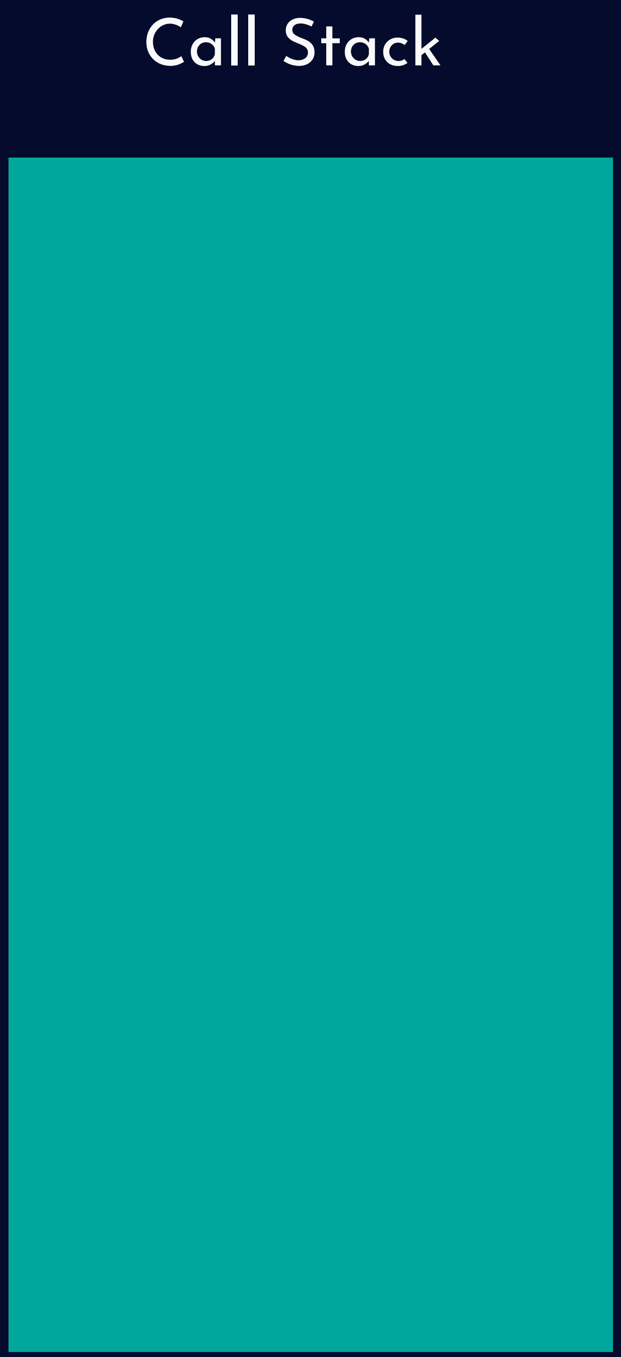
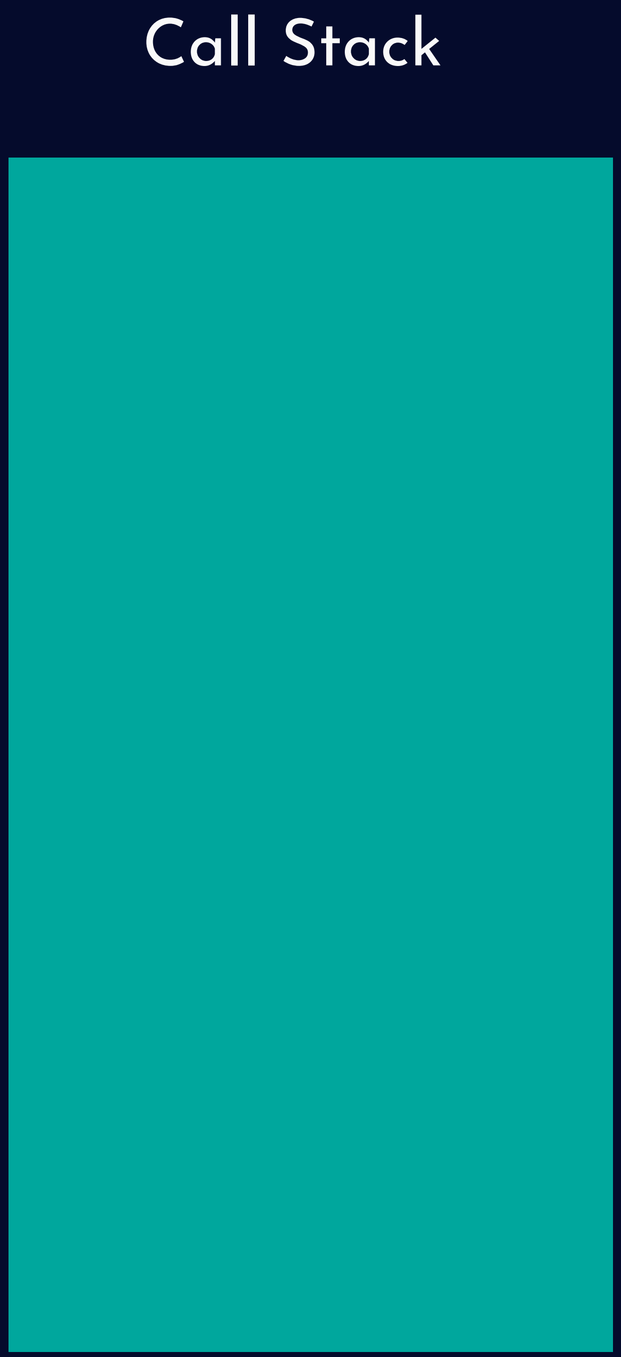
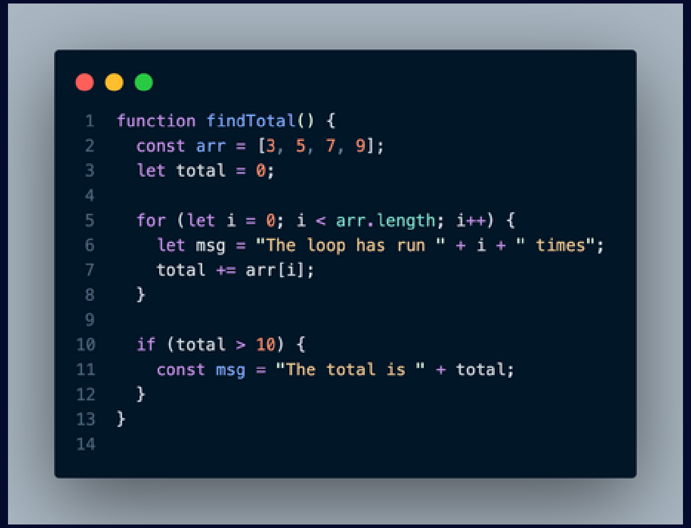
JS Callstack



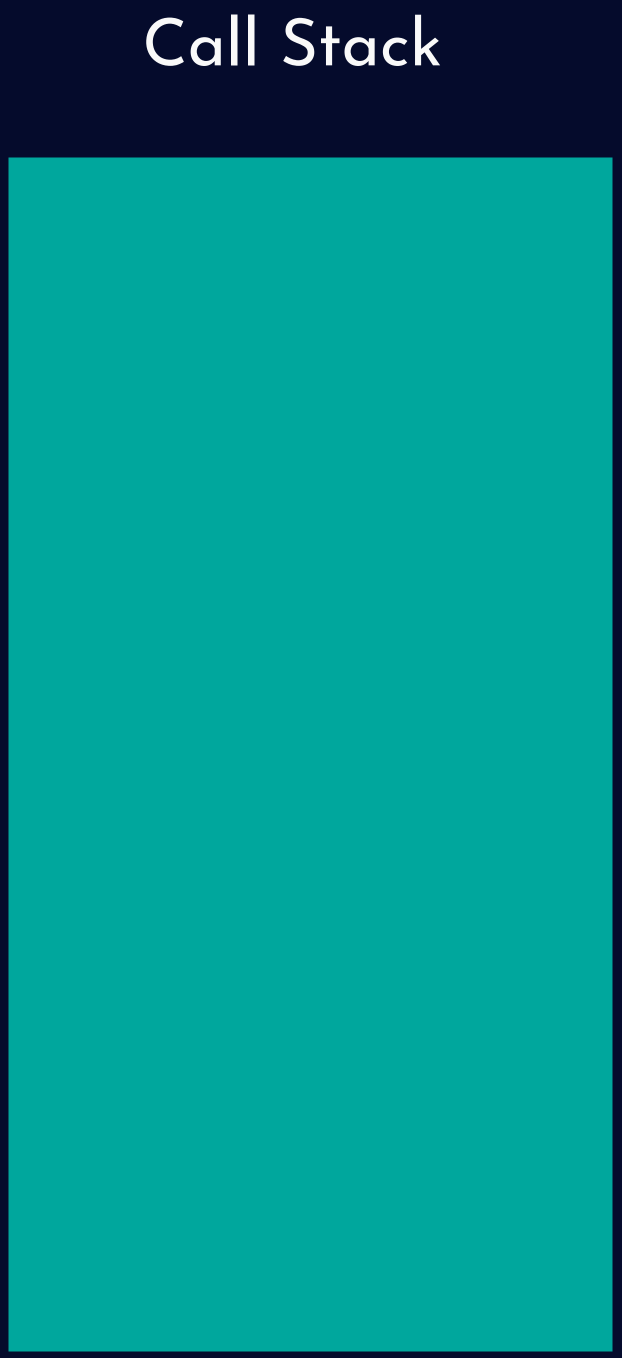
JS Callstack

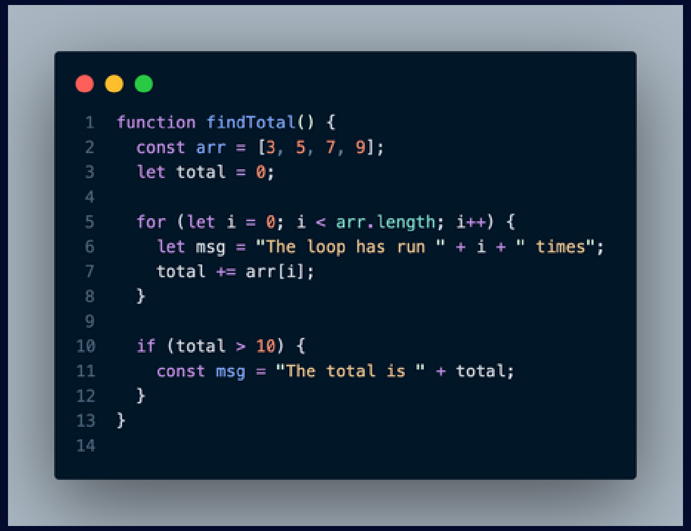




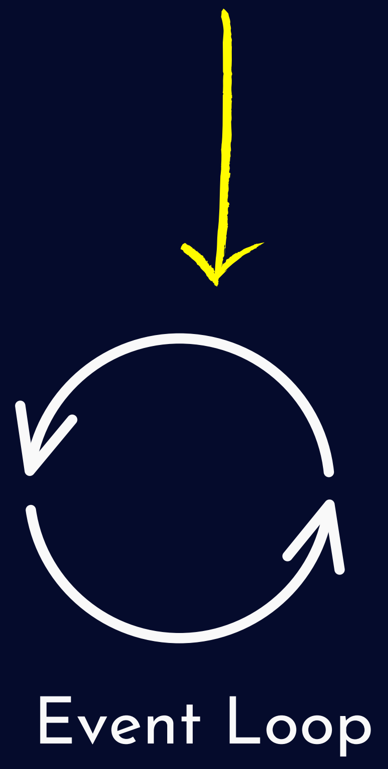
```
function findTotal() {
  const arr = [3, 5, 7, 9];
  let total = 0;

  for (let i = 0; i < arr.length; i++) {
    let msg = "The loop has run " + i + " times";
    total += arr[i];
  }

  if (total > 10) {
    const msg = "The total is " + total;
  }
}
```

JS Callstack

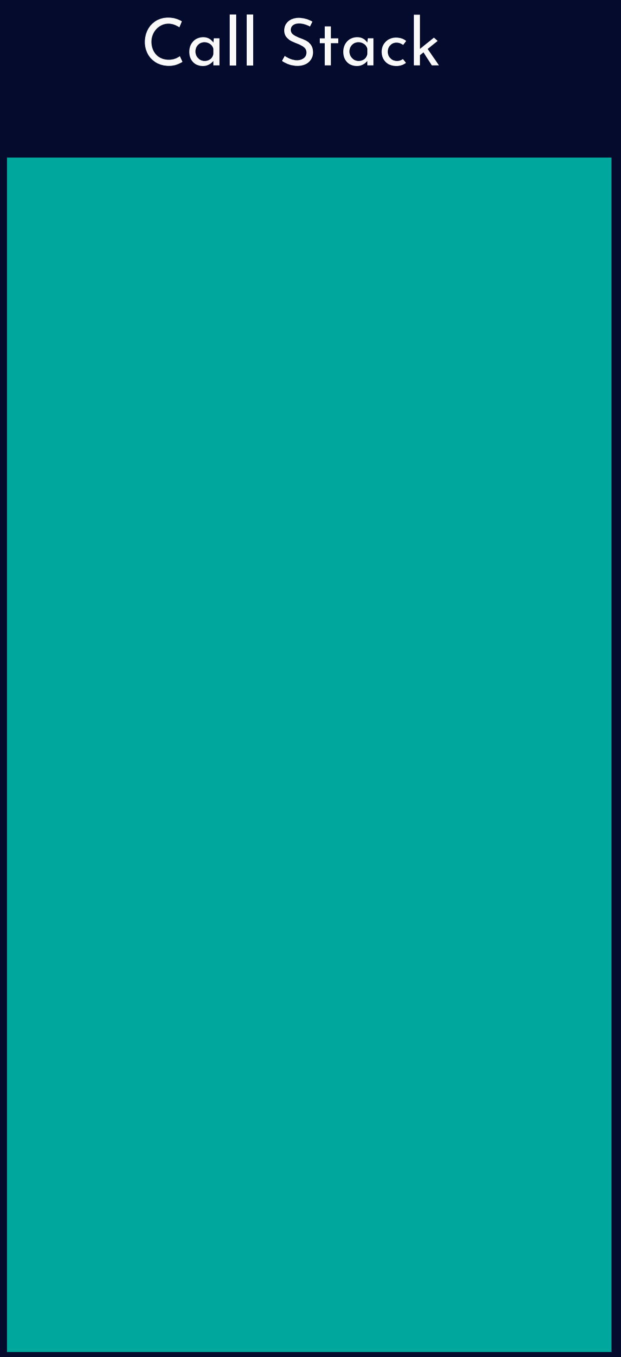


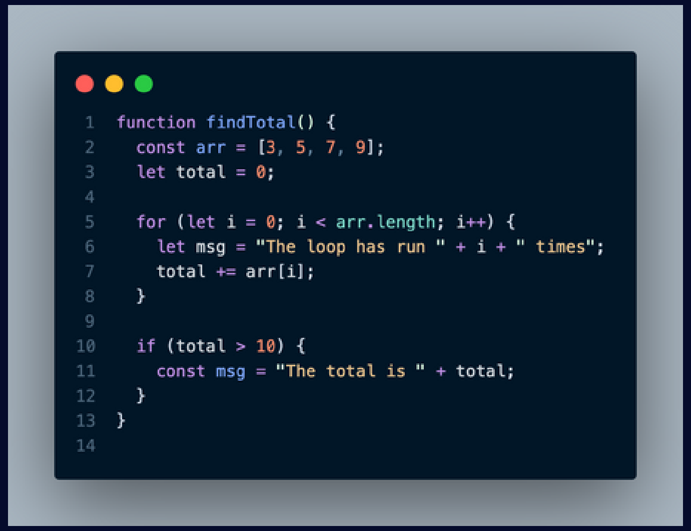


```javascript
function findTotal() {
  const arr = [3, 5, 7, 9];
  let total = 0;

  for (let i = 0; i < arr.length; i++) {
    let msg = "The loop has run " + i + " times";
    total += arr[i];
  }

  if (total > 10) {
    const msg = "The total is " + total;
  }
}
```

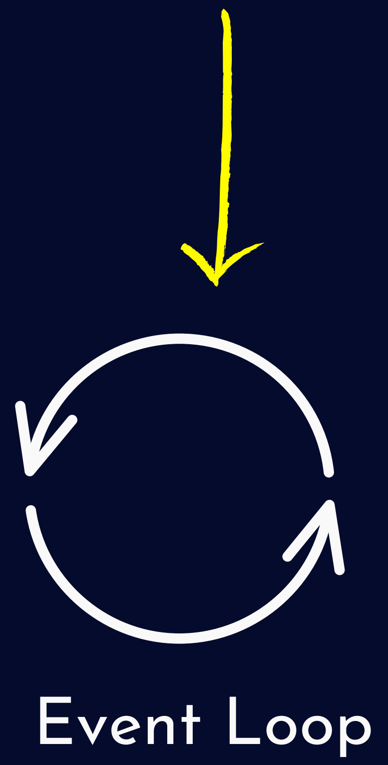


JS Callstack



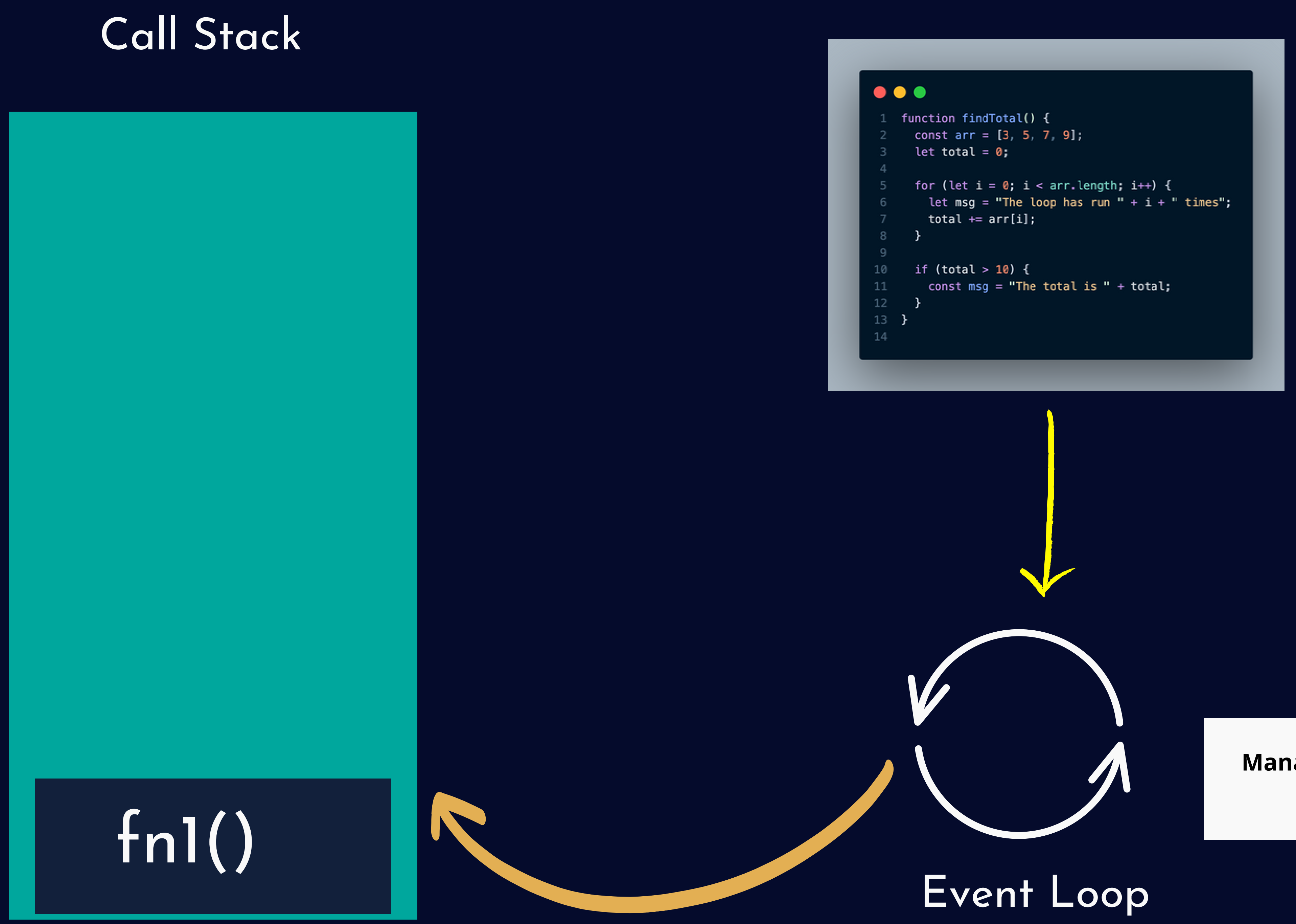


```
function findTotal() {
  const arr = [3, 5, 7, 9];
  let total = 0;

  for (let i = 0; i < arr.length; i++) {
    let msg = "The loop has run " + i + " times";
    total += arr[i];
  }

  if (total > 10) {
    const msg = "The total is " + total;
  }
}

```



Manages which callback function to be put inside callstack

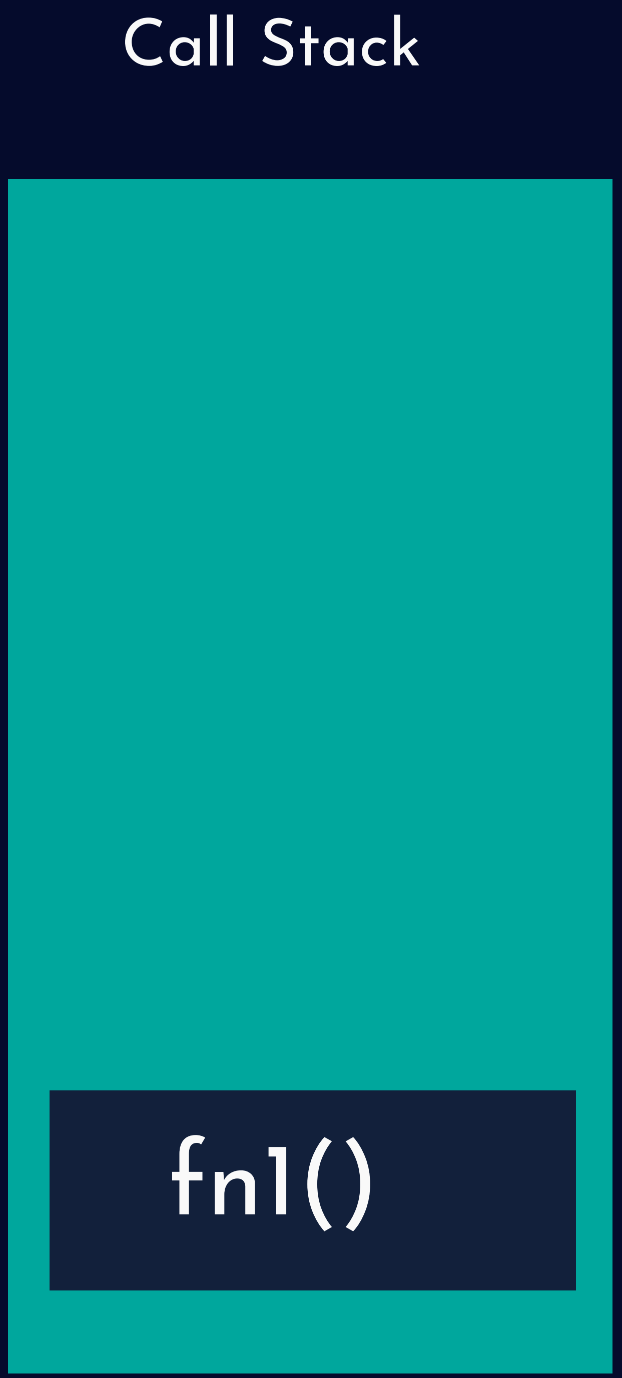
JS Callstack



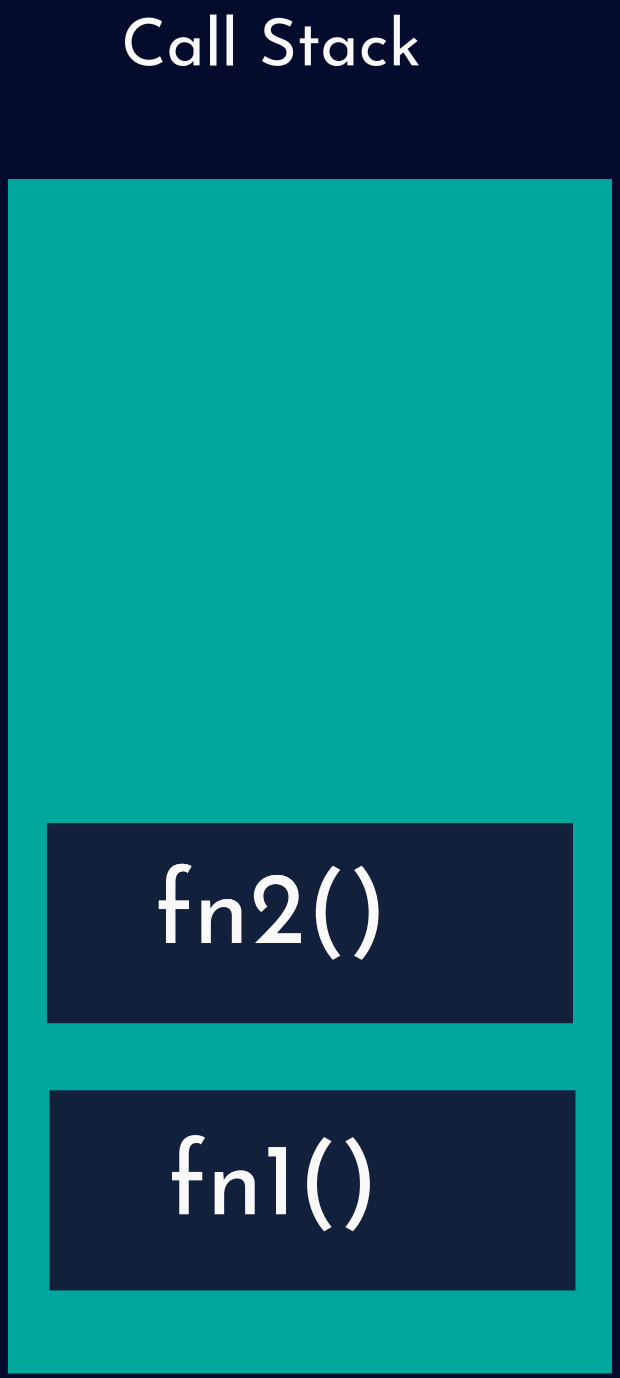
```
function findTotal() {
  const arr = [3, 5, 7, 9];
  let total = 0;

  for (let i = 0; i < arr.length; i++) {
    let msg = "The loop has run " + i + " times";
    total += arr[i];
  }

  if (total > 10) {
    const msg = "The total is " + total;
  }
}

```

Manages which callback function to be put inside callstack

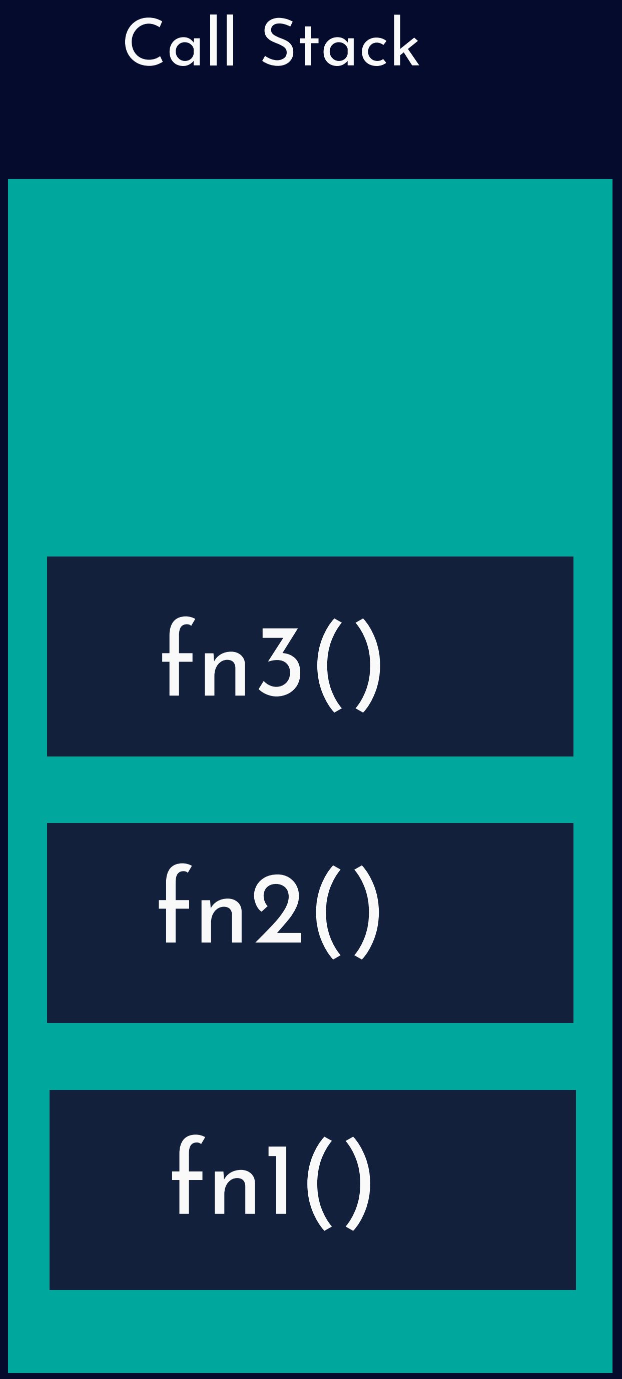
JS Callstack



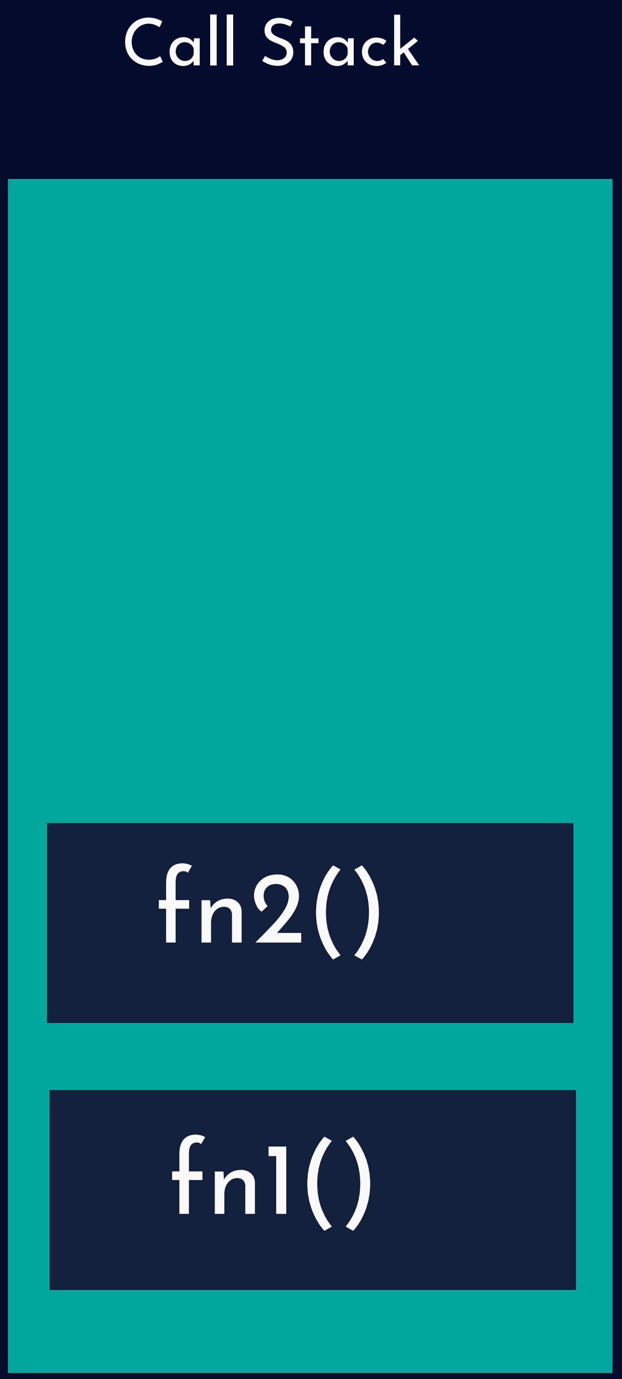
JS Callstack



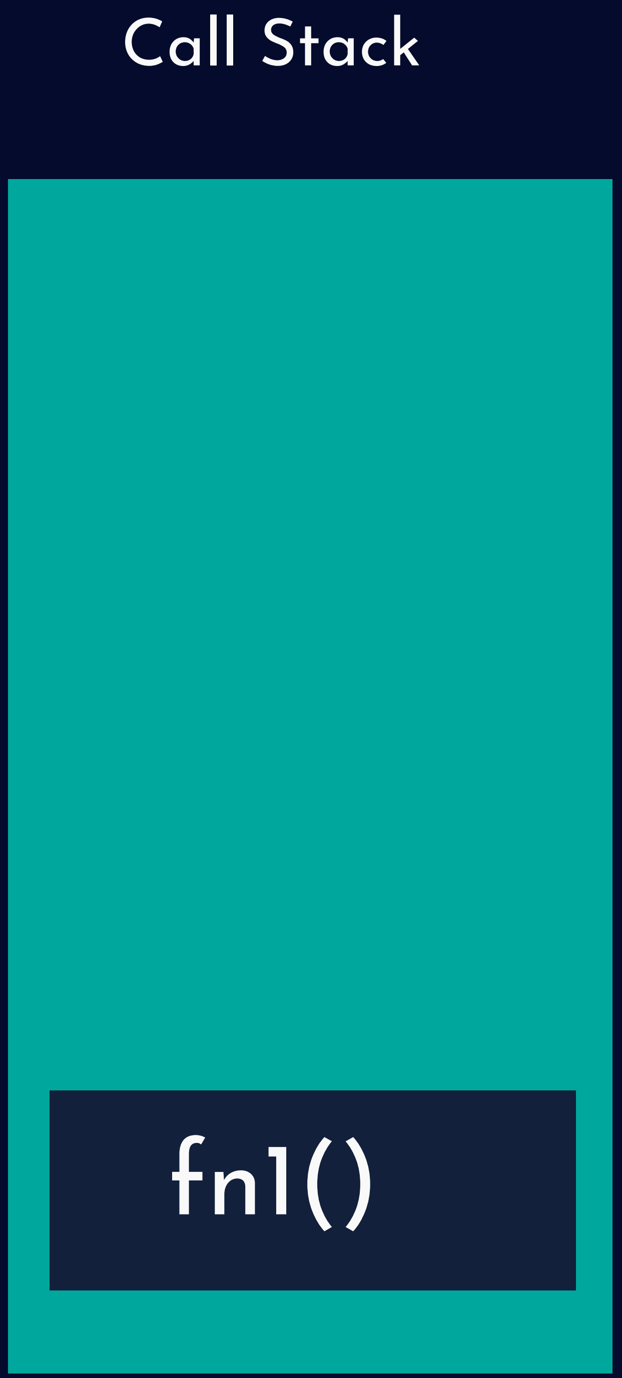
JS Callstack



JS Callstack

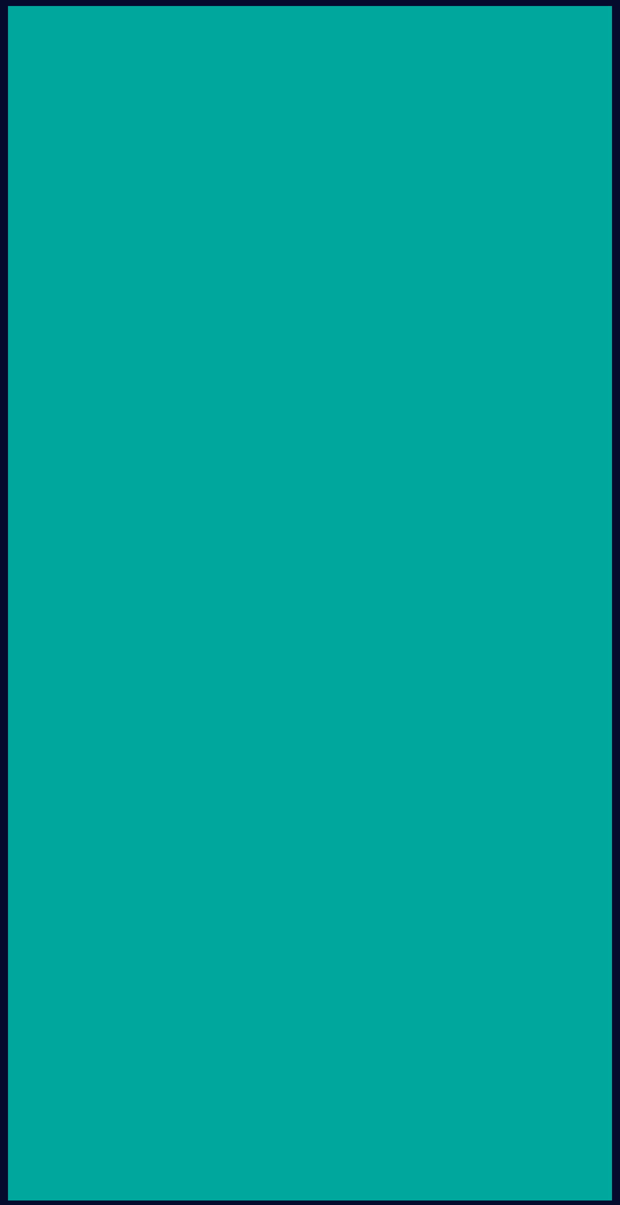


JS Callstack

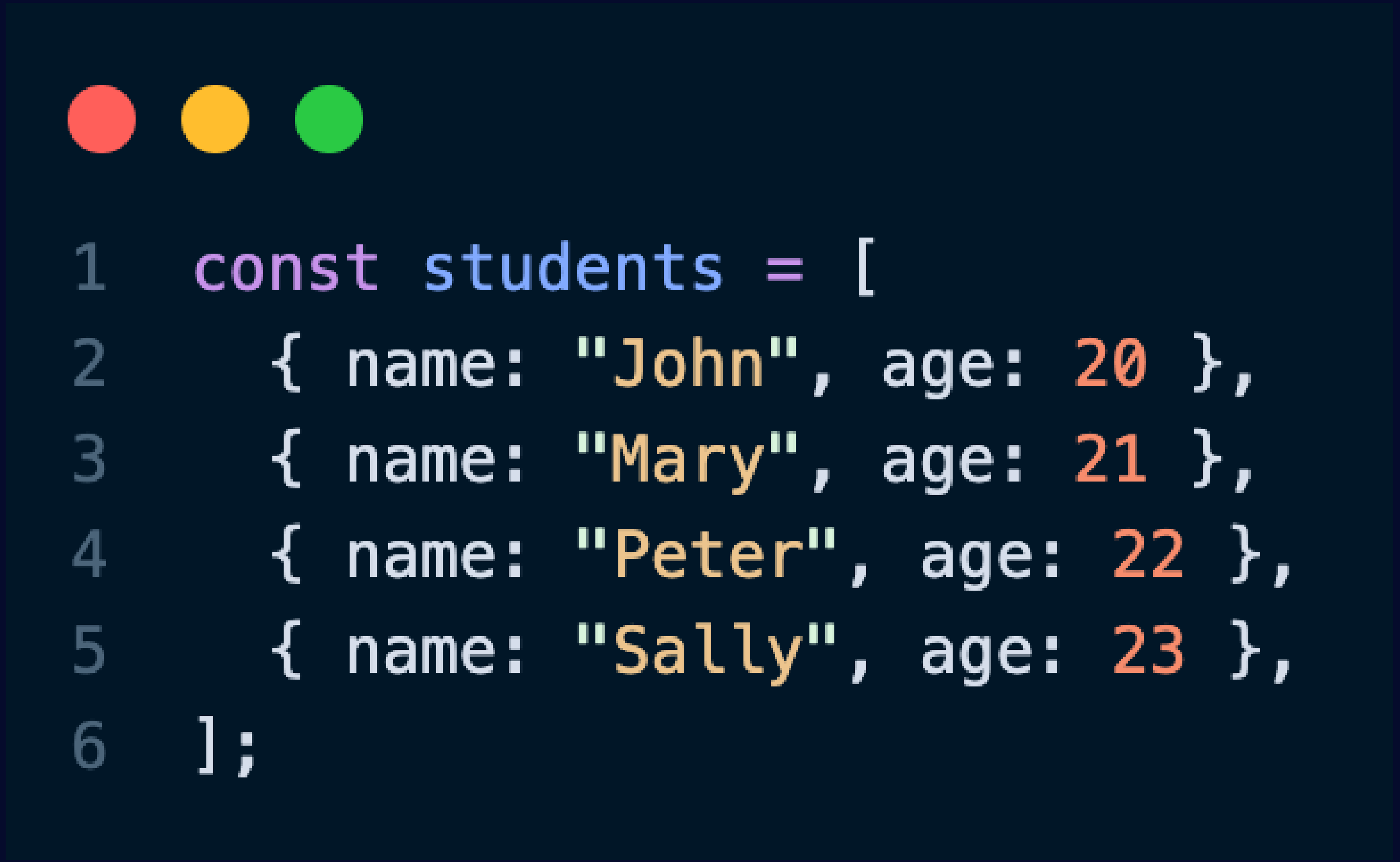


JS Callstack

Call Stack



JS Callstack



```
const students = [
  { name: "John", age: 20 },
  { name: "Mary", age: 21 },
  { name: "Peter", age: 22 },
  { name: "Sally", age: 23 },
];
```

Array Of Objects

JS

Data Mutation

Data Mutation

Data mutation is the process of changing the value of an existing data structure

JS

Data Mutation

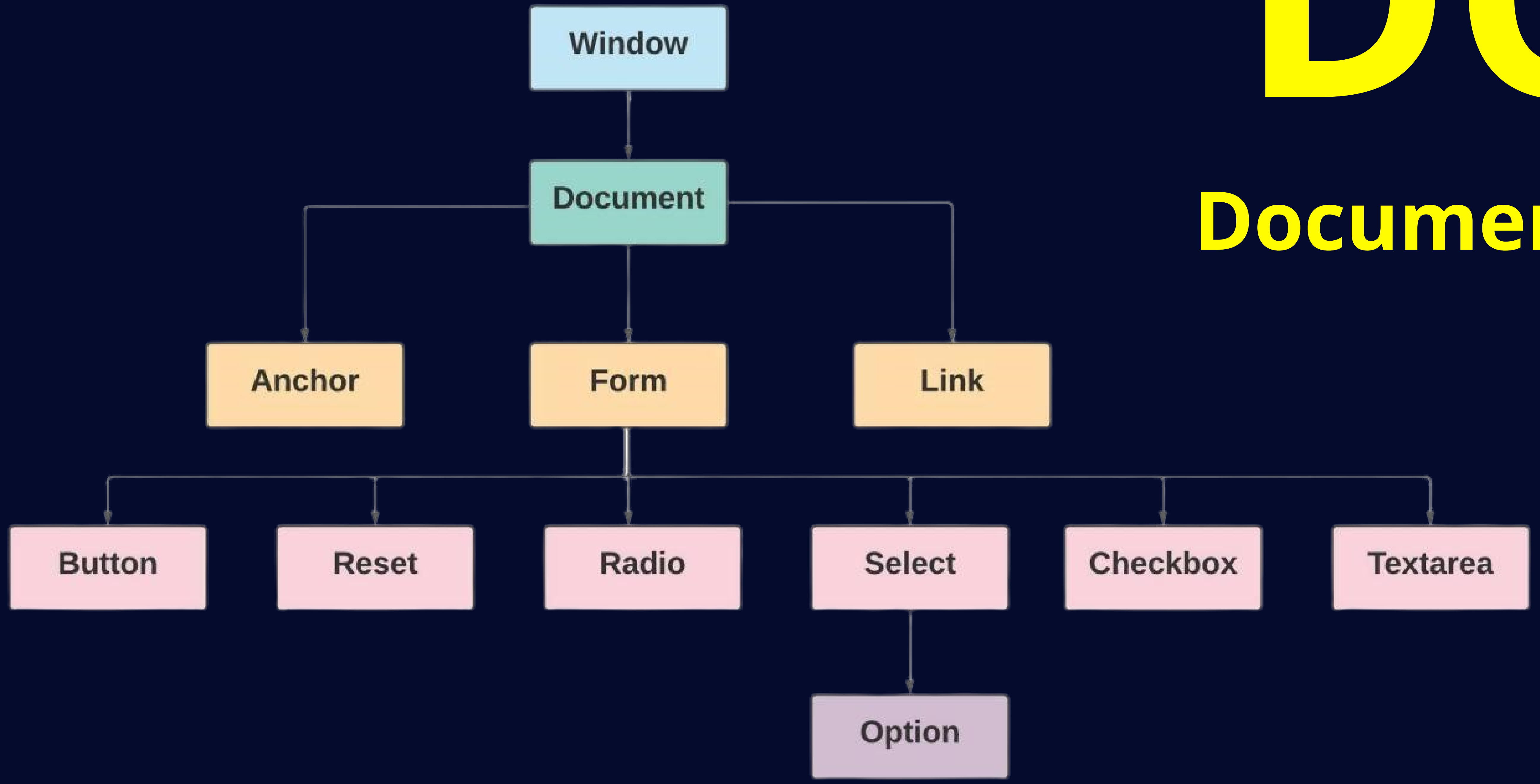
Avoid data mutation as much as possible.

Data Mutation

Avoid data mutation as much as possible.

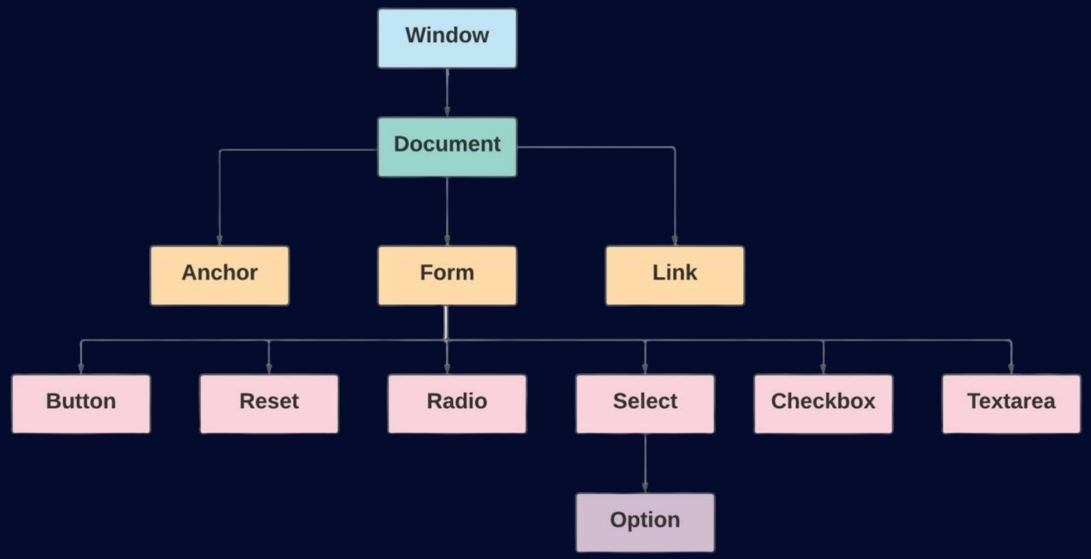
DOM

Document Object Model



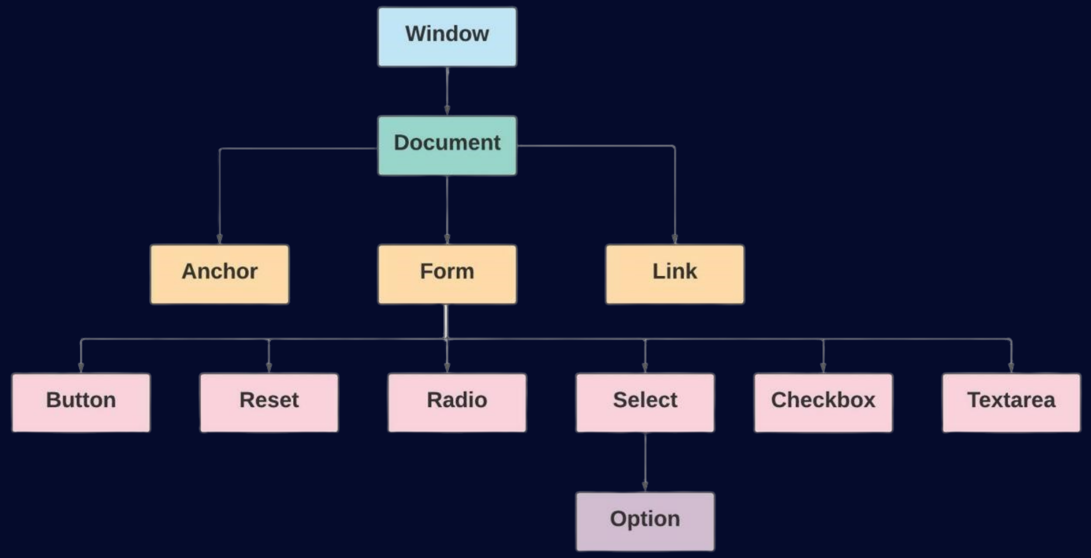
DOM

Document Object Model

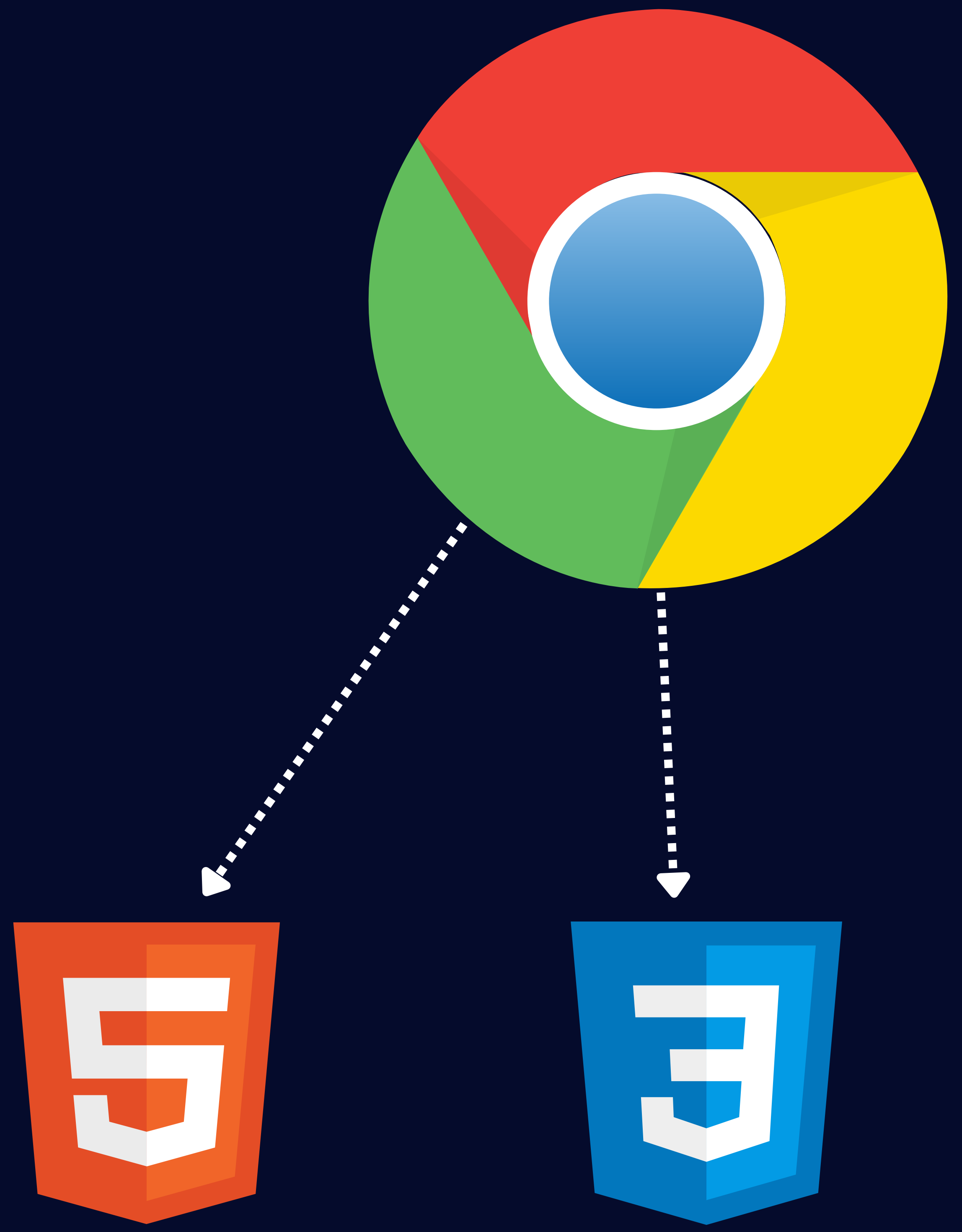


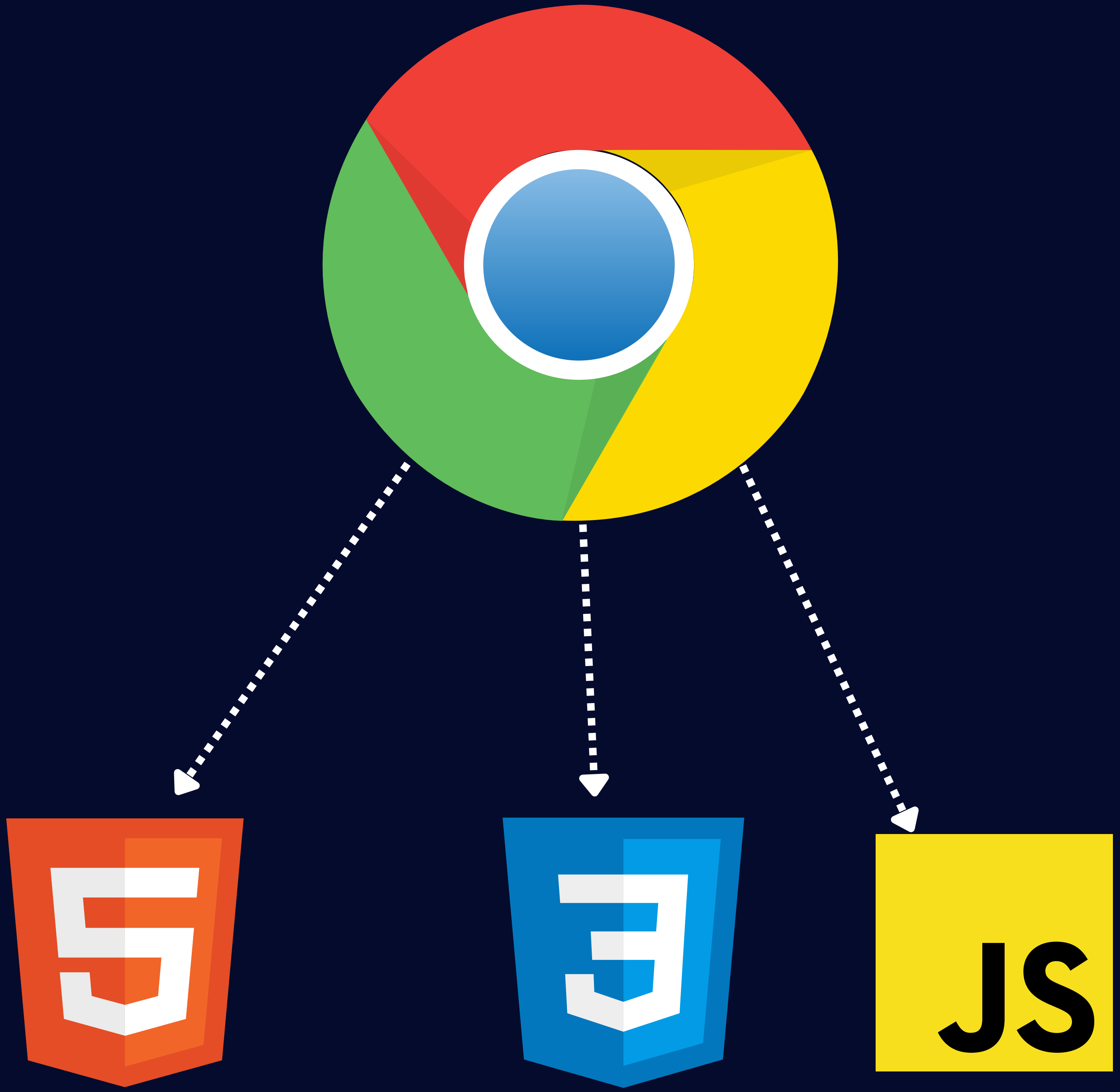
DOM

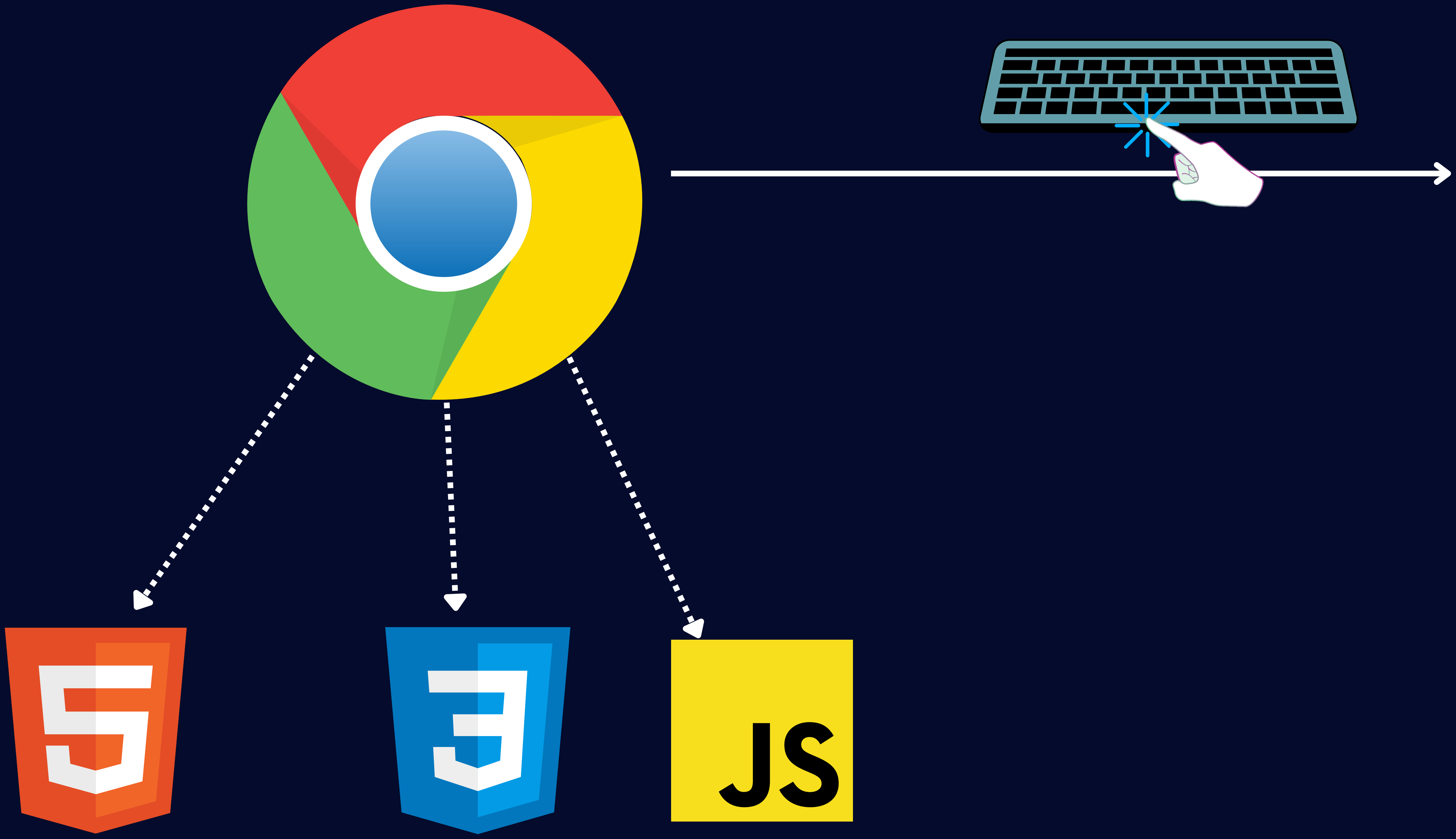
Document Object Model

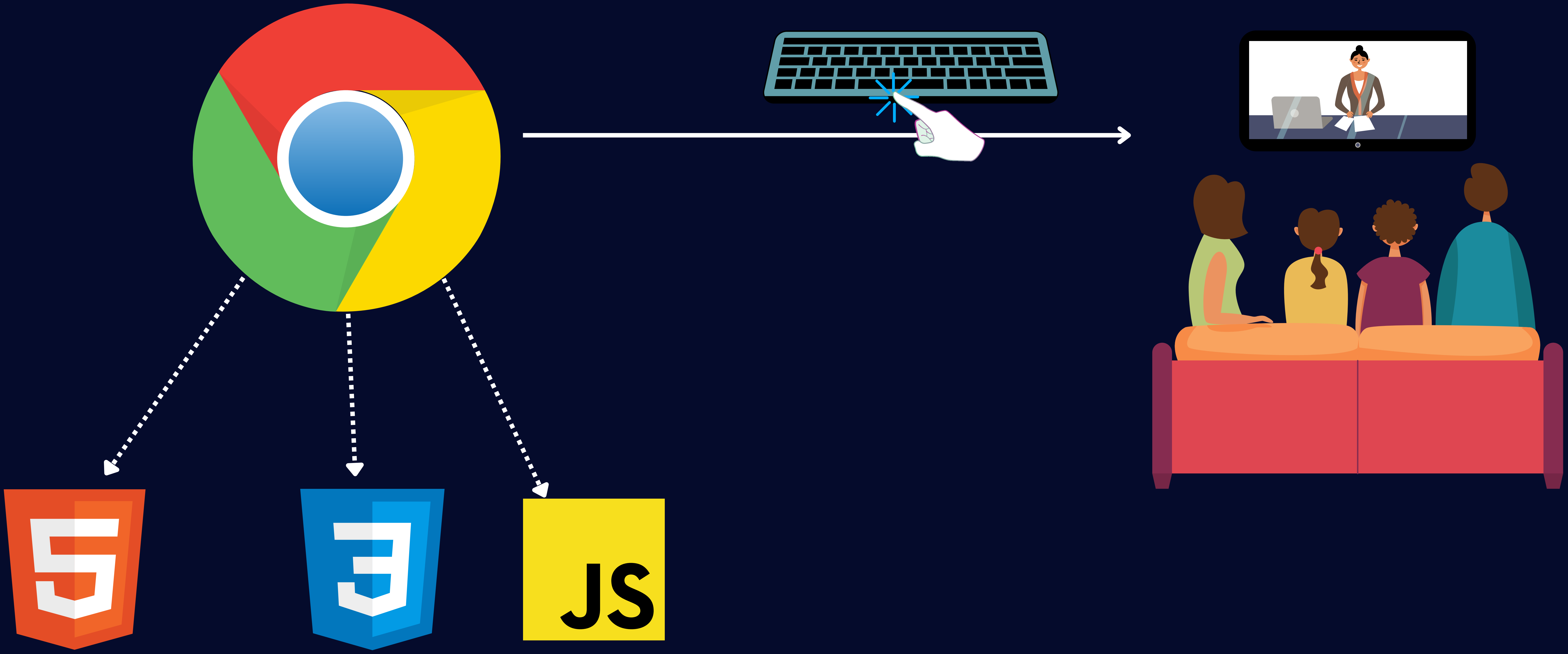


DOM is a tree structure that represents the structure of the HTML document






JS



JS

JS

DOM

DOM is the interface between the browser and the HTML document

DOM

- DOM is the interface between the browser and the HTML document
- DOM makes it possible to use javascript to interact with the HTML document

Uses of DOM

Uses of DOM

- To create new HTML element

Uses of DOM

- To create new HTML element
- To remove HTML elements

Uses of DOM

- To create new HTML element
- To remove HTML elements
- To add styles to HTML elements

Uses of DOM

- To create new HTML element
- To remove HTML elements
- To add styles to HTML elements
- To get values from input field

JS

Uses of DOM

- To create new HTML element
- To remove HTML elements
- To add styles to HTML elements
- To get values from input field
- To set attributes to element

Uses of DOM

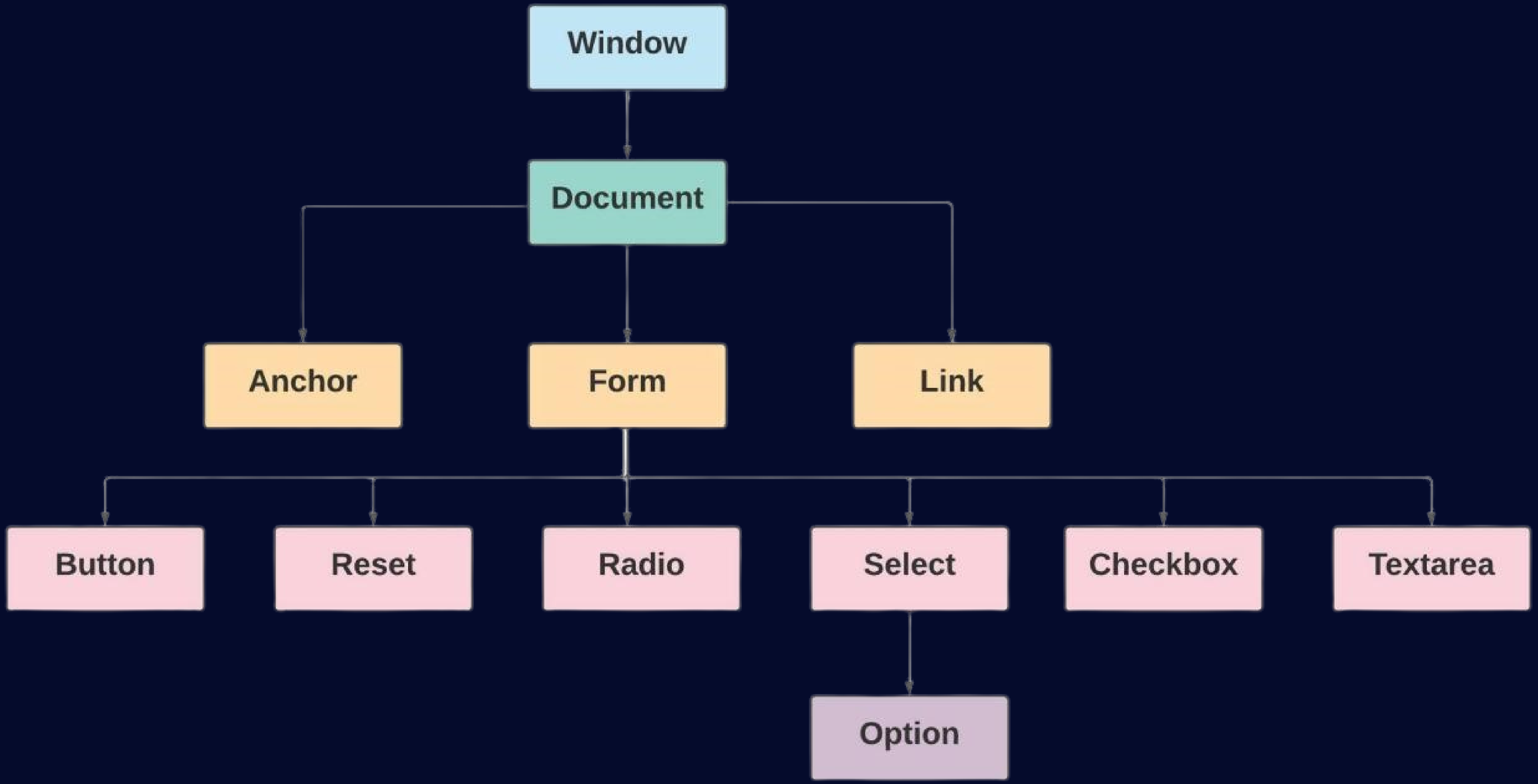
- To create new HTML element
- To remove HTML elements
- To add styles to HTML elements
- To get values from input field
- To set attributes to element
- To add event listener to element

NODE

NODE

A node is an individual parts of element in the DOM tree

NODE

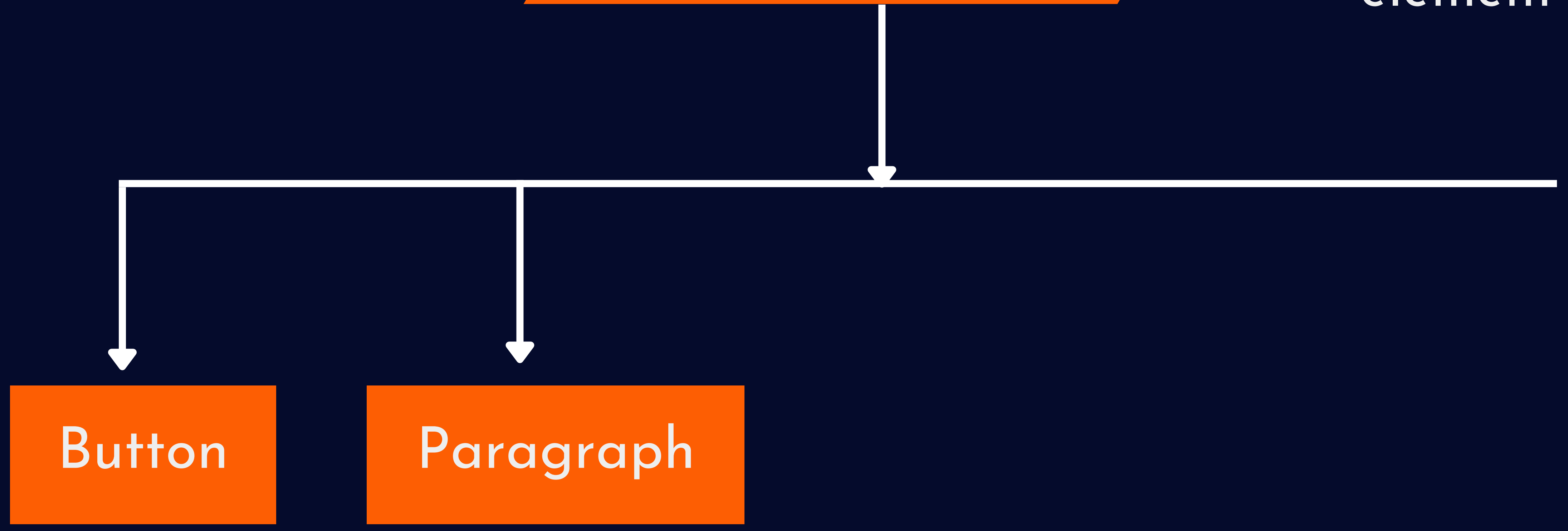

Window
Document
Anchor
Form
Link
Button
Reset
Radio
Select
Checkbox
Textarea
Option

NODE

A node is an individual parts of element in the DOM tree

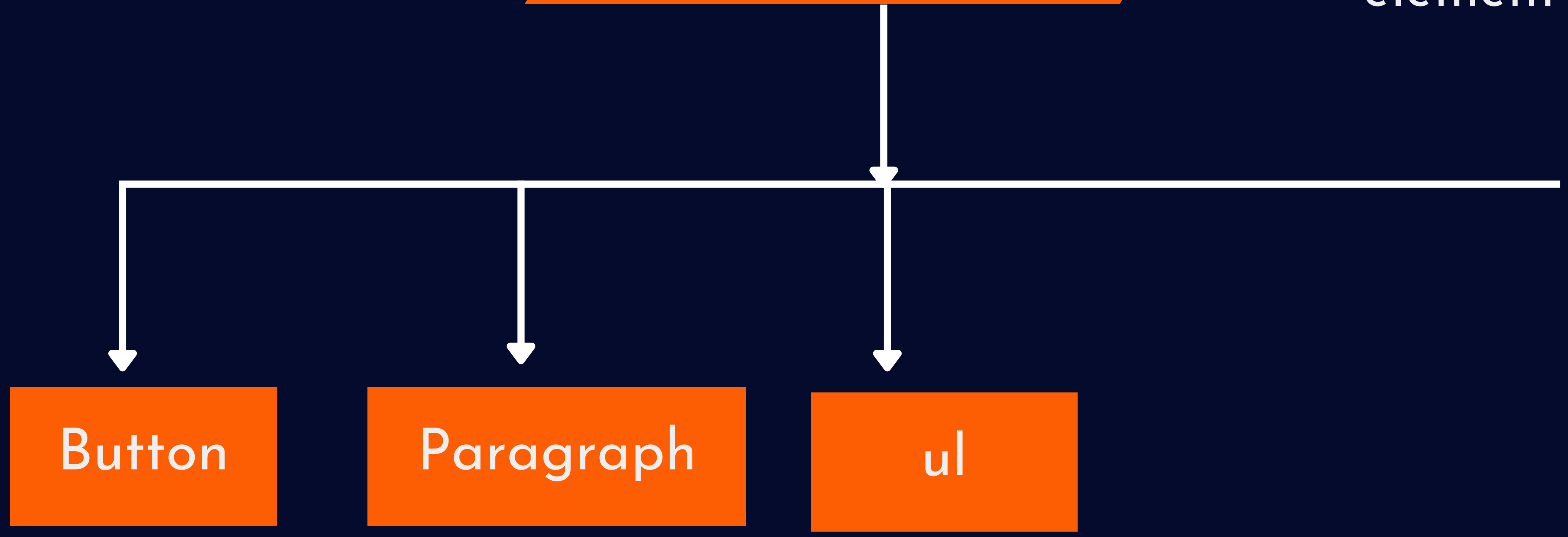
NODE

A node is an individual parts of element in the DOM tree



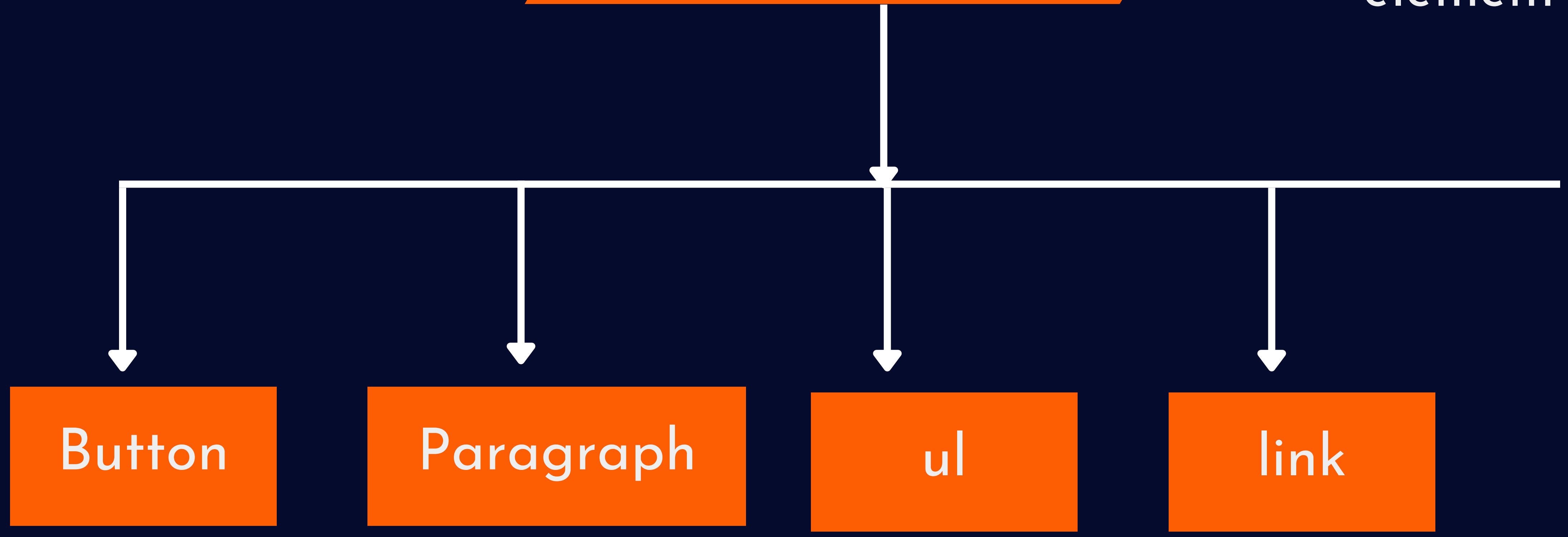
NODE

A node is an individual parts of element in the DOM tree



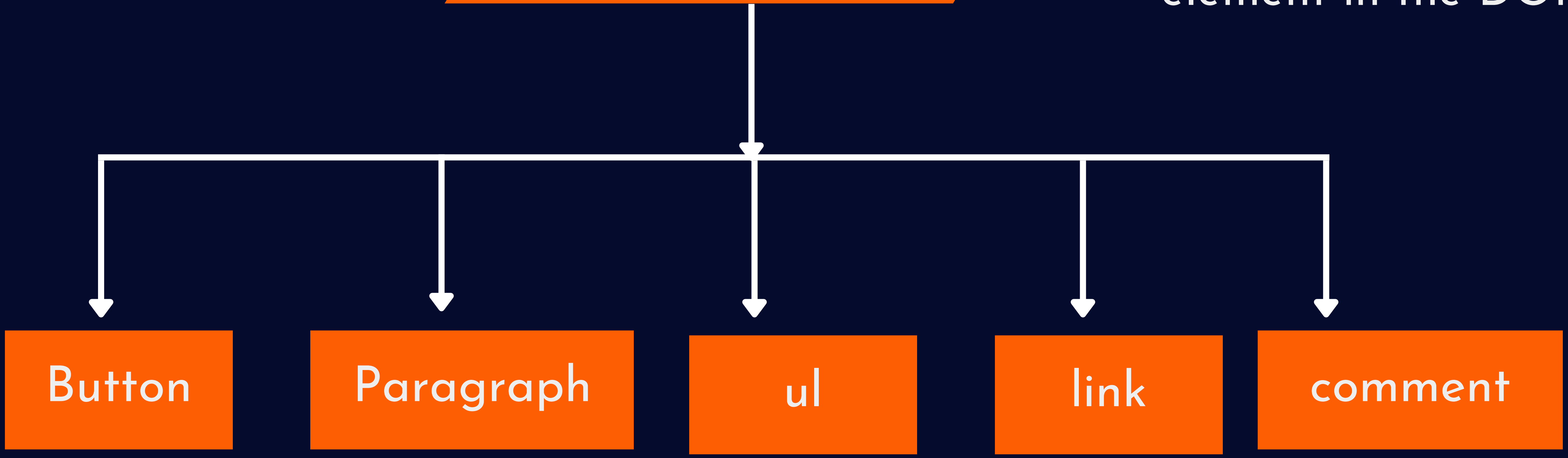
NODE

A node is an individual parts of element in the DOM tree



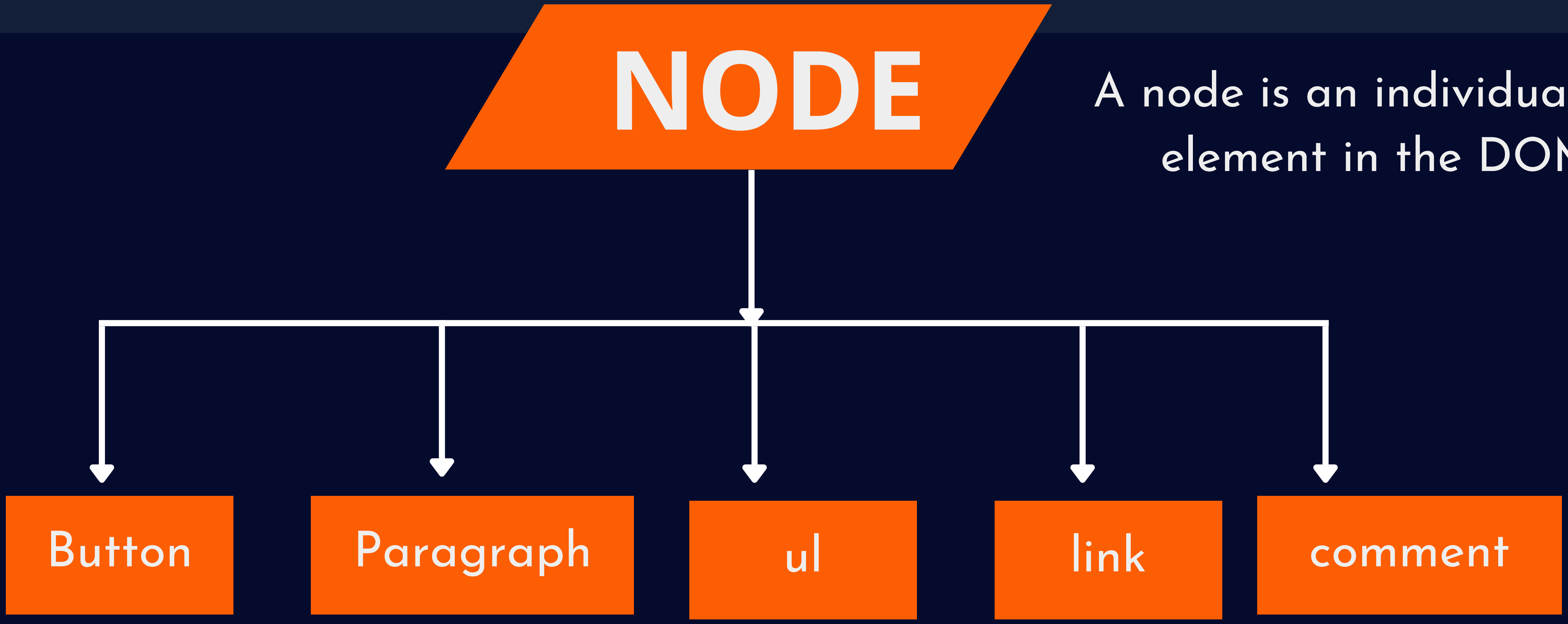
NODE

A node is an individual parts of element in the DOM tree



NODE

A node is an individual parts of element in the DOM tree



Node can have children/attributes/parents .....

JS

Facts about NODE

- The DOM node is an object
- The DOM node has properties and methods

JS

Examining the DOM

Types of Selectors

Types of Selectors

Single	

Types of Selectors

Single	Multiple

Types of Selectors

Single	Multiple
getElementById	

Types of Selectors

Single	Multiple
getElementById	
querySelector	

Types of Selectors

Single	Multiple
getElementById	getElementsByTagName
querySelector	

Types of Selectors

Single	Multiple
getElementById	getElementsByTagName
querySelector	getElementsByClassName

JS

Types of Selectors

Single	Multiple
getElementById	getElementsByTagName
querySelector	getElementsByClassName
	getElementsByName

Types of Selectors

Single	Multiple
getElementById	getElementsByTagName
querySelector	getElementsByClassName
	getElementsByName
	querySelectorAll

Types of Selectors

Single	Multiple
getElementById	getElementsByTagName
querySelector	getElementsByClassName
	getElementsByName
	querySelectorAll
	getElementsByTagName

JS

Changing Elements Properties

All selected elements has a property called style

JS

Changing Elements Properties

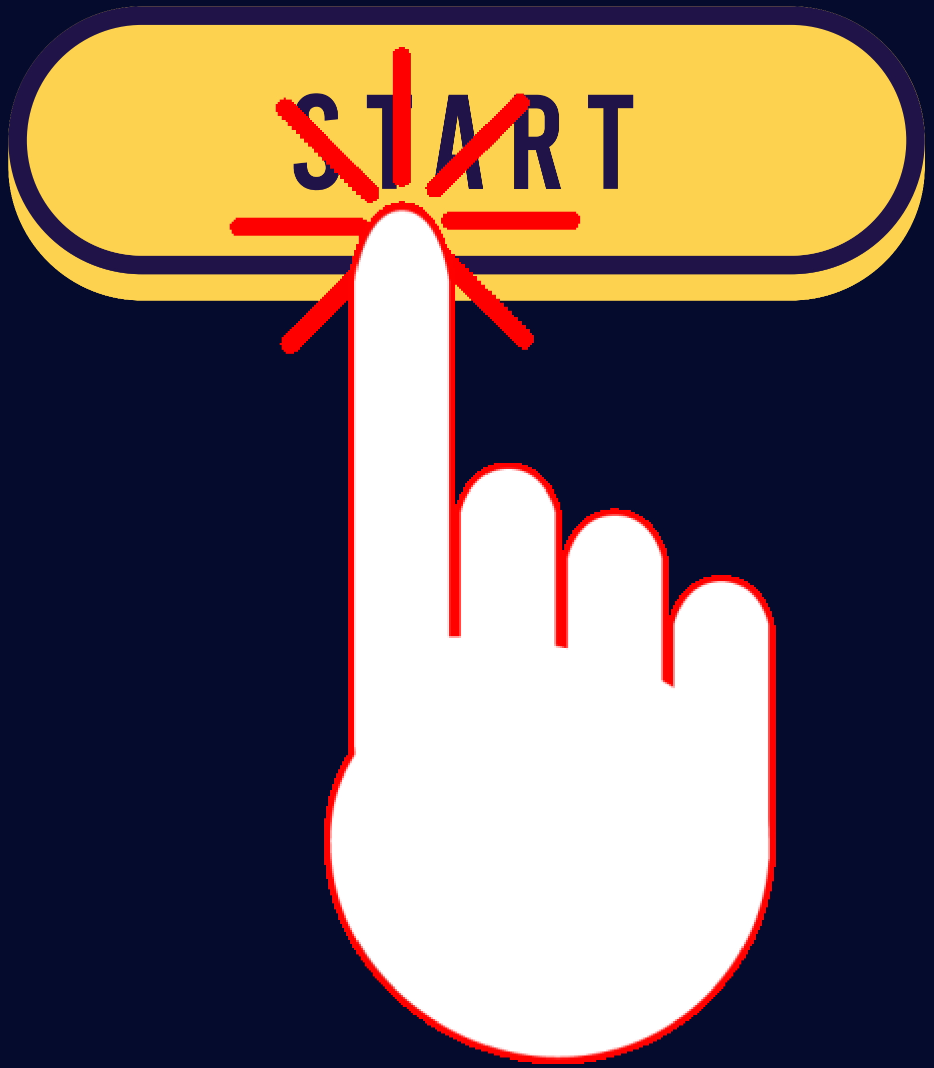
Changing Elements Properties

All selected elements has a property called style

Changing Elements Properties

All selected elements has a property called style

JS

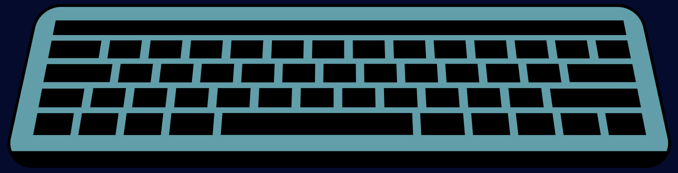


DOM EVENTS

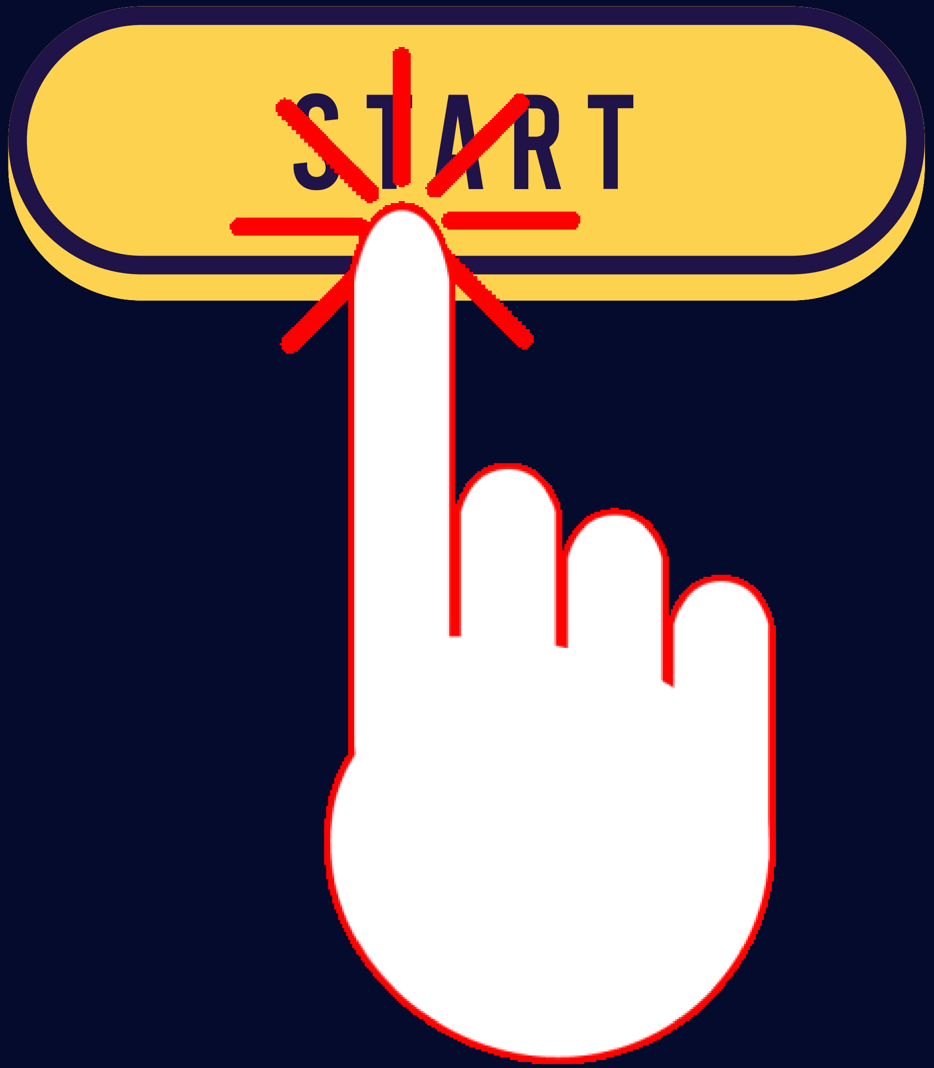
DOM EVENTS

- Mouse Events

- Keyboard Events



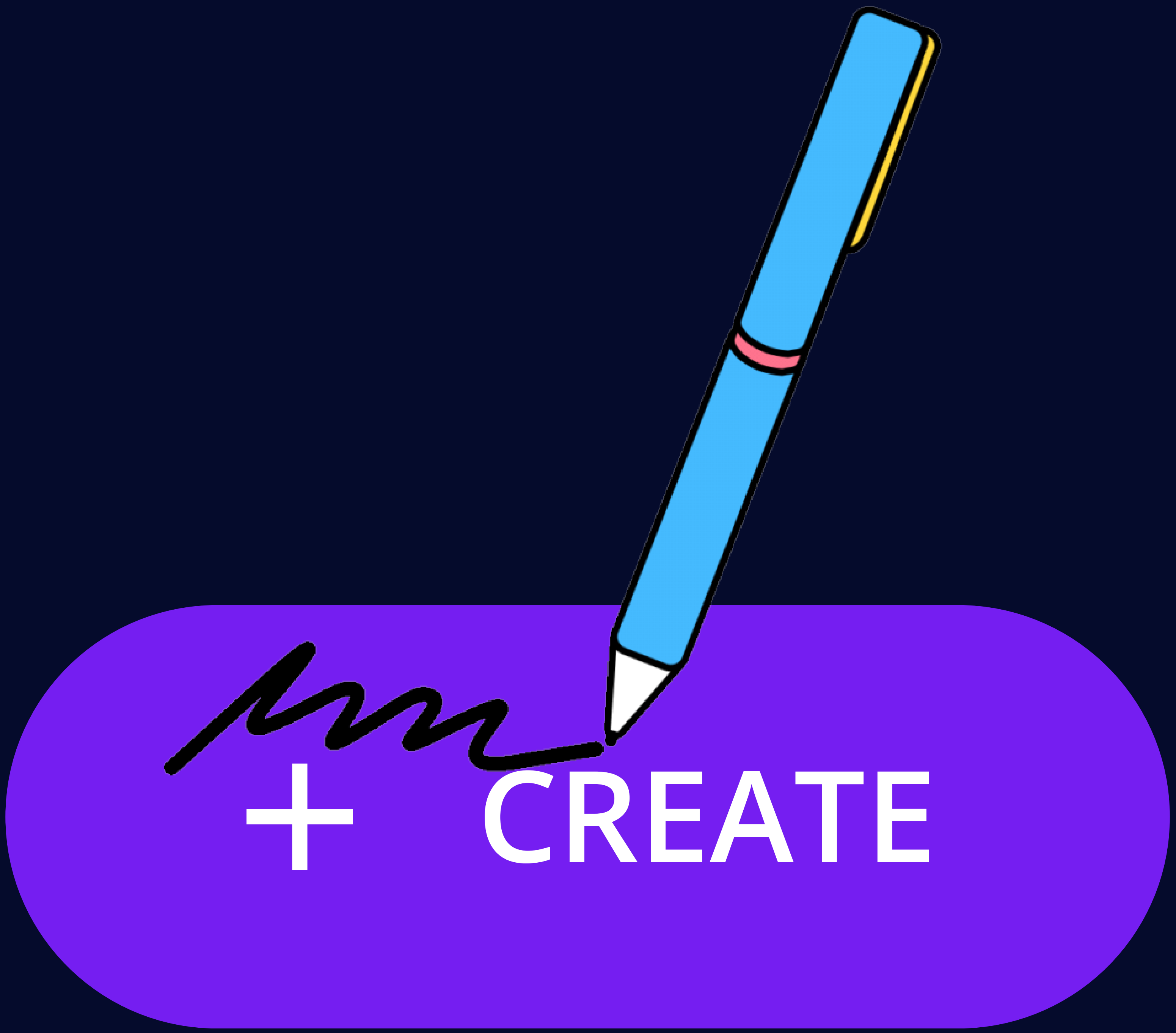
- Form Events



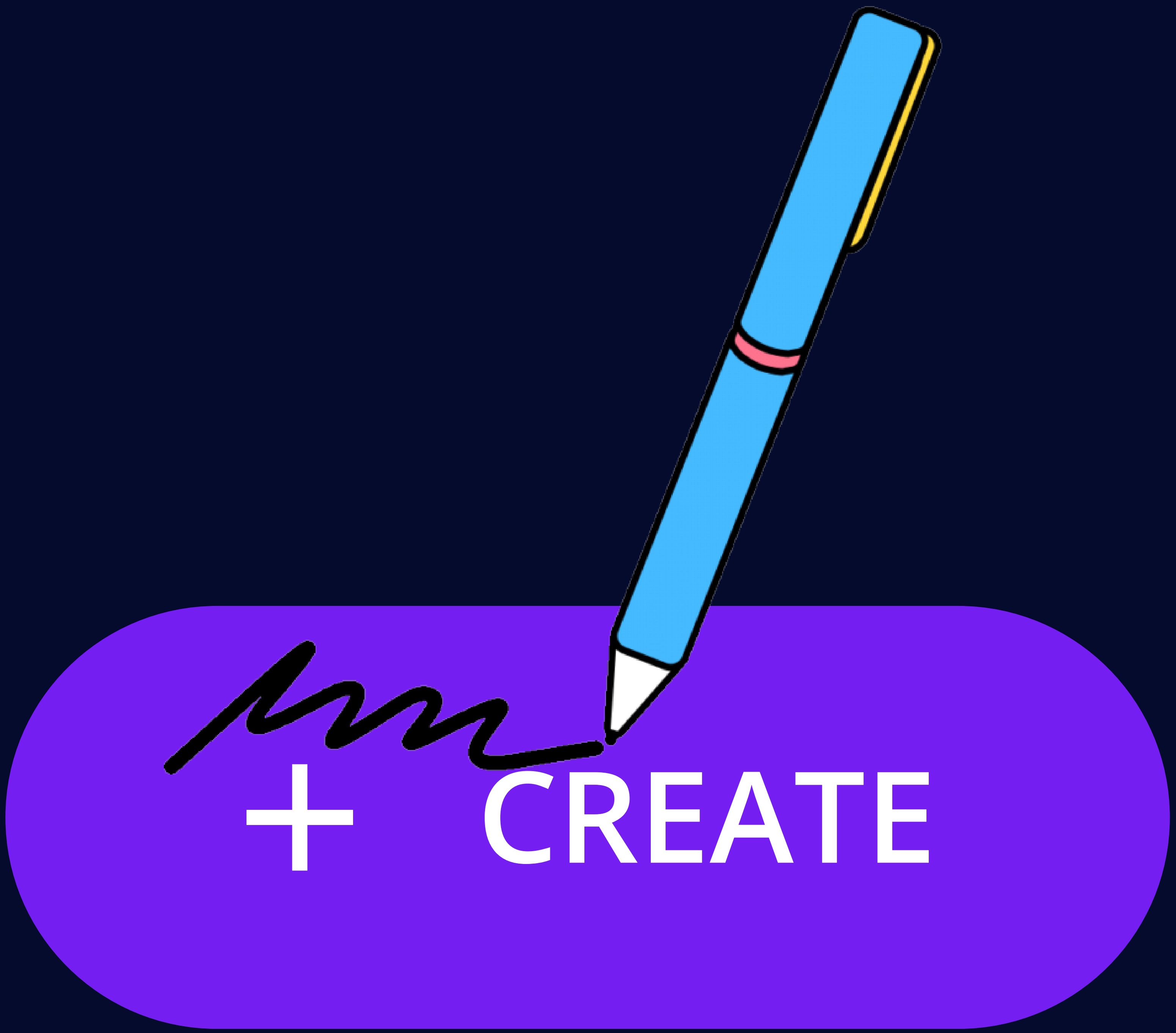
DOM EVENTS

JS

How to Create Elements



JS



How to Create Elements

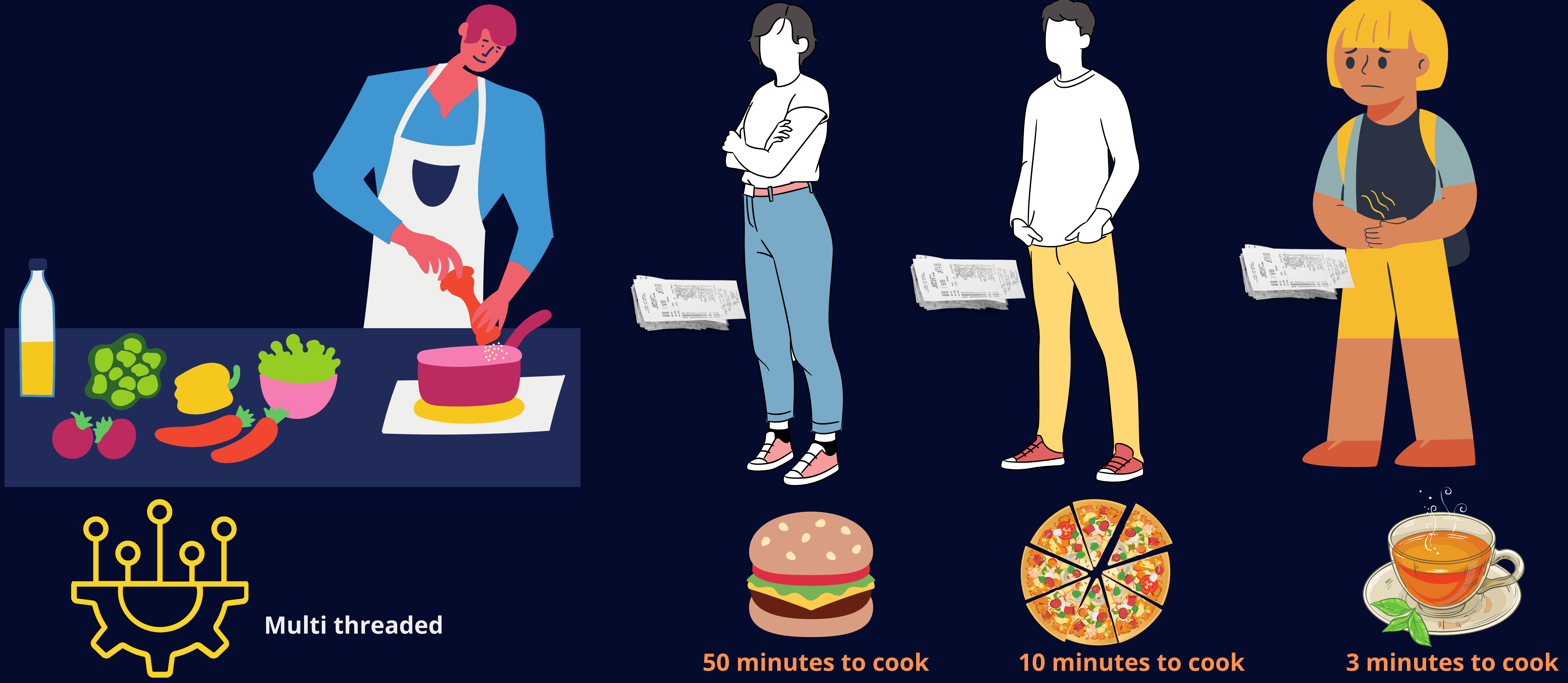
JS

Local Storage

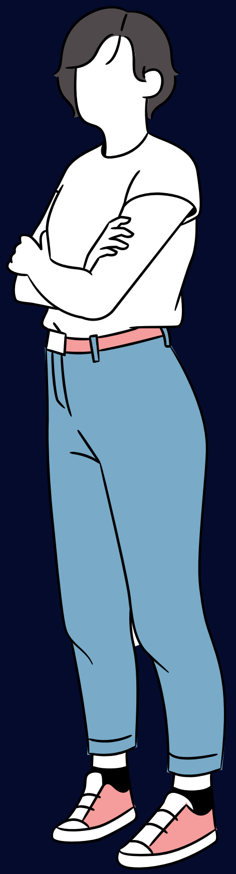
Asynchronous programming

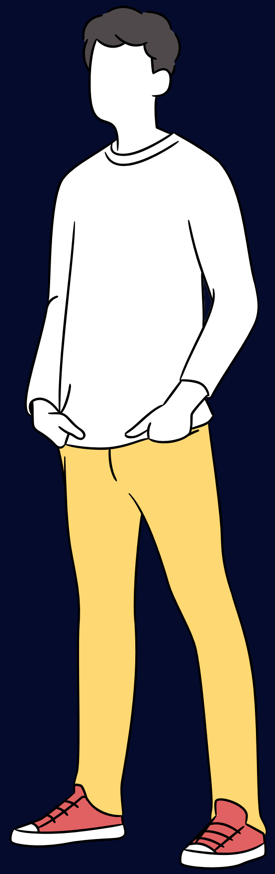
Multi threaded

Asynchronous programming



Asynchronous programming





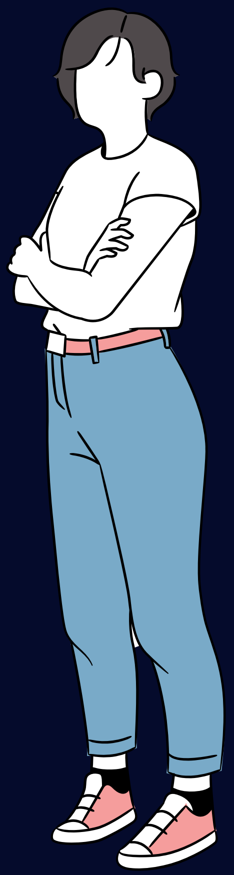
Multi threaded

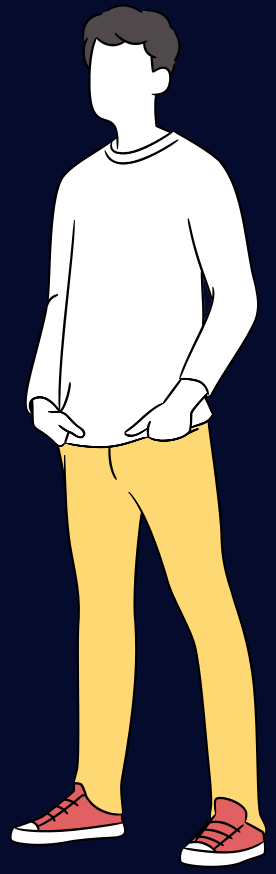
50 minutes to cook

10 minutes to cook

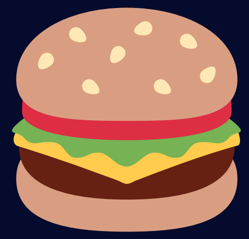
3 minutes to cook

Asynchronous programming





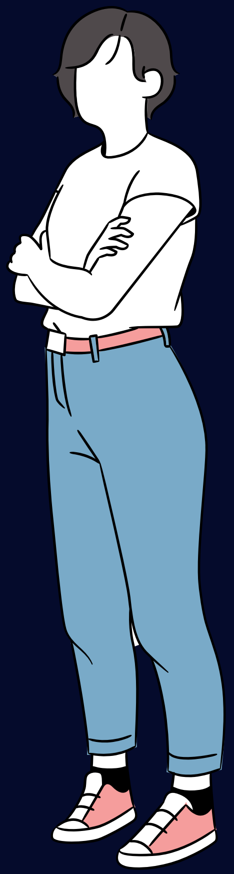
Multi threaded



50 minutes to cook

10 minutes to cook

Asynchronous programming



Multi threaded

50 minutes to cook

JS

Asynchronous programming

Multi threaded

Ways of writing

Async Code

Ways of writing Async Code

SetTimeout

Ways of writing Async Code

SetTimeout

Callback

Ways of writing Async Code

SetTimeout

Callback

Promise

Ways of writing Async Code

SetTimeout

Callback

Promise

Async/Await

Async Programming

SetTimeout

Async Programming

SetTimeout

Async Programming

SetTimeout

SetTimeout is a function that runs after a certain amount of time has passed and it is not blocking the code from executing

Async Programming

SetTimeout

SetTimeout is a function that runs after a certain amount of time has passed and it is not blocking the code from executing

Syntax

```
setTimeout(function(){}, time)
```

Callbacks

Callbacks

Callbacks

Callbacks

Callbacks

Callbacks

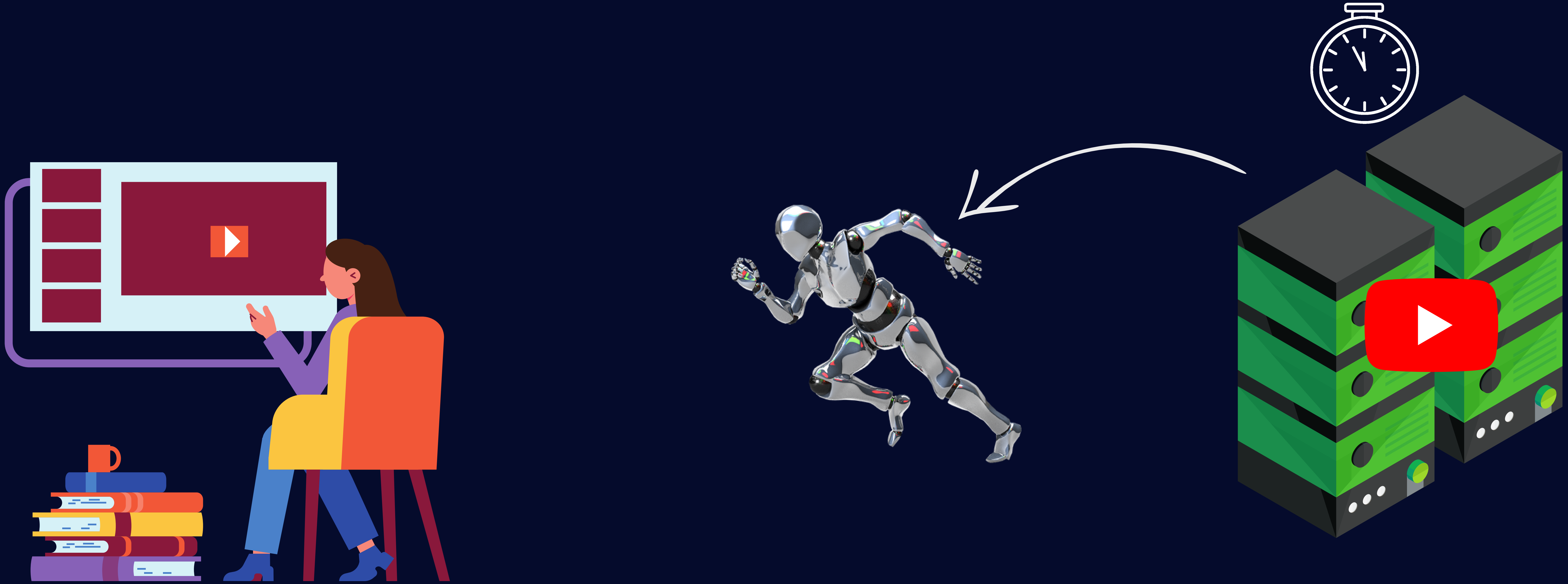
Callbacks

www.inovotekacademy.com

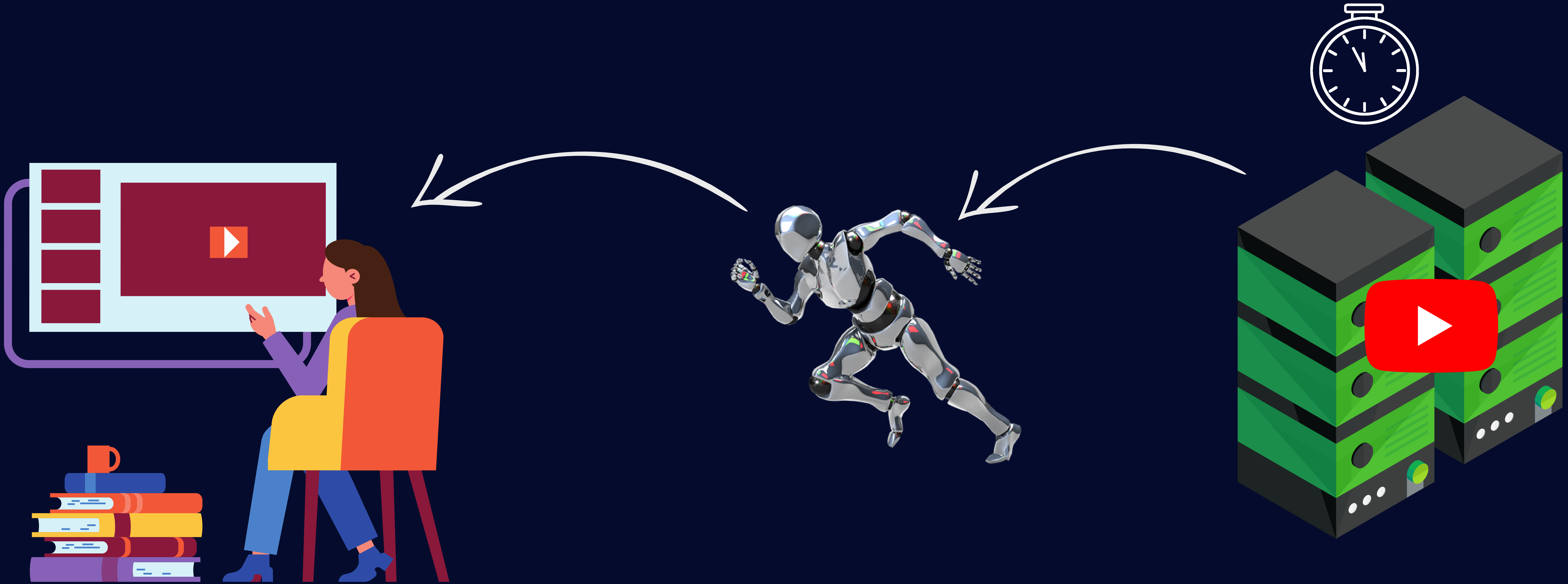
i-novotek academy

inovotekacademy

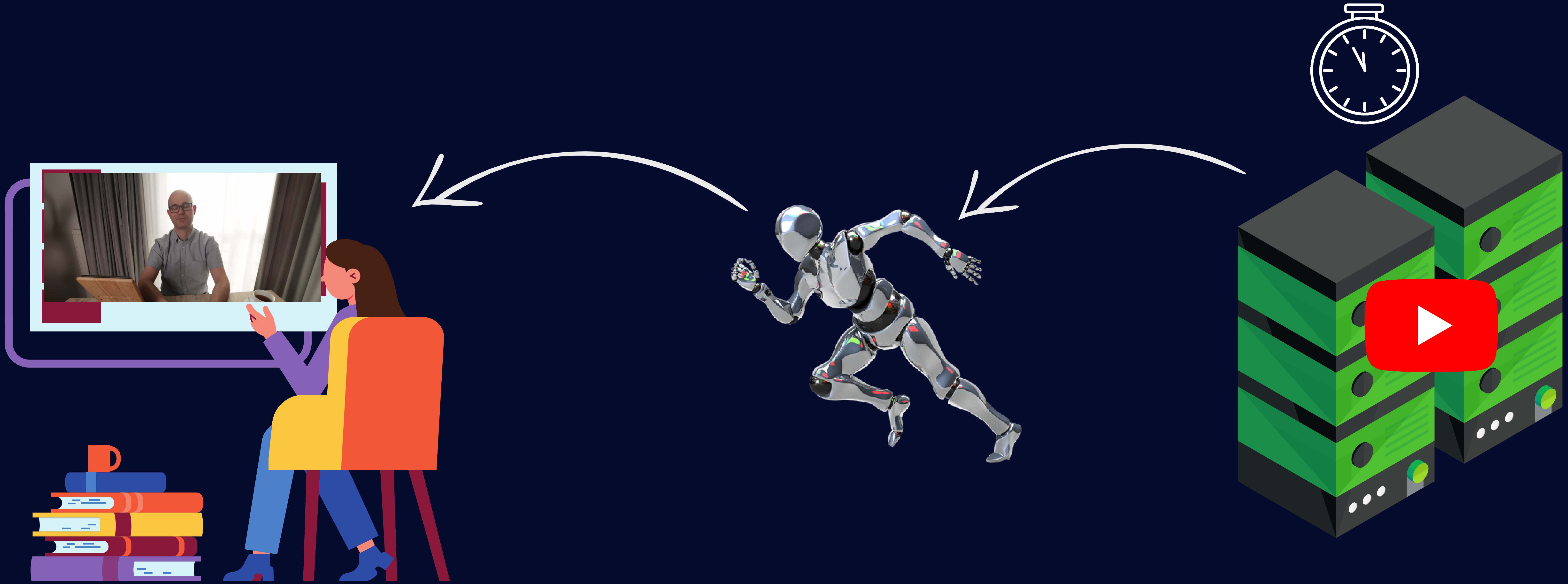
Callbacks



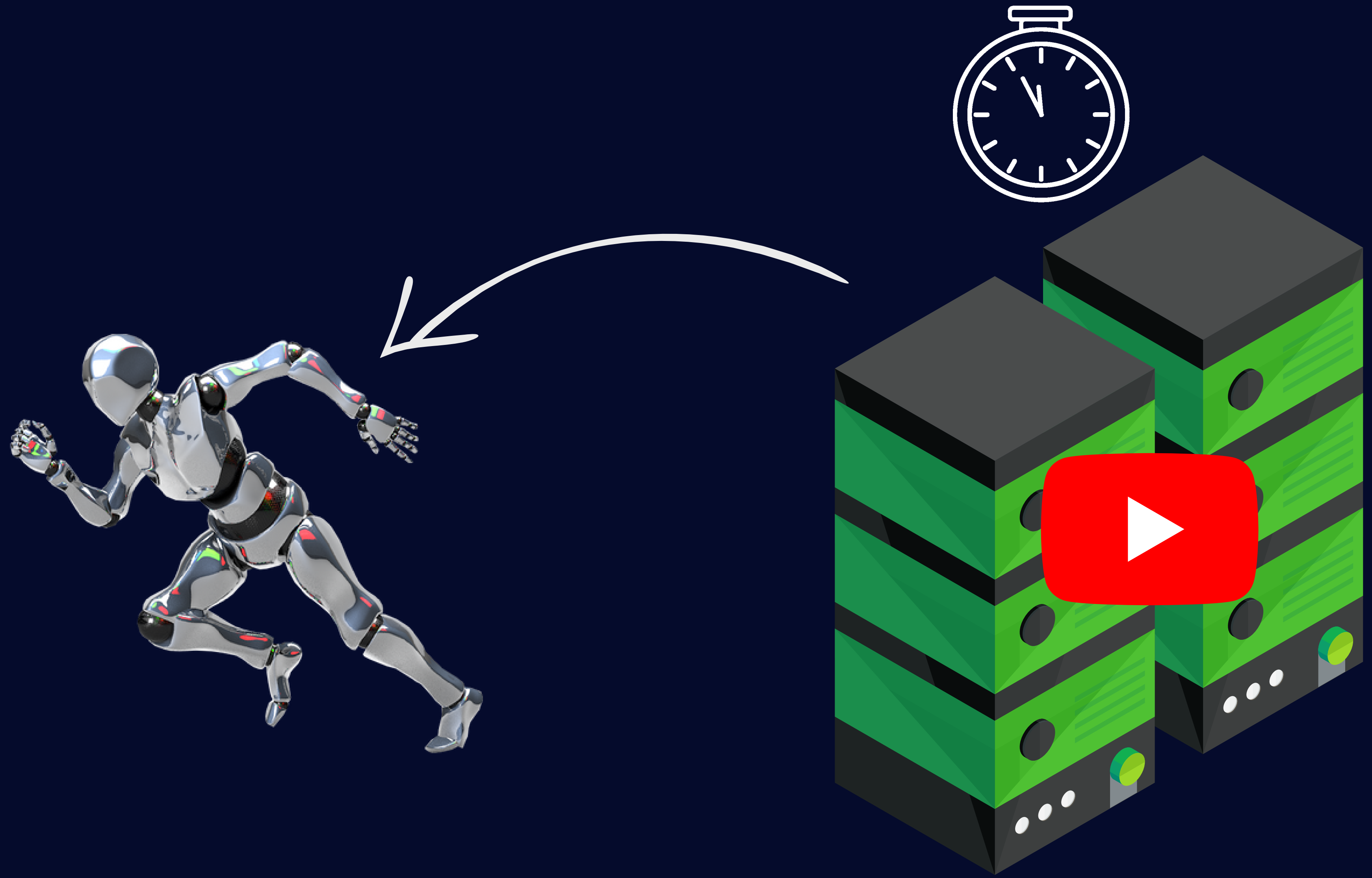
Callbacks



Callbacks



Callbacks



Callbacks

Async Programming

Callbacks

Async Programming

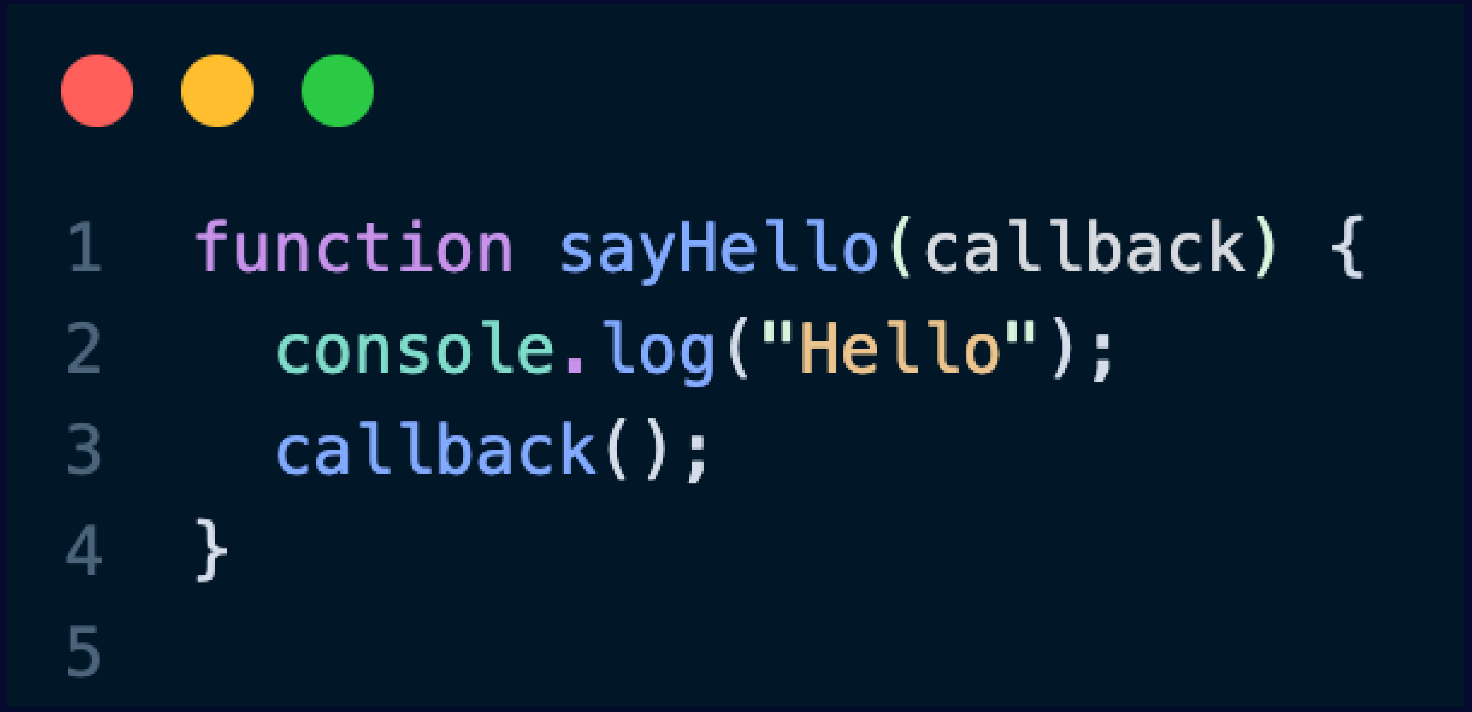
Callbacks

This is a function that is passed to another function as an argument. This function is then executed after the function that is passed to is executed.

Async Programming

Callbacks

This is a function that is passed to another function as an argument. This function is then executed after the function that is passed to is executed.

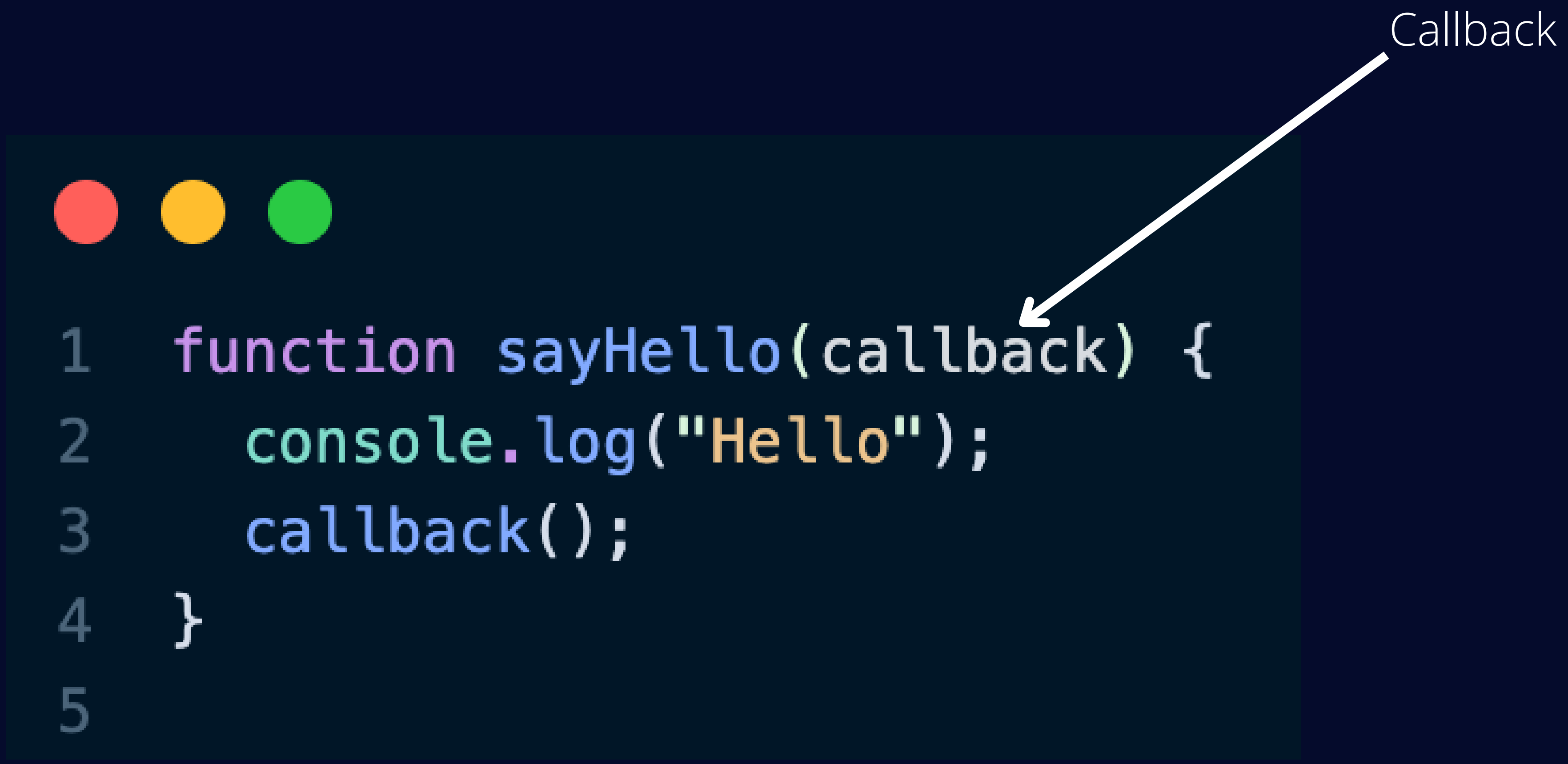


```
function sayHello(callback) {
  console.log("Hello");
  callback();
}

```

Async Programming

Callbacks

This is a function that is passed to another function as an argument. This function is then executed after the function that is passed to is executed.



```javascript
function sayHello(callback) {
  console.log("Hello");
  callback();
}

```

Async Programming

Promise

Async Programming

Promise

Promise represents an object that may produce a value sometime in the future.

Async Programming

Promise

Promise represents an object that may produce a value sometime in the future.

The values can be:

Async Programming

Promise

Promise represents an object that may produce a value sometime in the future.

The values can be:

- pending

Async Programming

Promise

Promise represents an object that may produce a value sometime in the future.

The values can be:

- pending
- success (fulfilled)

Async Programming

Promise

Promise represents an object that may produce a value sometime in the future.

The values can be:

- pending
- success (fulfilled)
- failure

Facts about promises

Promise

JS

Facts about promises

Promise

When a promise is created, it is in the pending state

JS

Facts about promises

Promise

When a promise is created, it is in the pending state

When a promise is resolved, it is in the fulfilled state

JS

Facts about promises

Promise

When a promise is created, it is in the pending state

When a promise is resolved, it is in the fulfilled state

When a promise is rejected, it is in the rejected state

JS

Facts about promises

Promise

When a promise is created, it is in the pending state

When a promise is resolved, it is in the fulfilled state

When a promise is rejected, it is in the rejected state

When returning a promise from a function, the function will return a promise but not the value of the promise. And the value of the promise will be returned when the promise is resolved.

JS

Promise

How to Create a Promise

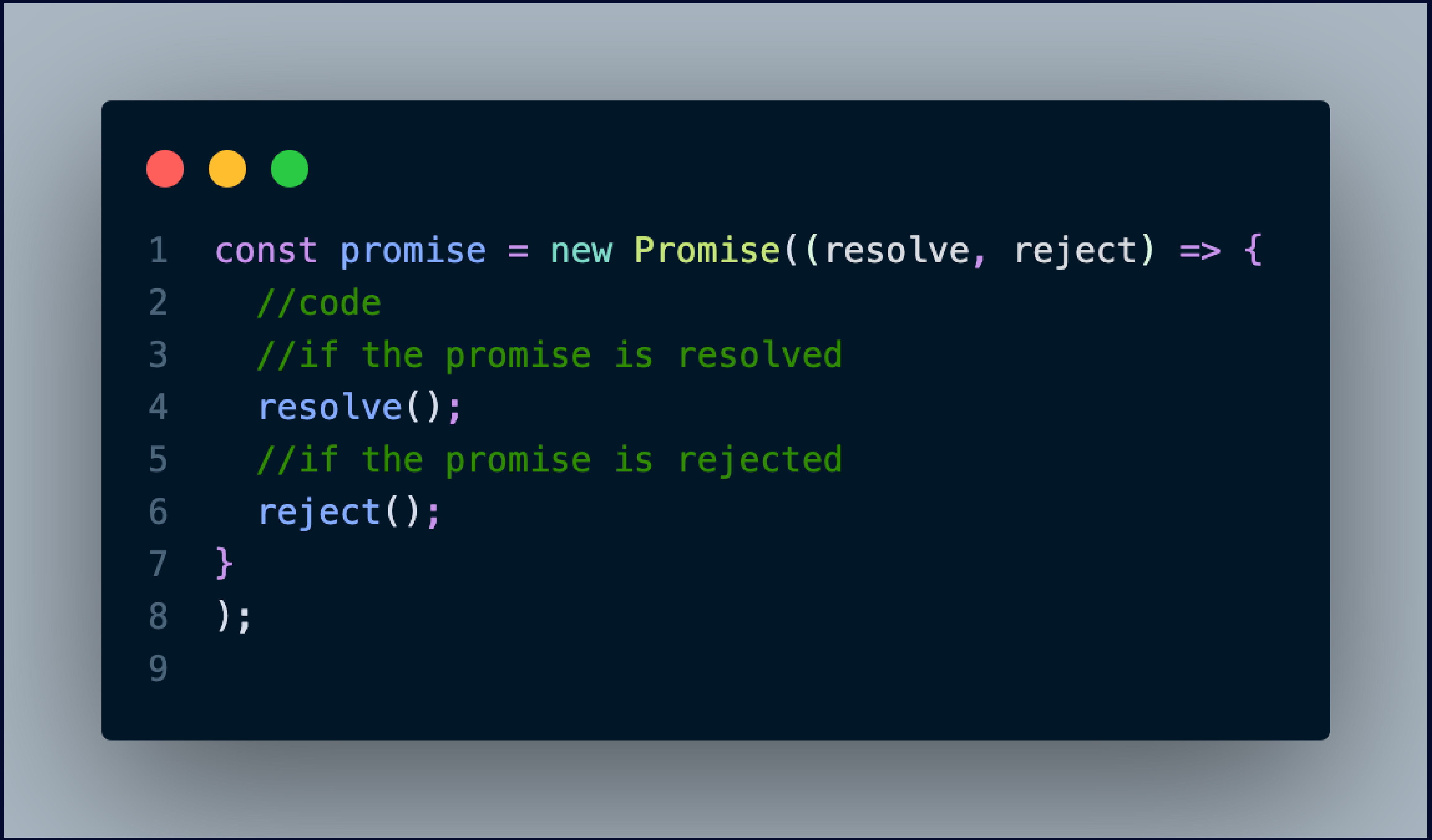
www.inovotekacademy.com

i-novotek academy

inovotekacademy

Async Programming

Promise



```javascript
const promise = new Promise((resolve, reject) => {
  //code
  //if the promise is resolved
  resolve();
  //if the promise is rejected
  reject();
}
);
```

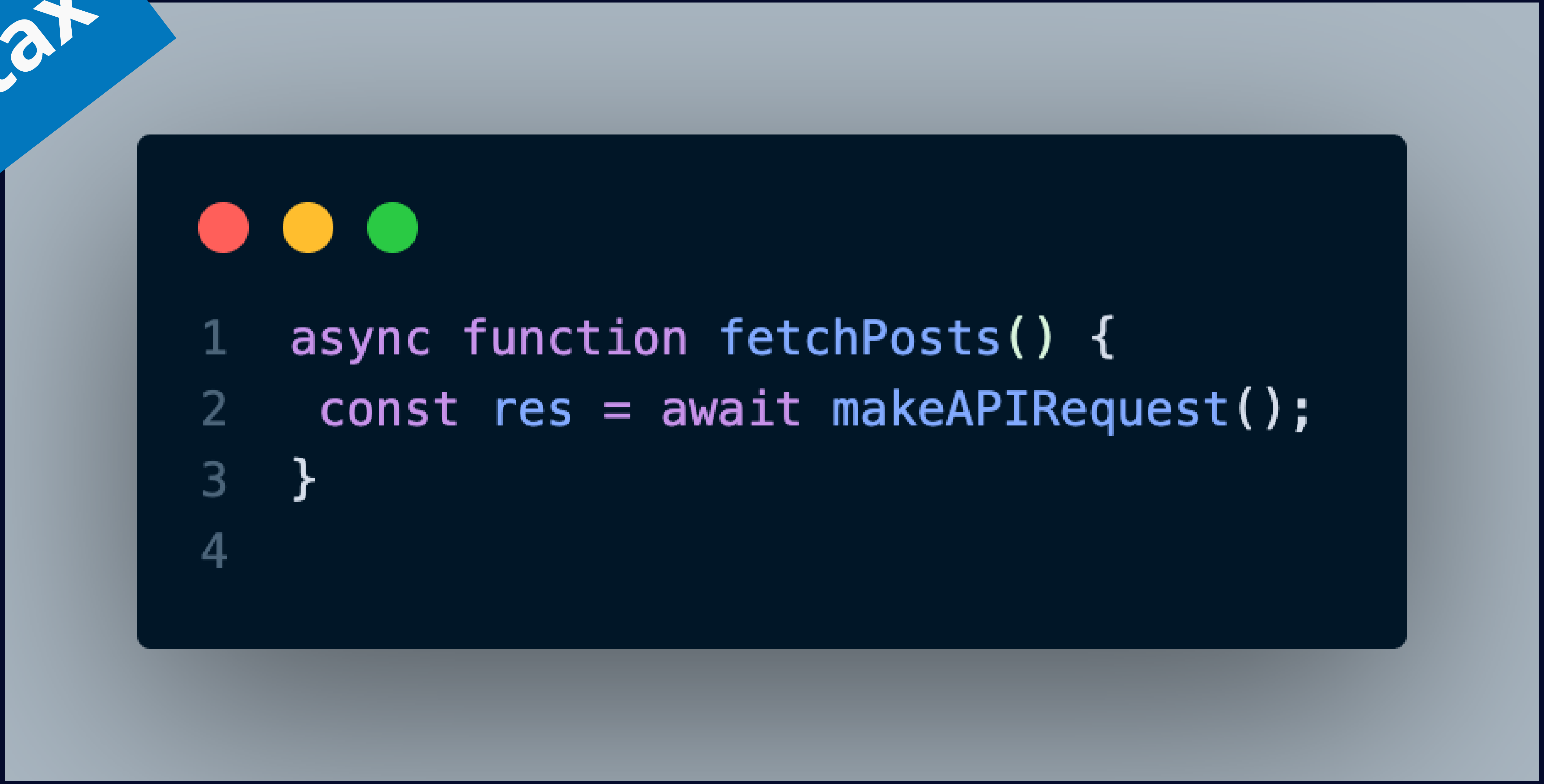
Promise

Function returning a promise

JS

Async/Await

Async/Await



```
async function fetchPosts() {
 const res = await makeAPIRequest();
}
```

Async/Await

Rules

It is a syntactic sugar for promises

Async/Await

Rules

It is a syntactic sugar for promises

The await keyword is only valid inside async functions

Async/Await

Rules

It is a syntactic sugar for promises

The await keyword is only valid inside async functions

Async function must start with async keyword

Async/Await

Rules

It is a syntactic sugar for promises

The await keyword is only valid inside async functions

Async function must start with async keyword

Await is put in front of any promise based API call or function that will take some time to complete.

Async/Await

Rules

It is a syntactic sugar for promises

The await keyword is only valid inside async functions

Async function must start with async keyword

Await is put in front of any promise based API call or function that will take some time to complete.

Async function returns a promise.

Async/Await

Rules

It is a syntactic sugar for promises

The await keyword is only valid inside async functions

Async function must start with async keyword

Await is put in front of any promise based API call or function that will take some time to complete.

Async function returns a promise.

use use try/catch to handle success and errors in async/await

Async/Await

Rules

It is a syntactic sugar for promises

The await keyword is only valid inside async functions

Async function must start with async keyword

Await is put in front of any promise based API call or function that will take some time to complete.

Async function returns a promise.

use use try/catch to handle success and errors in async/await

use await to wait for the promise to resolve.