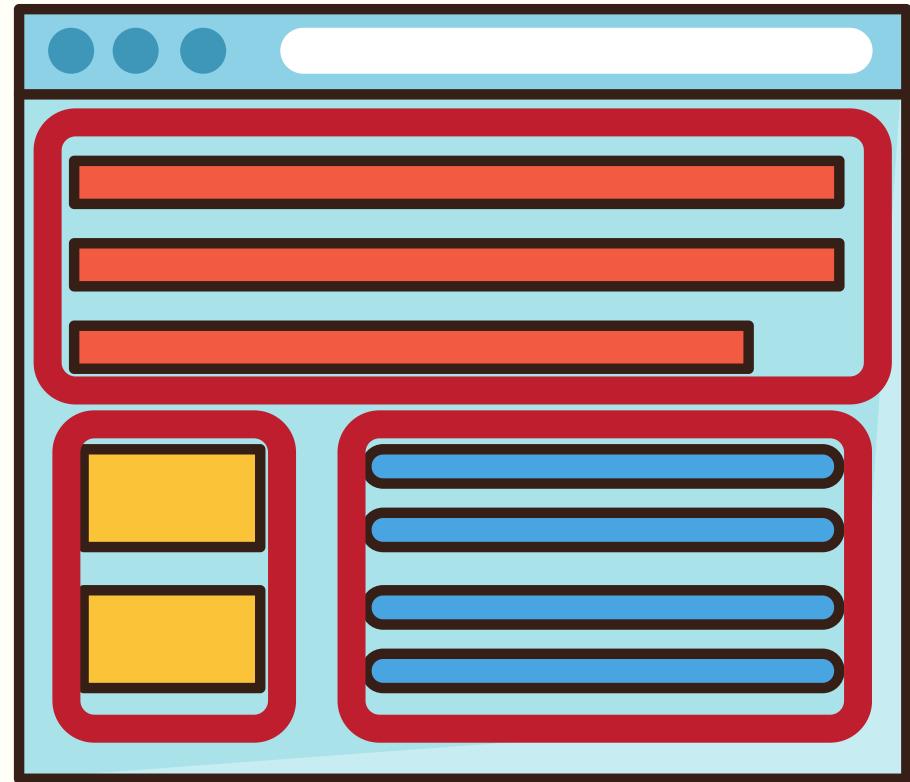




HOW TO THINK IN REACT ?

JS ↗

COMPONENTS



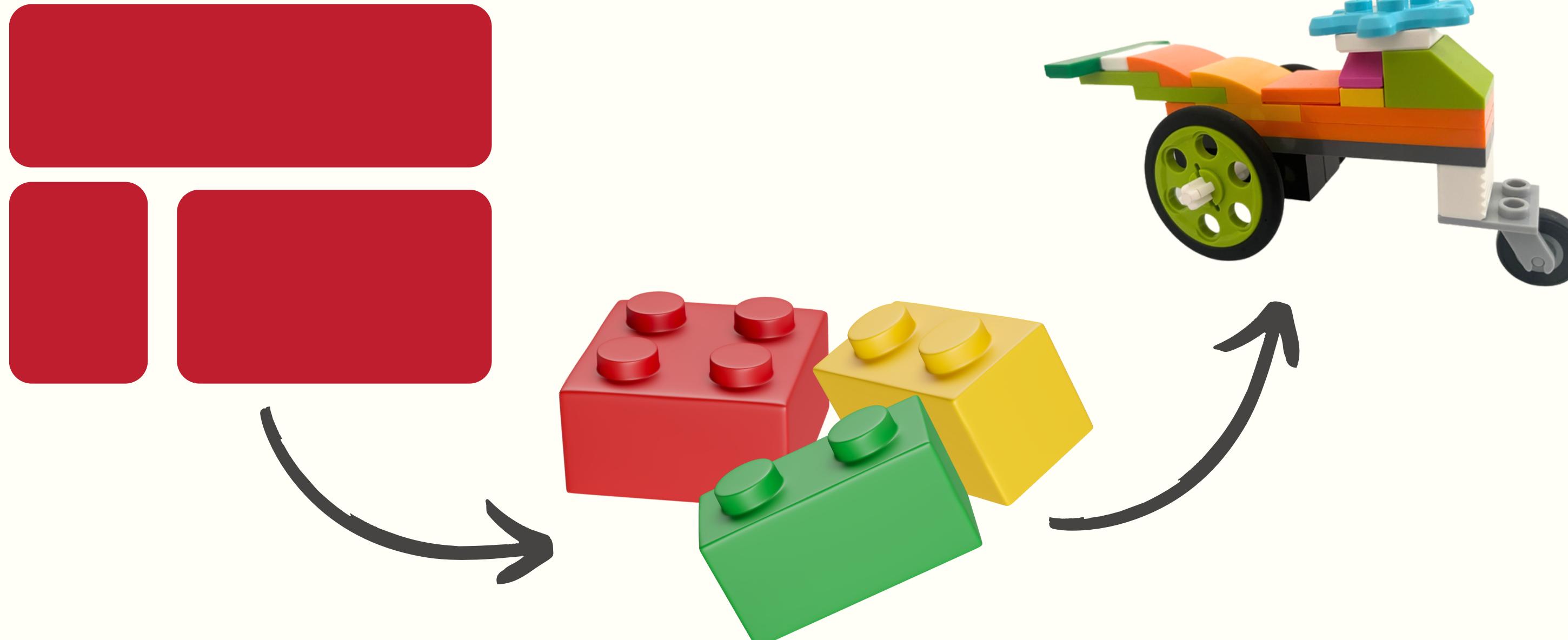
Components



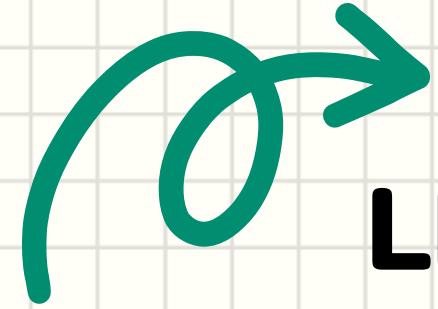
A component is a modular and reusable building block that represents a specific part or section that constructs a web page

COMPONENTS

Components



components help us use reusable modular and better organized code



THIS IS THE FINAL UI OF FACEBOOK

LETS BREAK THAT DOWN TO COMPONENTS

The screenshot displays the Facebook mobile application interface. At the top, there is a dark navigation bar with several icons: a blue 'f' logo, a search icon, a home icon (which is underlined in blue), a people icon, a video icon, a folder icon, a menu icon, a plus sign icon, a message icon with a red notification badge (showing 3), a bell icon, and a profile picture.

The main content area shows a post from a group named 'DZ DÉVELOPPEURS'. The post was made by Abdallah Jsk 13 hours ago. The text of the post is:

salam 3likom
wachnoh al 7al m3a chargily hadi

Below the text is a block of terminal-like log output:

```
$ npm i chargily-epay-js
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/chargily-epay-js - Not found
npm ERR! 404
npm ERR! 404 'chargily-epay-js@*' is not in this registry.
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in: C:\Users\dell\AppData\Local\npm-cache\_logs\2023-12-06T22_54_45_361Z-debug-0.log
```

At the bottom of the post, there are interaction buttons: a like button with 1 like, a comment button with 1 comment, a 'J'aime' button, a 'Commenter' button, and a 'Partager' button.

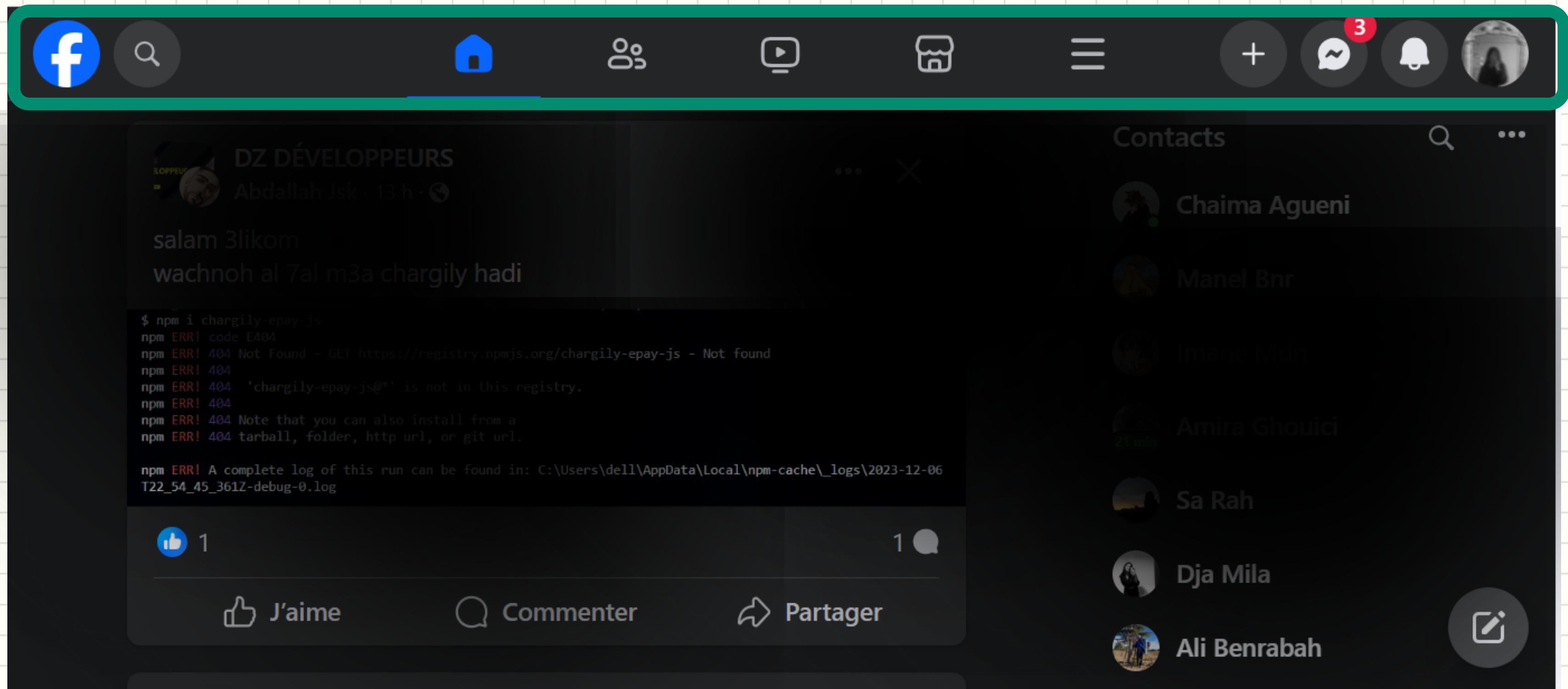
To the right of the post, there is a sidebar titled 'Contacts' with a search icon and three dots. It lists several contacts with their names and profile pictures:

- Chaima Agueni
- Manel Bnr
- Imane Mdn
- Amira Ghouici
- Sa Rah
- Dja Mila
- Ali Benrabah

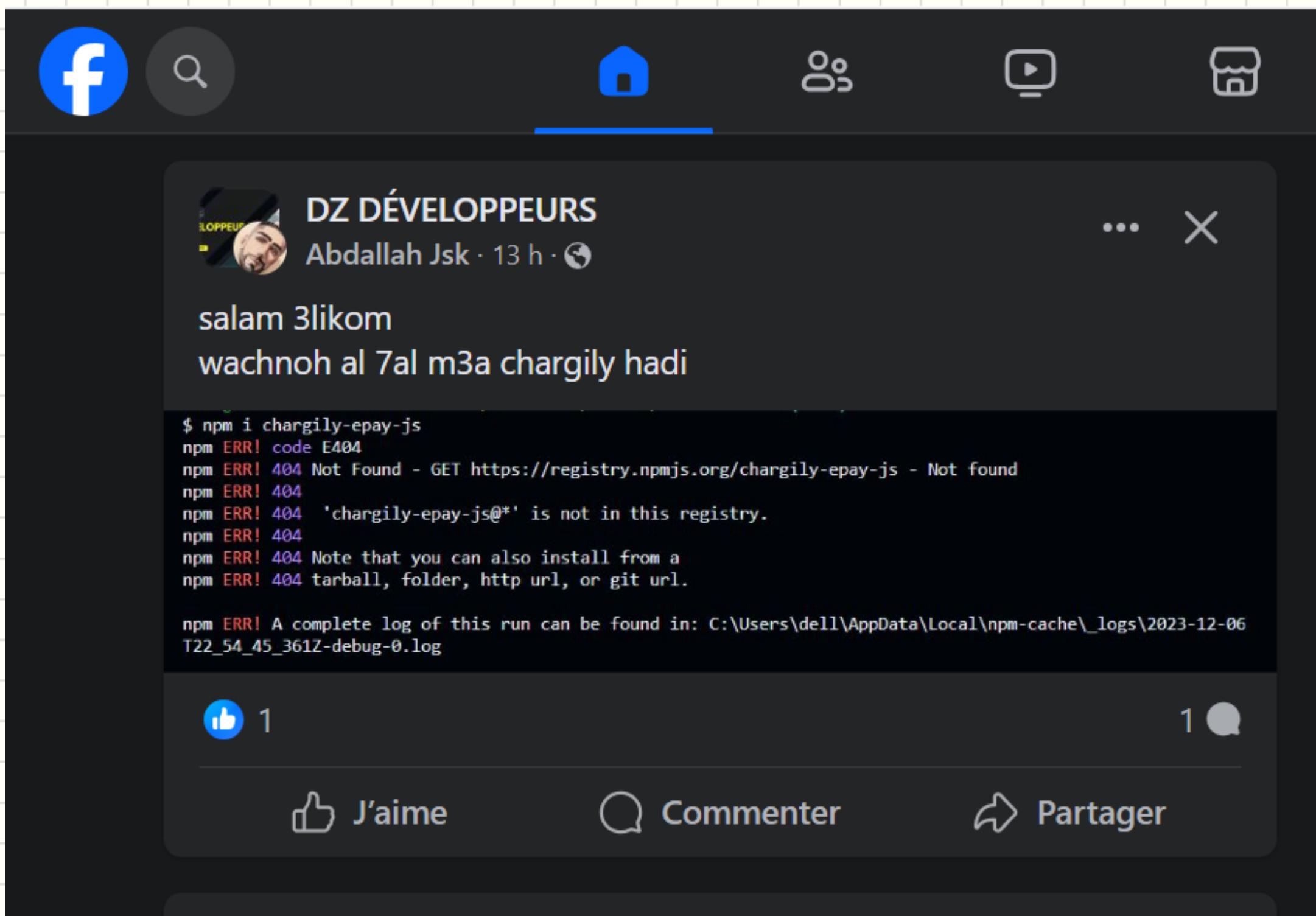
A pink hand-drawn style arrow points from the bottom right corner towards the contact list.

①

NAV BAR



2 CONTACT



A screenshot of a Facebook post from a group named "DZ DÉVELOPPEURS". The post was made by Abdallah Jsk 13 hours ago. The message content is:

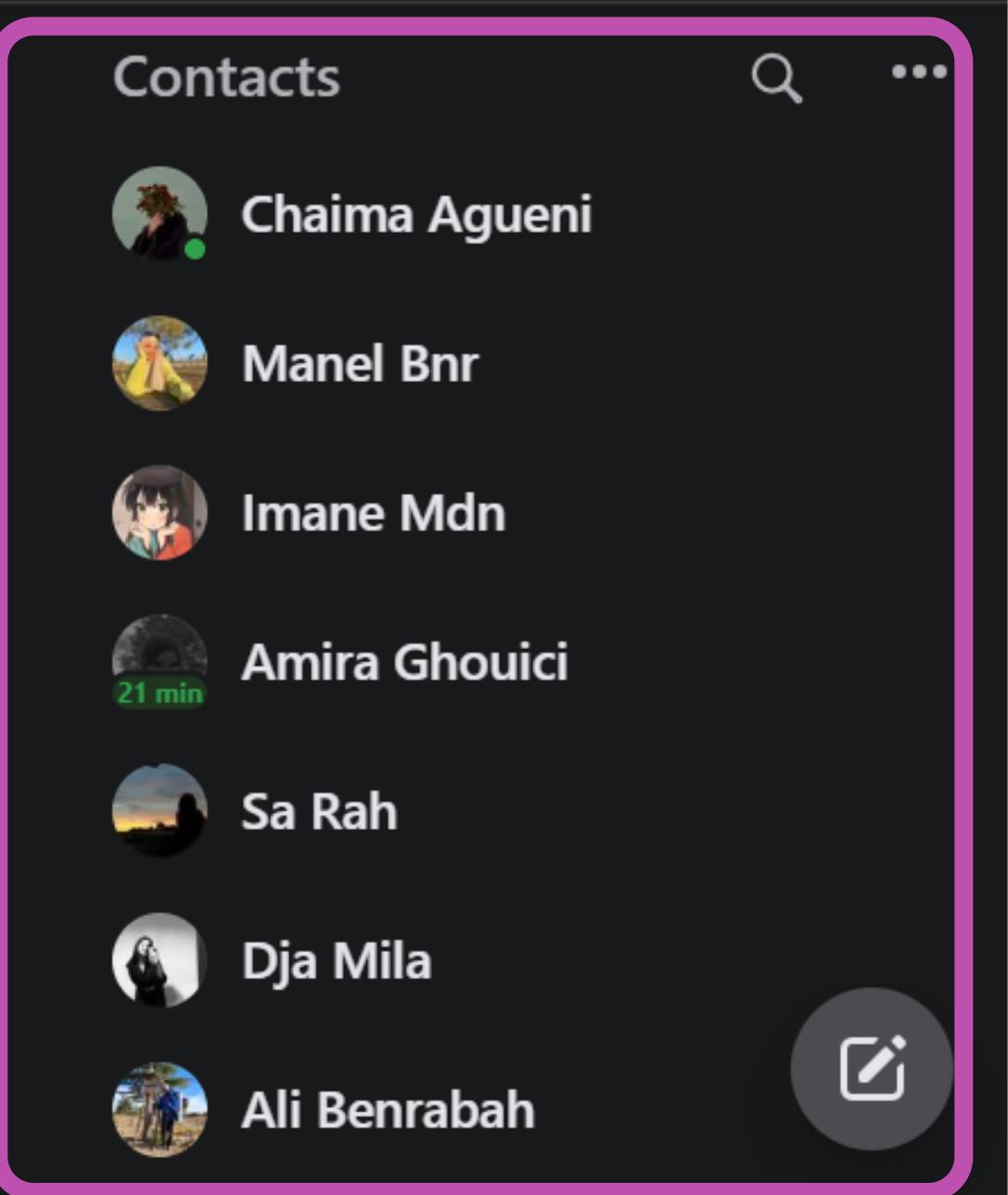
salam 3likom
wachnoh al 7al m3a chargily hadi

The post includes a command-line error log for npm:

```
$ npm i chargily-epay-js
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/chargily-epay-js - Not found
npm ERR! 404
npm ERR! 404 'chargily-epay-js@*' is not in this registry.
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in: C:\Users\dell\AppData\Local\npm-cache\_logs\2023-12-06T22_54_45_361Z-debug-0.log
```

Below the post are interaction buttons: 1 like, 1 comment, and sharing options.



A contact list titled "Contacts" with 8 entries:

- Chaima Agueni
- Manel Bnr
- Imane Mdn
- Amira Ghouici
- Sa Rah
- Dja Mila
- Ali Benrabah

A pink rounded rectangle highlights the contact list.

③ POST COMPONENT

The image shows a dark-themed user interface for a social media platform, similar to Facebook. At the top, there is a navigation bar with icons for the logo, search, home, groups, video, and more. Below the navigation bar, a post from a group named "DZ DÉVELOPPEURS" is displayed. The post was made by Abdallah Jsk 13 hours ago. The content of the post is a message in Arabic: "salam 3likom" and "wachnoh al 7al m3a chargily hadi". Below the message is a block of terminal output showing an npm error related to the "chargily-epay-js" package. The post has 1 like and 1 comment. At the bottom of the post are buttons for "J'aime", "Commenter", and "Partager". To the right of the post, a sidebar titled "Contacts" lists several users with their profile pictures and names: Chaima Agueni, Manel Bnr, Imane Mdn, Amira Ghouici, Sa Rah, Dja Mila, and Ali Benrabah. A red rounded rectangle highlights the main post area.

DZ DÉVELOPPEURS

Abdallah Jsk · 13 h

salam 3likom

wachnoh al 7al m3a chargily hadi

```
$ npm i chargily-epay-js
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/chargily-epay-js - Not found
npm ERR! 404
npm ERR! 404 'chargily-epay-js@*' is not in this registry.
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in: C:\Users\dell\AppData\Local\npm-cache\_logs\2023-12-06T22_54_45_361Z-debug-0.log
```

1

1

J'aime

Commenter

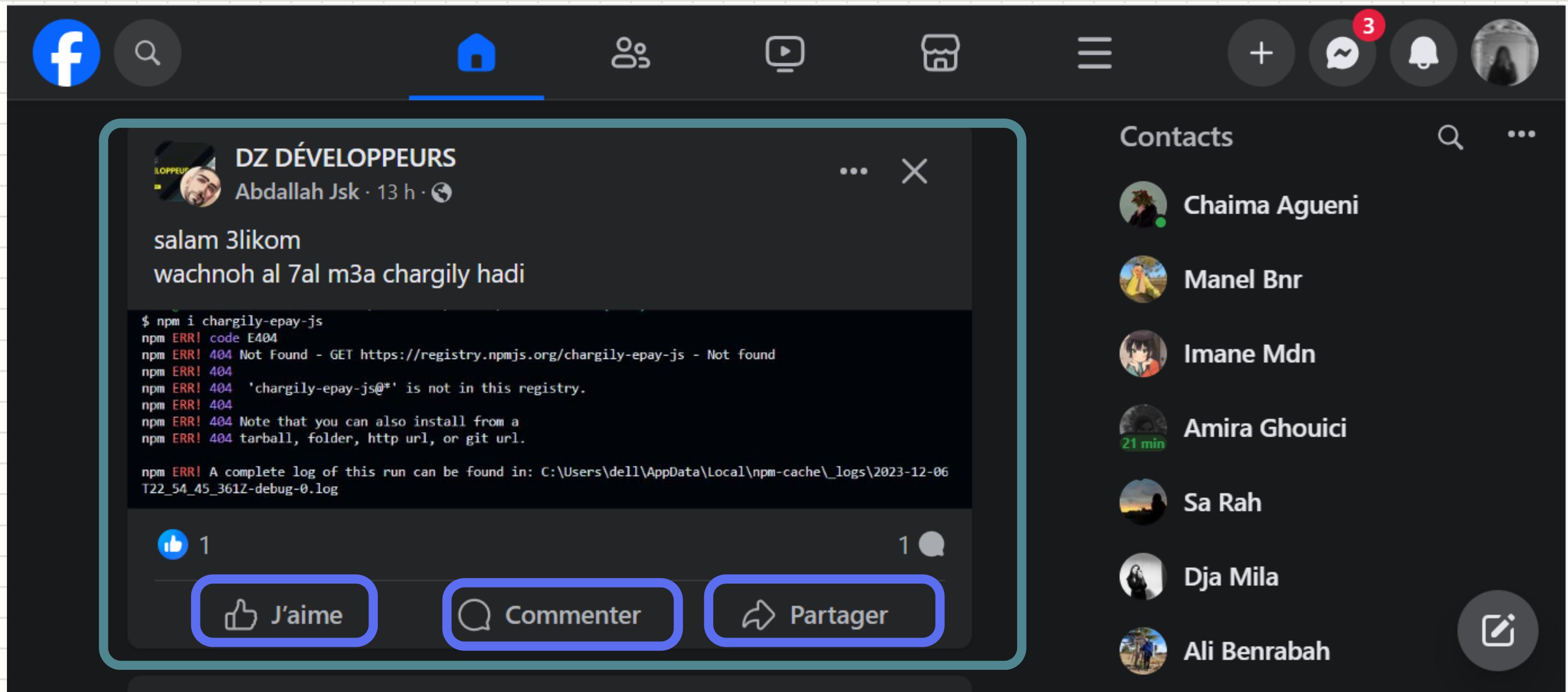
Partager

Contacts

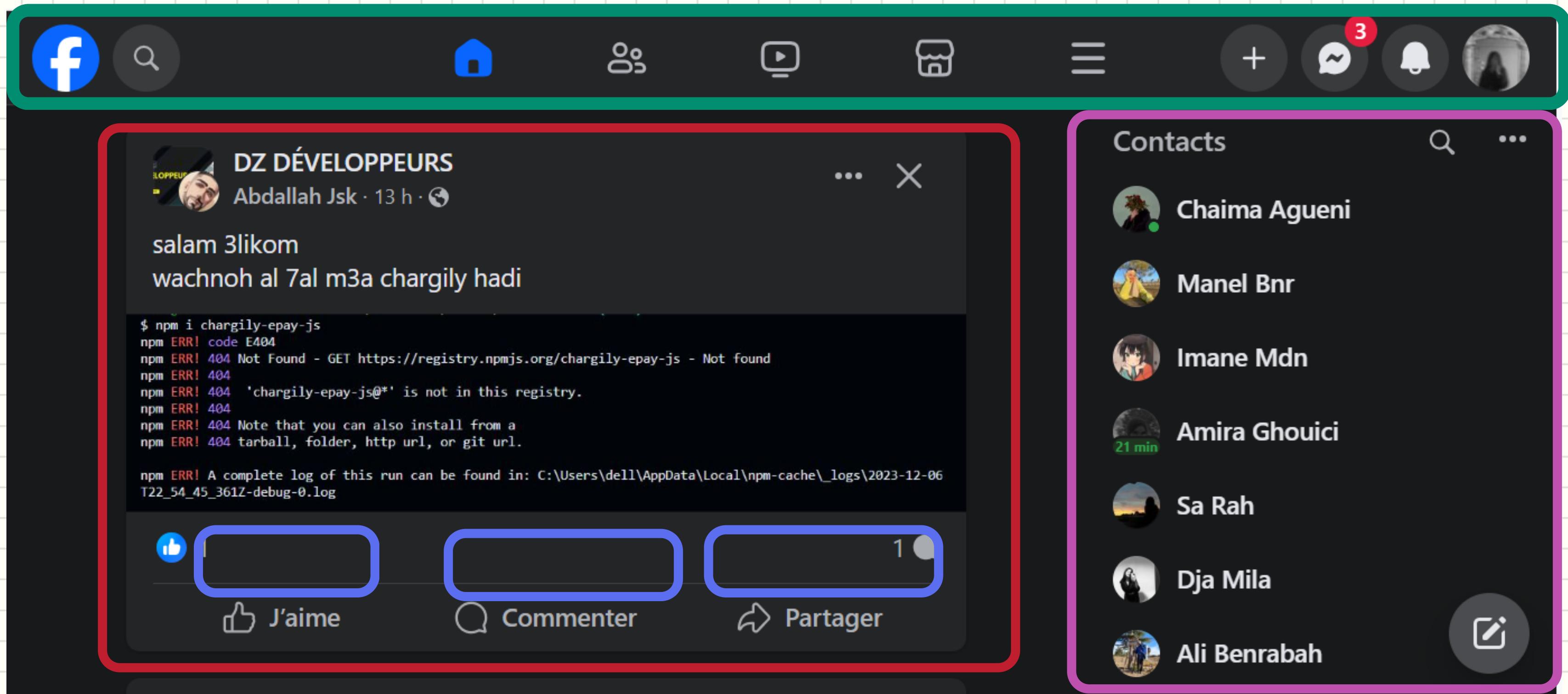
- Chaima Agueni
- Manel Bnr
- Imane Mdn
- Amira Ghouici
- Sa Rah
- Dja Mila
- Ali Benrabah

③ POST COMPONENT

- ④ LIKE
- ⑤ COMMENT
- ⑥ SHARE



we can devide each section of my ui into components then build each component individually and combine them to build this fb app







APP

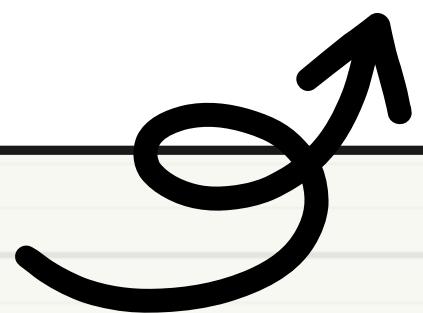
Nav bar

Contact

Post

Like

This way of organizing things makes React
powerful and easy to work with



APP

Nav bar

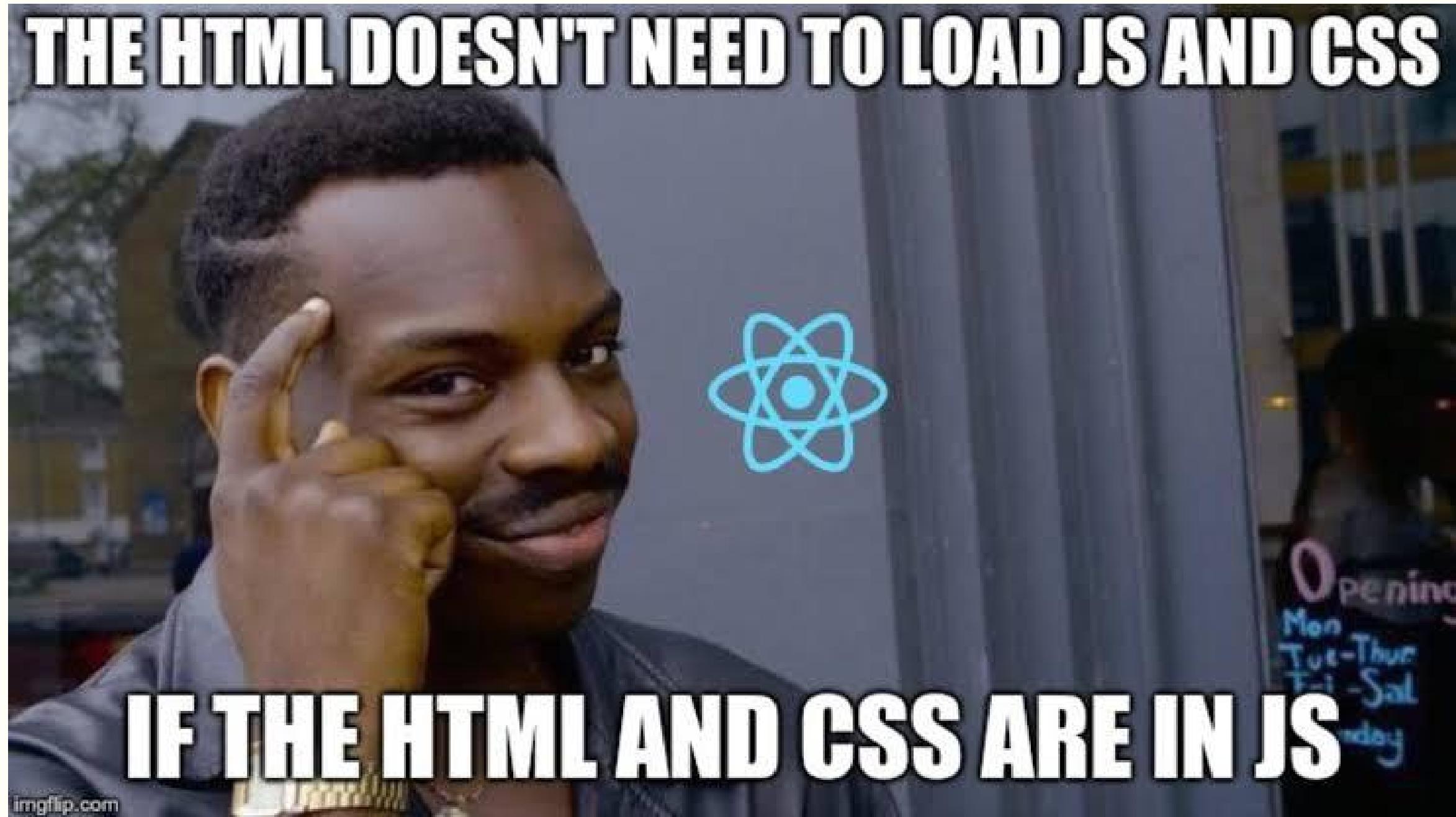
Contact

Post

Like

**This way of organizing things makes React
powerful and easy to work with**

JSX



React



JSX

- stands for JavaScript XML.
- allows us to write HTML in React.
- makes it easier to write and add HTML in React.
- converts HTML tags into react elements.

EXPRESSIONS IN JSX

With JSX you can write JS expressions inside curly braces { }

ONE TOP LEVEL ELEMENT

- The HTML code must be wrapped in ONE top level element
- To write HTML on multiple lines, put the HTML inside parentheses:
- you can use a "fragment" to wrap multiple lines



```
const myElement = (  
  <>  
    <p>I am a paragraph.</p>  
    <p>I am a paragraph too.</p>  
  </>  
)
```

ATTRIBUTE CLASS = CLASSNAME

```
const myElement = <h1 className="myclass">Hello World</h1>;
```

HOOKS

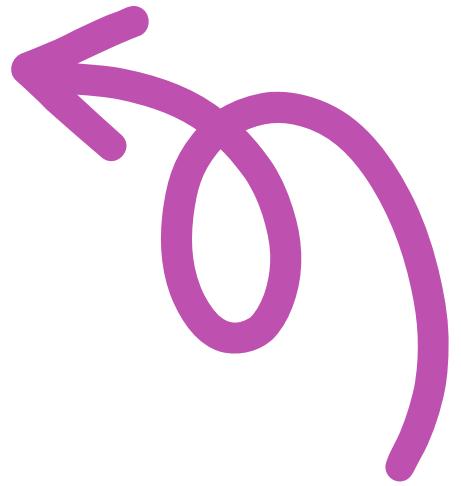
Hooks allow us to "hook" into React features such as state and lifecycle methods.

Example



HOOKS RULES

- Hooks can only be called inside React function components.
- Hooks can only be called at the top level of a component.
- Hooks cannot be conditional



USESTATE HOOK

allows us to track state in a function component

State: data or properties that need to be tracking in an application.

Example



USESTATE

```
import { useState } from "react";

export default function FavoriteColor() {
  const [color, setColor] = useState("red");

  return (
    <>
      <h1>My favorite event is {color}!</h1>
      <button
        type="button"
        onClick={() => setColor("blue")}
      >Blue</button>
    </>
  )
}
```

①

Importing useState

②

Initializing the state

③

Reading the state

④

Updating the state



UPDATING STATES

When state is updated,
the entire state gets overwritten.

We use the spread operator then

```
export default function Car() {
  const [car, setCar] = useState({
    brand: "Ford",
    model: "Mustang",
    year: "1964",
    color: "red"
  });

  const updateColor = () => {
    setCar(previousState => {
      return { ...previousState, color: "blue" }
    });
  }

  return (
    <>
      <h1>My {car.brand}</h1>
      <p>It is a {car.color} {car.model} from {car.year}.</p>
      <button
        type="button"
        onClick={updateColor}
        >Blue</button>
    </>
  )
}
```



USEEFFECT

`useEffect(setup, dependencies?)`

- **useEffect is a React Hook that lets you synchronize a component with an external system.**
- If you're not trying to synchronize with some external system, you probably don't need an Effect.



USEEFFECT

```
import { useEffect } from 'react';
import { createConnection } from './chat.js';

function ChatRoom({ roomId }) {
  const [serverUrl, setServerUrl] = useState('https://localhost:1234');

  useEffect(() => {
    const connection = createConnection(serverUrl, roomId);
    connection.connect();
    return () => {
      connection.disconnect();
    };
  }, [serverUrl, roomId]);
  // ...
}
```

<< IMPORTING USEEFFECT

<< SETUP CODE

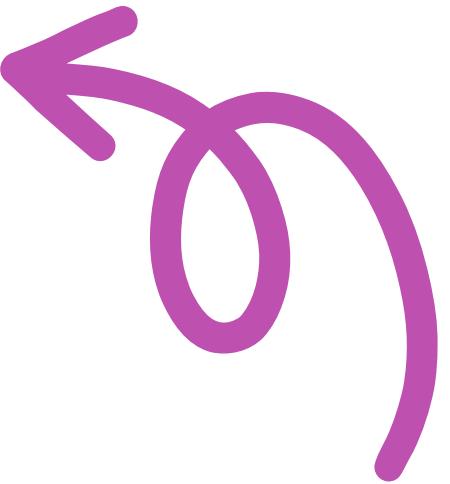
<< CLEANUP FUNCTION (OPTIONAL)

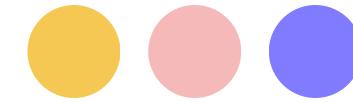
<< LIST OF DEPENDENCIES



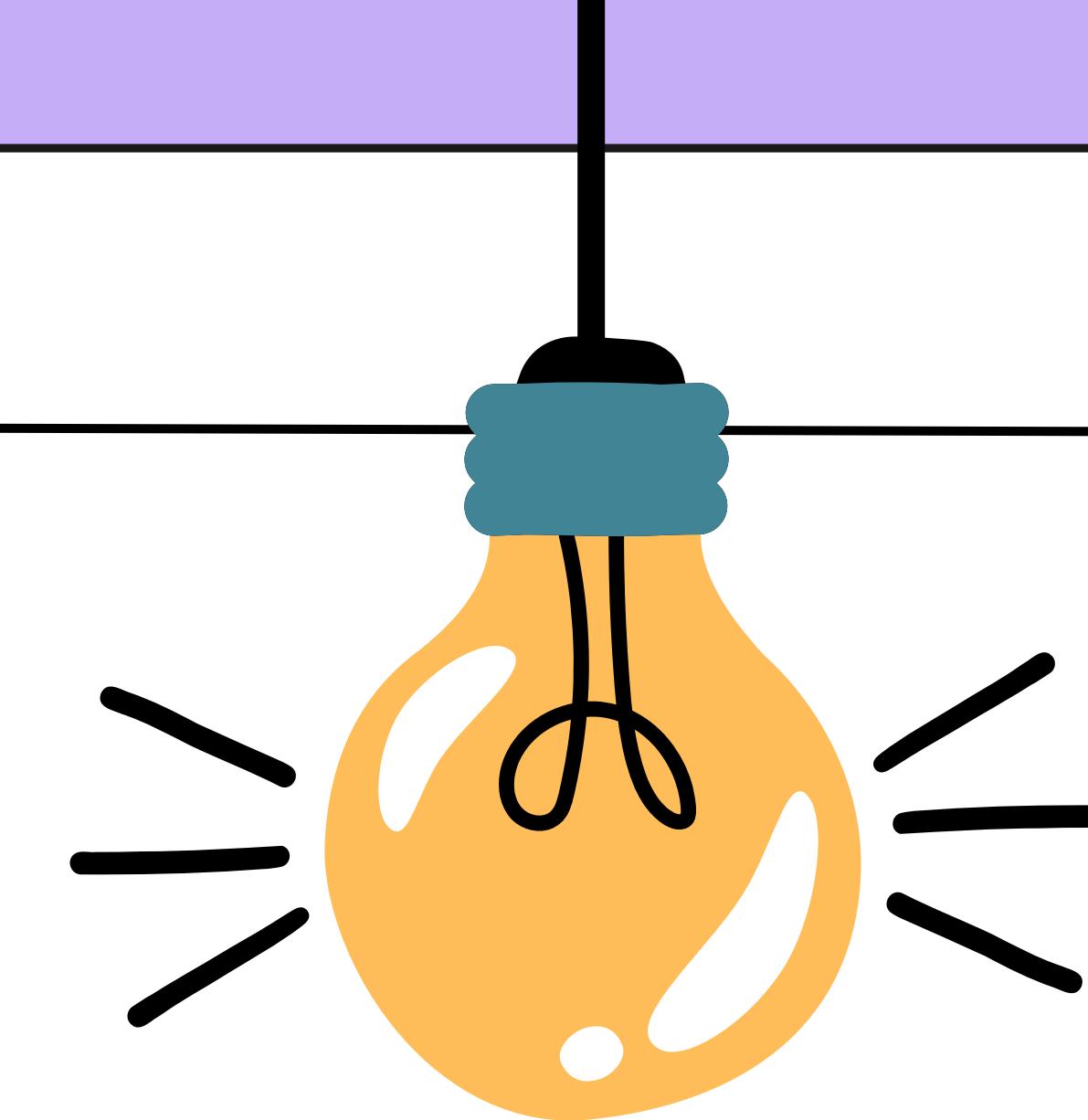
HOOKS RULES

- Hooks can only be called inside React function components.
- Hooks can only be called at the top level of a component.
- Hooks cannot be conditional

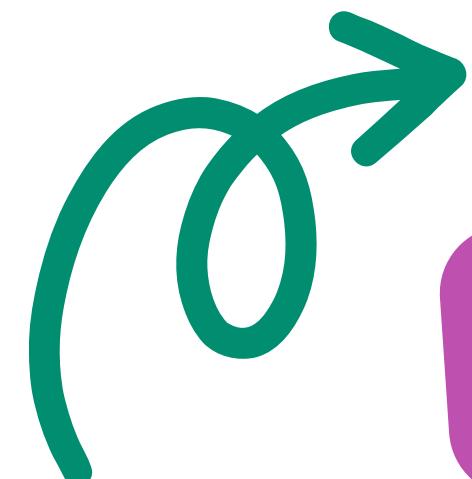




THANK YOU



For your attention



NOW LETS CODE !!!

BUT LETS EAT FIRST