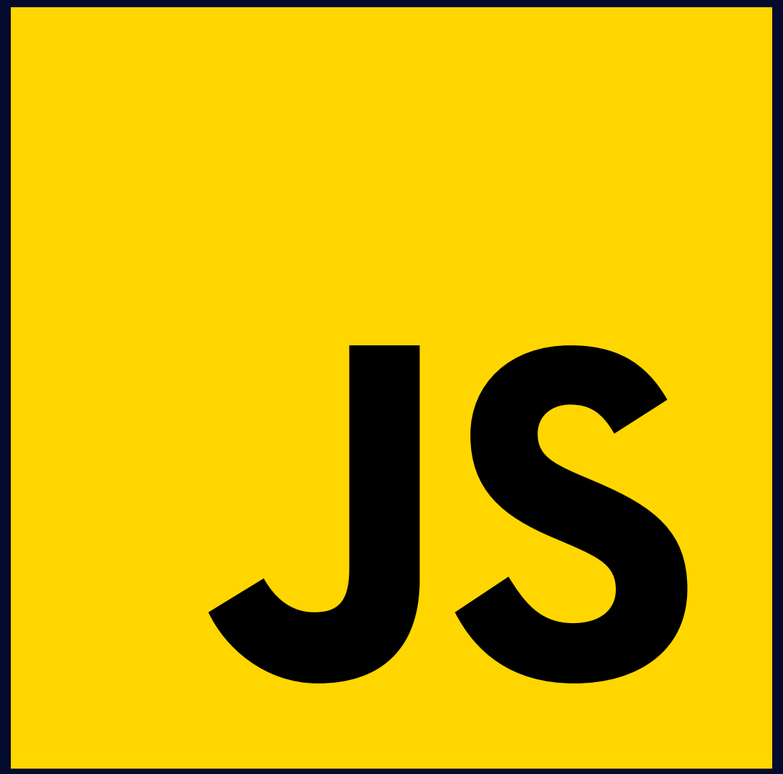
WELCOME TO FULL STACK WEB DEVELOPMENT

V2.2



Welcome to Javascript

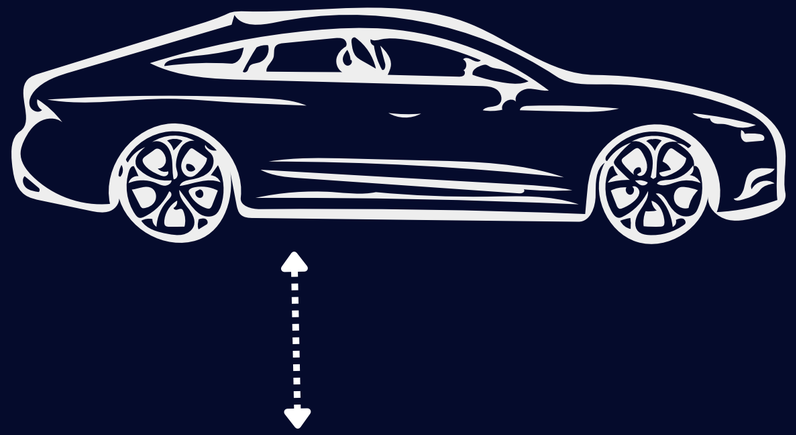
JS

The Basics of Javascript

What is Javascript

The Basics of Javascript

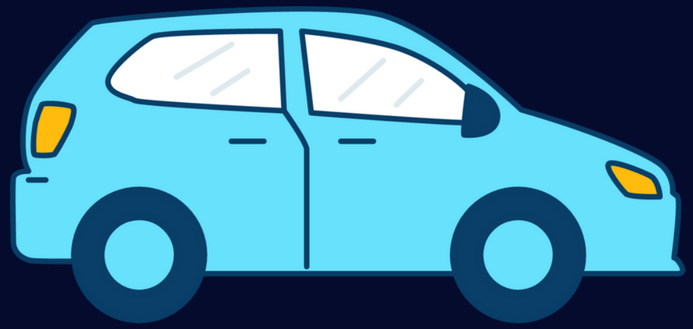
The Basics of Javascript



The Basics of Javascript

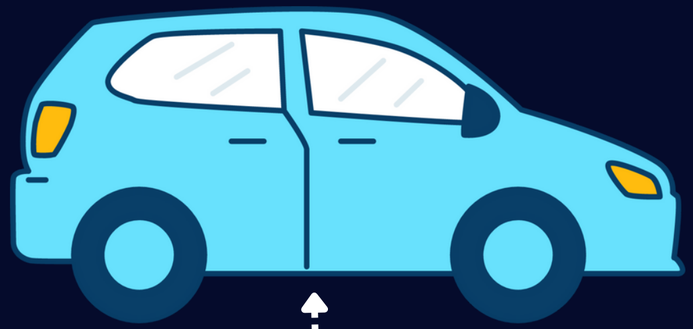
Noun

The Basics of Javascript



Noun

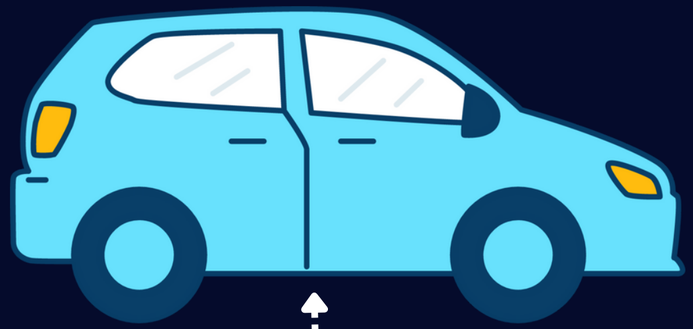
The Basics of Javascript



Noun

Adjective

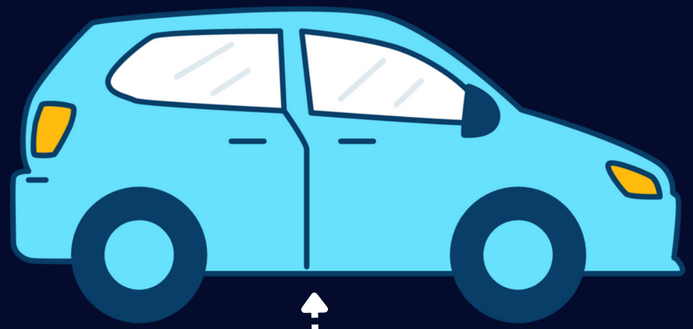
The Basics of Javascript



Noun

Adjective

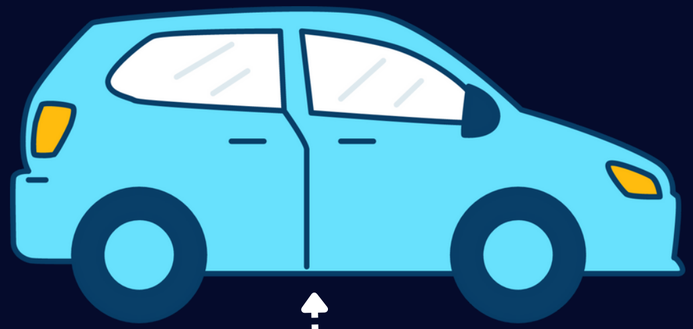
The Basics of Javascript



Noun

Adjective

The Basics of Javascript



Noun

Adjective

Verb

JS

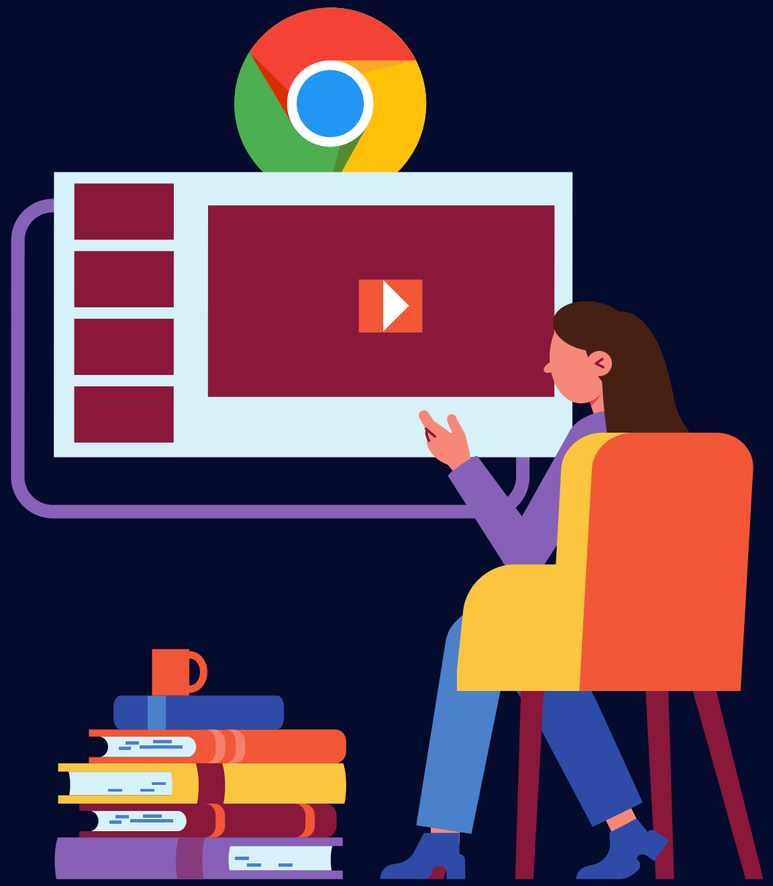
The Basics of Javascript

How Does Javascript Work?

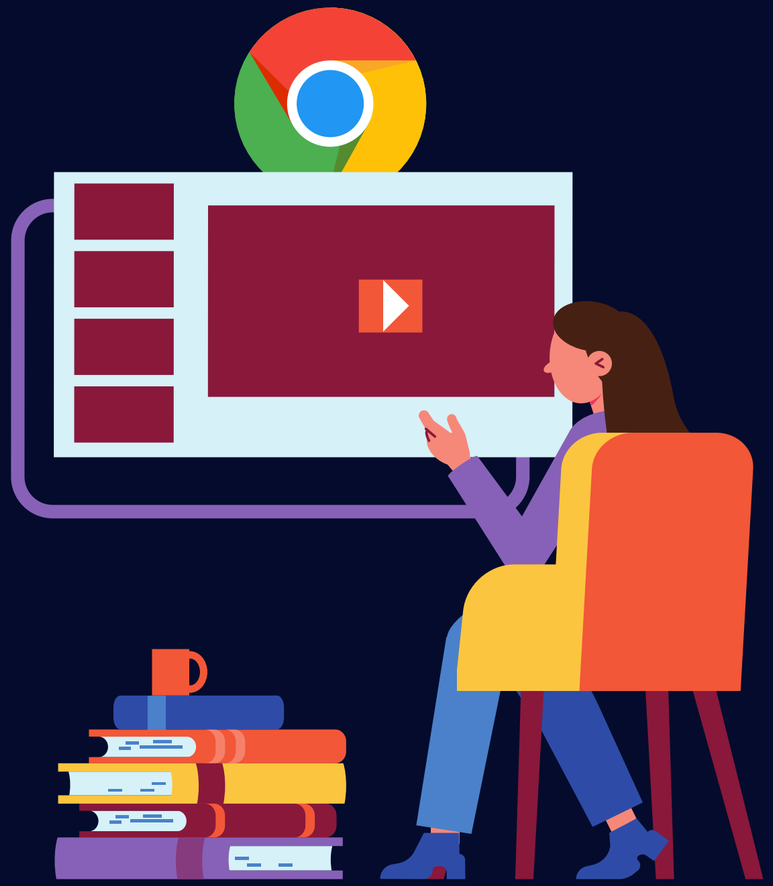
JS

The Basics of Javascript

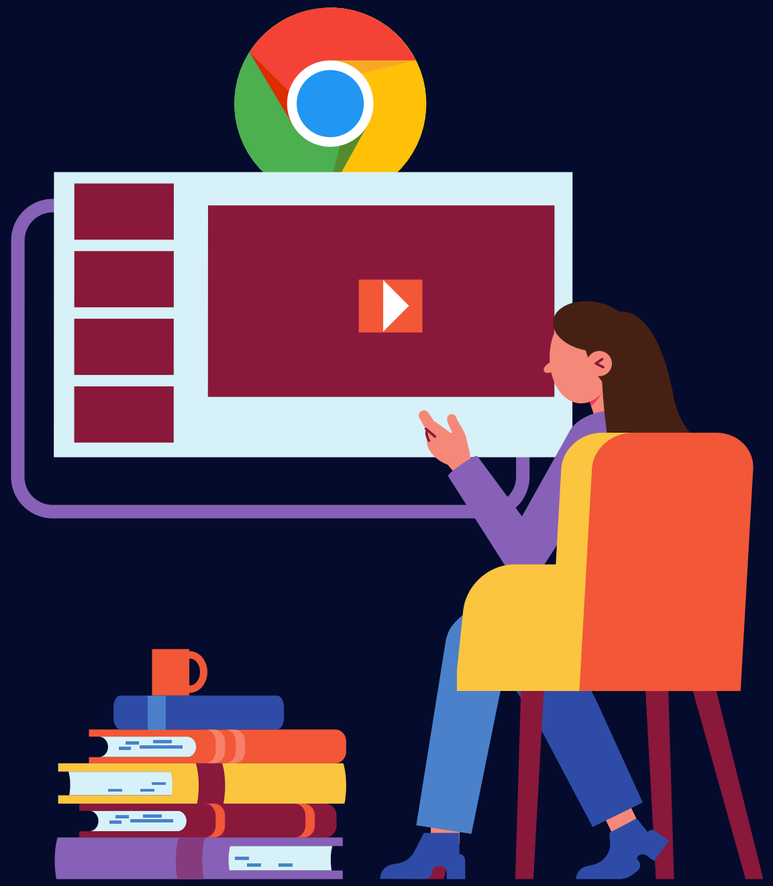
The Basics of Javascript



The Basics of Javascript

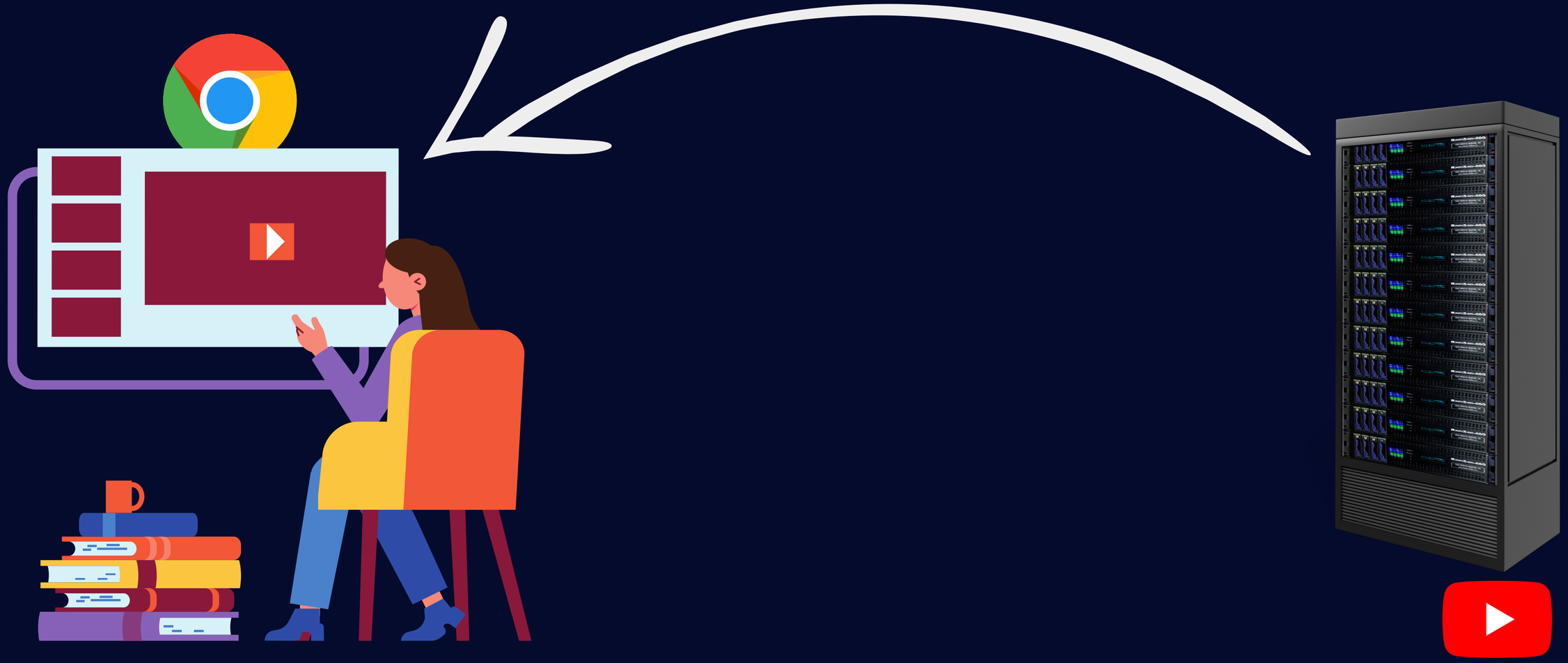


The Basics of Javascript

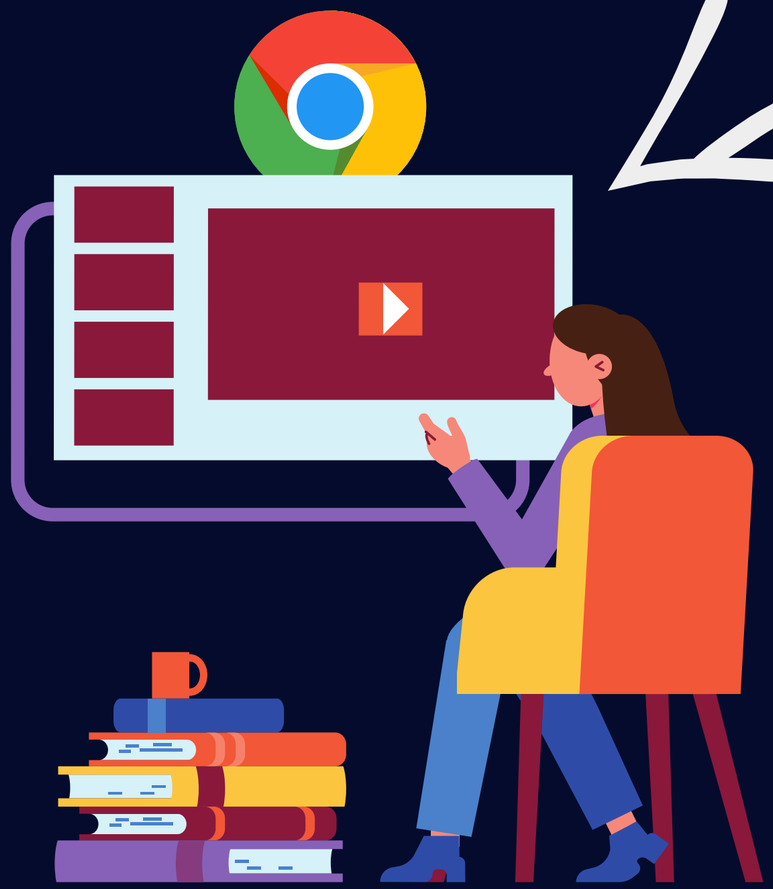


JS

The Basics of Javascript



The Basics of Javascript



The Basics of Javascript

View in browser console

JS

The Benefits of Javascript

JS

Javascript Variables

JS

What is a Variables ?

JS

JavaScript variables are used to store data

JS

Basic rules of JavaScript syntax

JS

Variable naming conventions

JS

Javascript data types

Javascript Operators

JS

Arithmetic operators

JS

Assignment operators

JS

Comparison operators

JS

Logical operators

JS

Conditional Statements

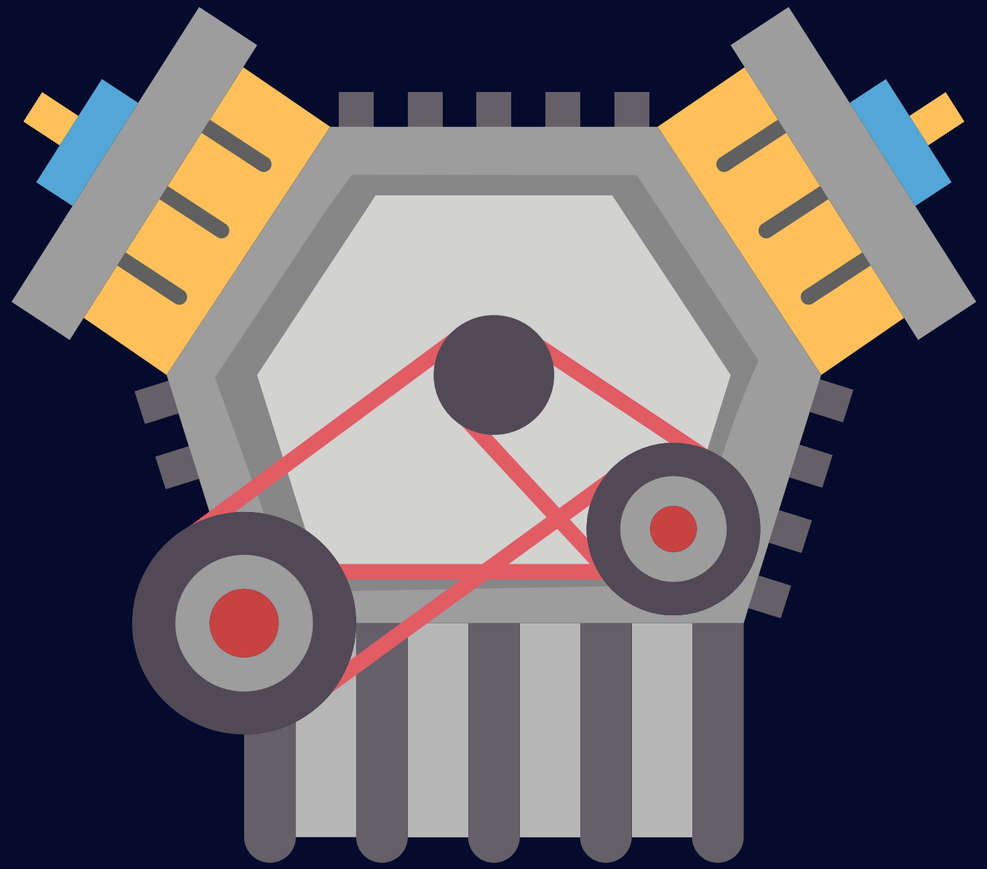
JS

Truthy and falsy values

JS

JavaScript Loops

JS



Javascript Functions

JS

Functions Syntax

function declaration

```
function fnName(parameter) {
    // code to be executed
}
```

JS

Functions Syntax

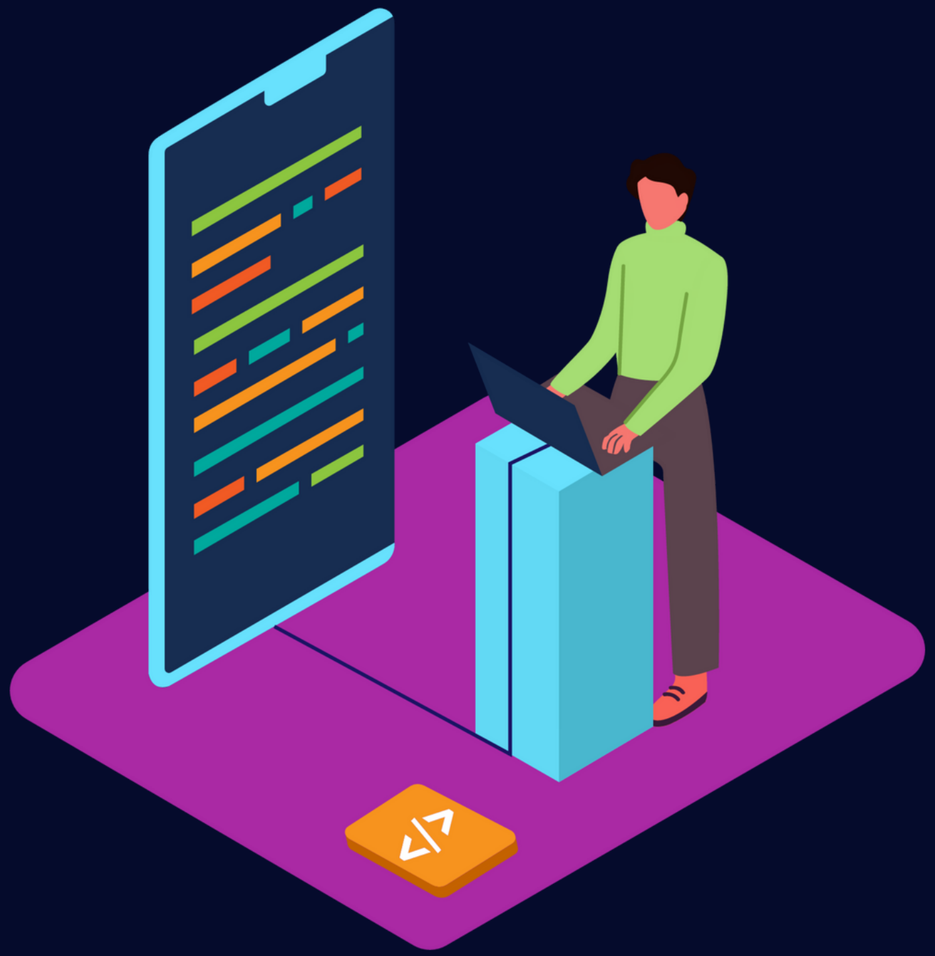
function Expression

```
let fnName = function(parameter) {
    // code to be executed
  }
```

JS

Javascript

Strings



```
1
2  const myString = "Hello World";
3
4  const myString2 = new String("Hello World");
5
```

JS

Javascript

ARRAYS

```
1  const myArr = new Array("css", "html", "nodejs");
2   //or
3  const myArr2 = ["css", "html", "nodejs"];
4
5
```

JS

```
1  const myArr = new Array("css", "html", "nodejs");
2   //or
3  const myArr2 = ["css", "html", "nodejs"];
4
5
```

index 0 index 1 index 2

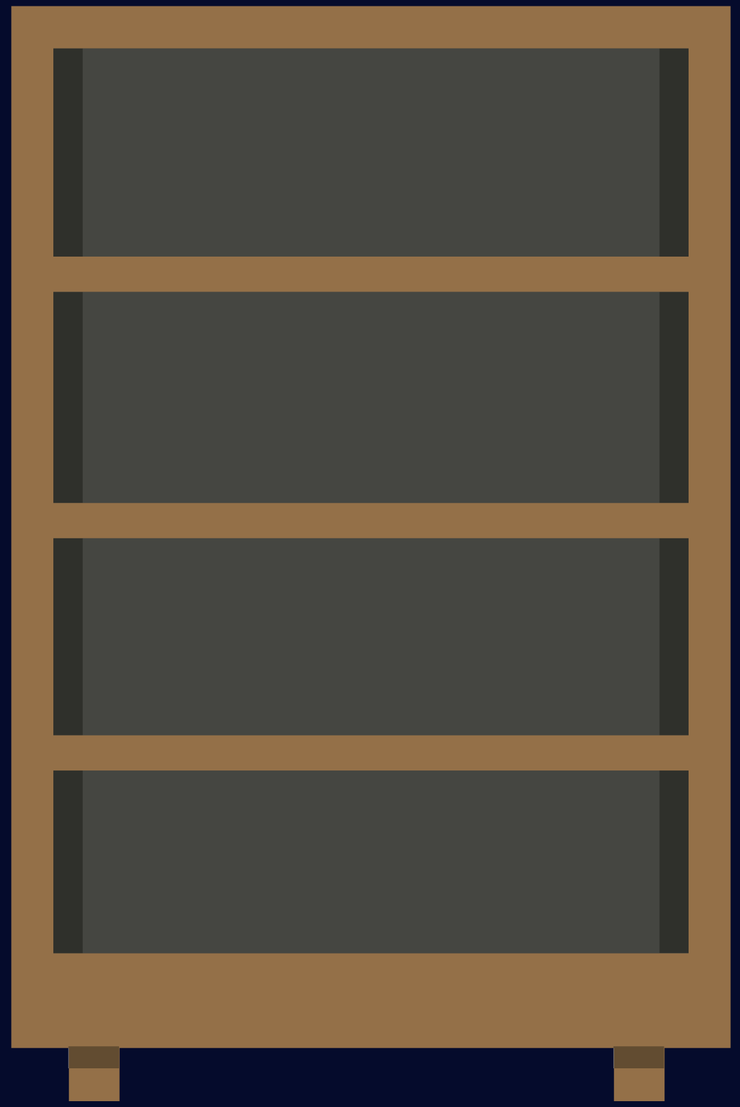
JS

I-NOVOTEK ACADEMY

ARRAYS OF BOOKS

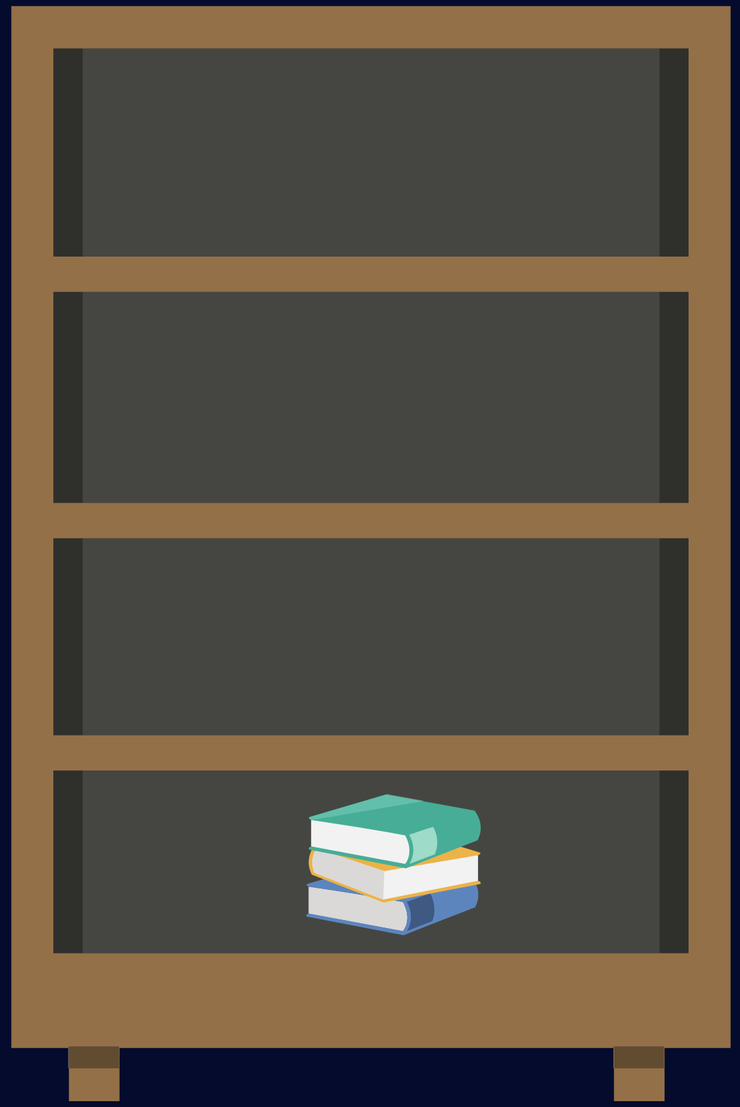
JS

ARRAYS OF BOOKS

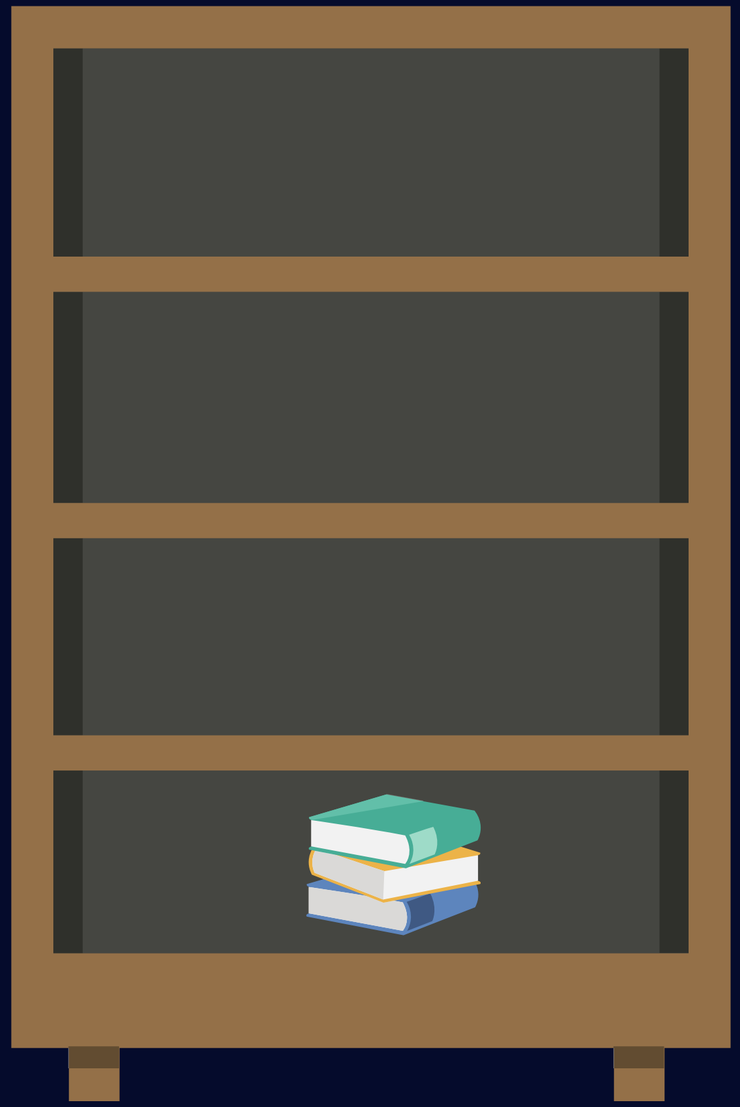


JS

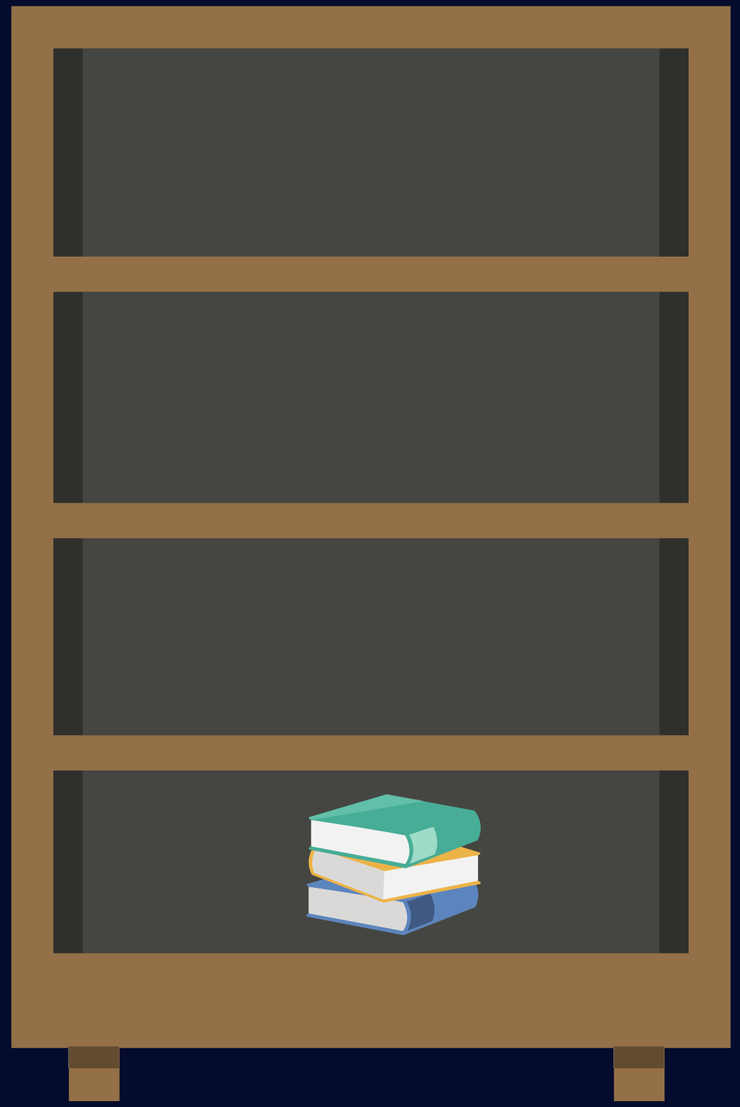
ARRAYS OF BOOKS



ARRAYS OF BOOKS



ARRAYS OF BOOKS



0

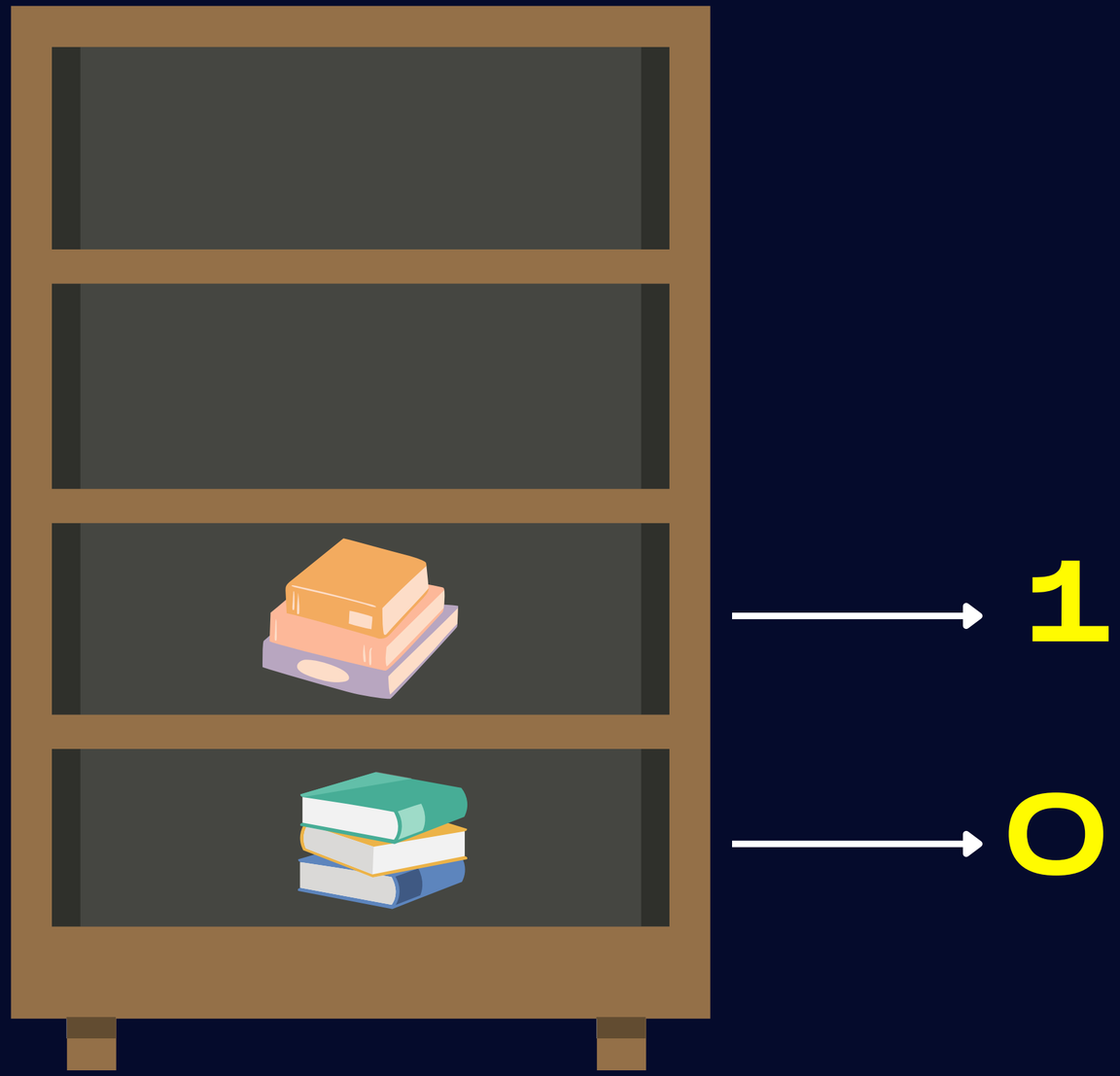
ARRAYS OF BOOKS

0

ARRAYS OF BOOKS

0

ARRAYS OF BOOKS



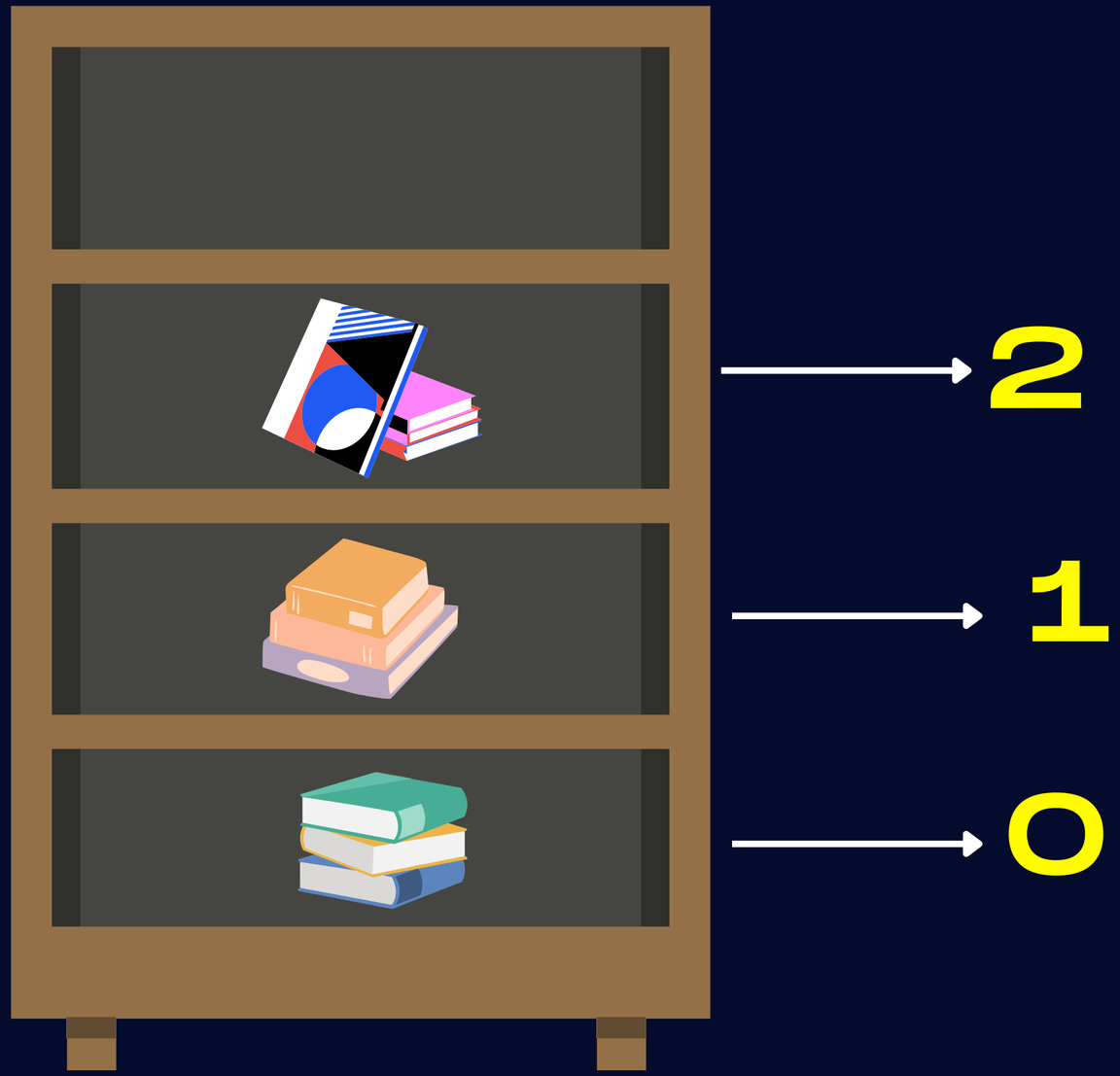
ARRAYS OF BOOKS

1

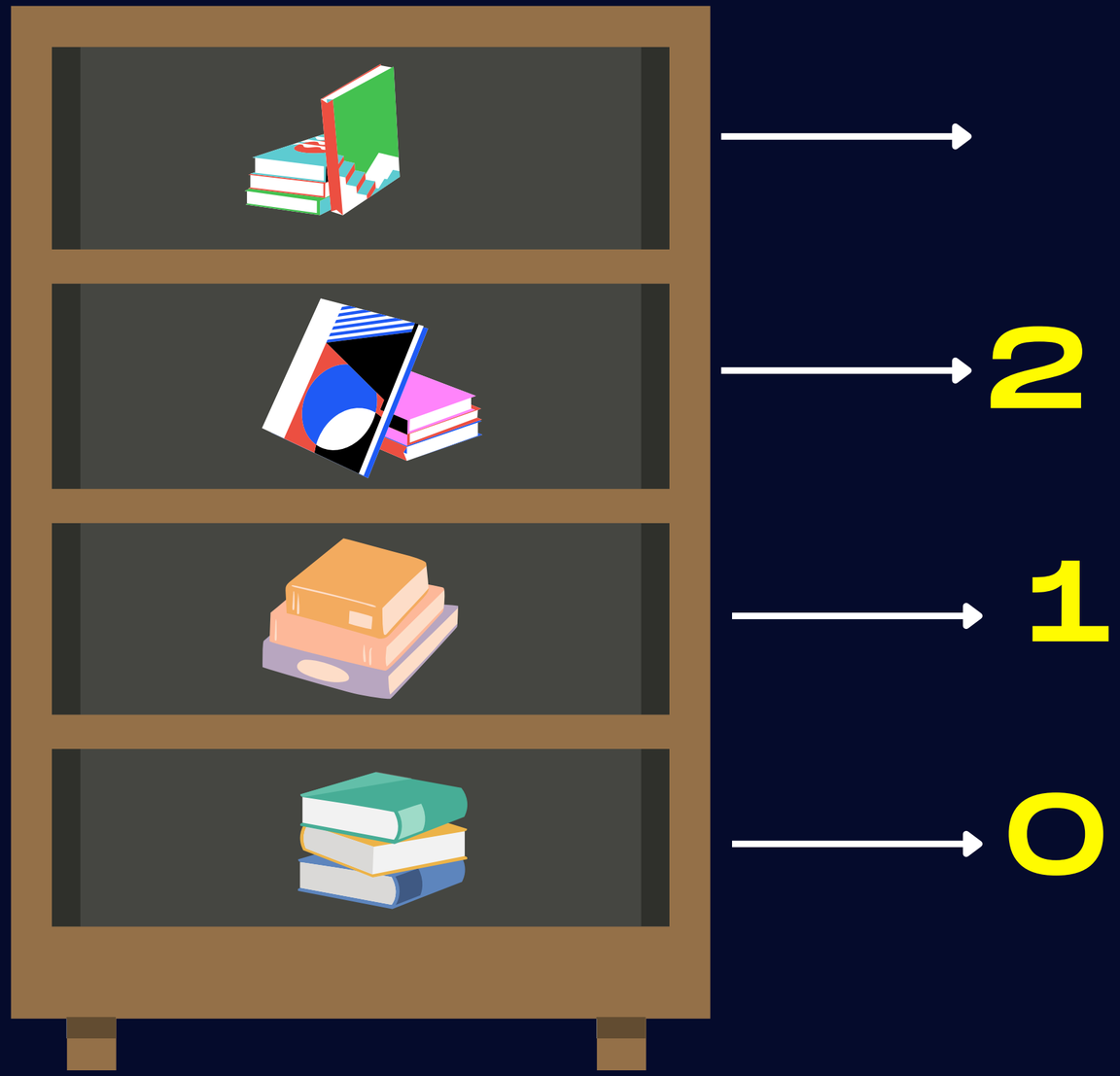
0

JS

ARRAYS OF BOOKS

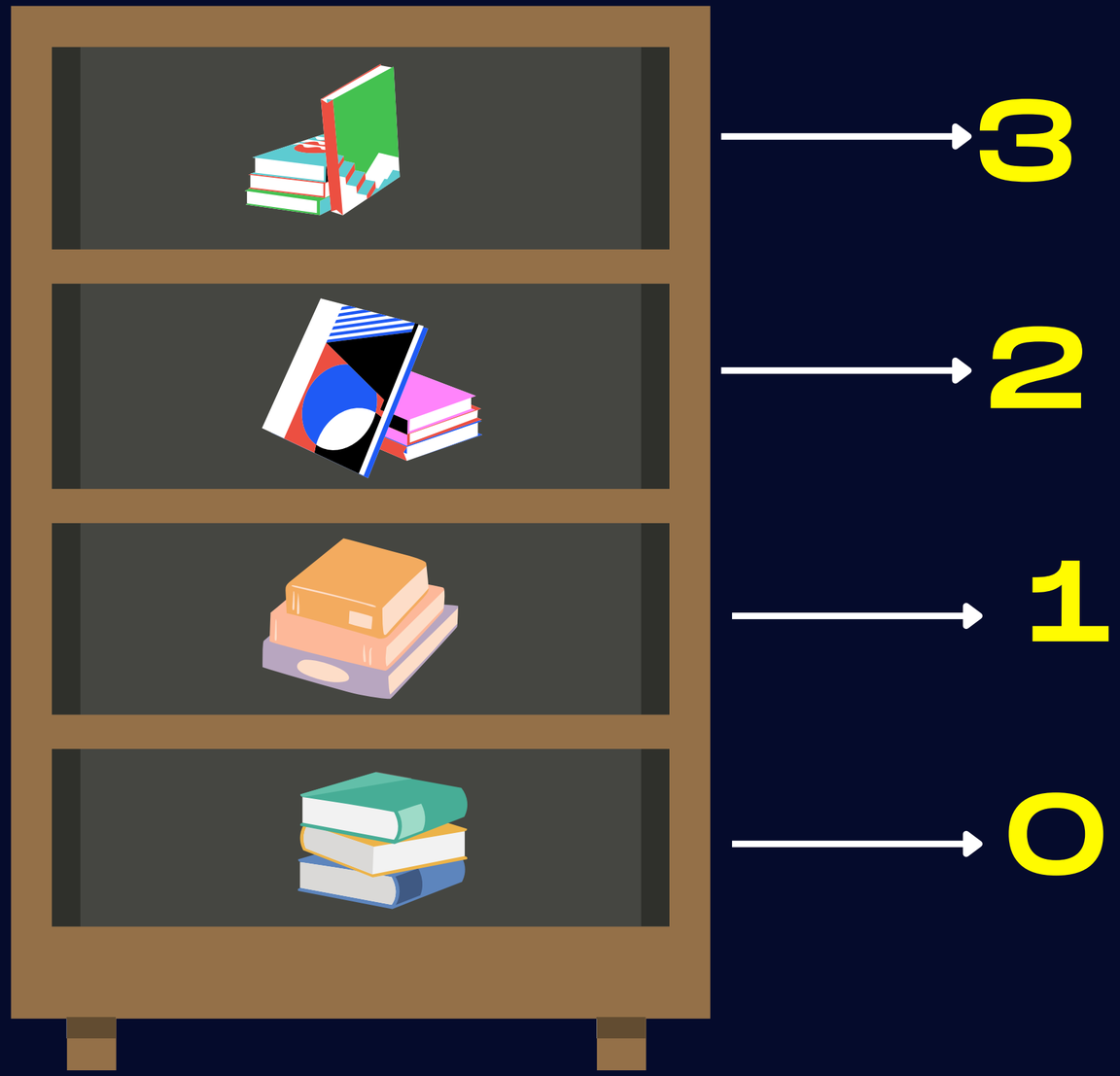


ARRAYS OF BOOKS

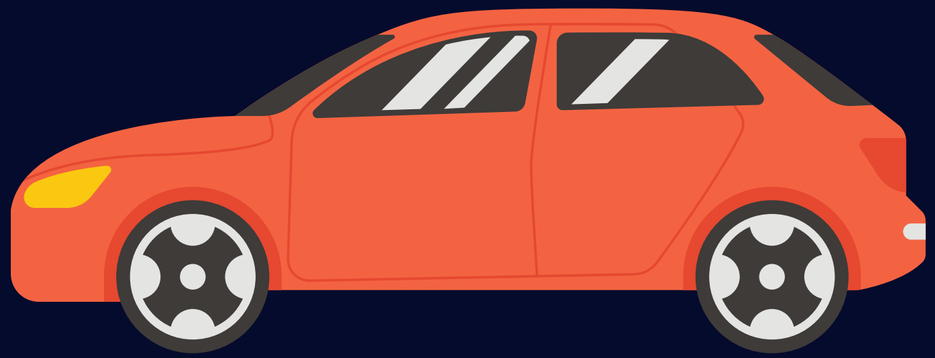


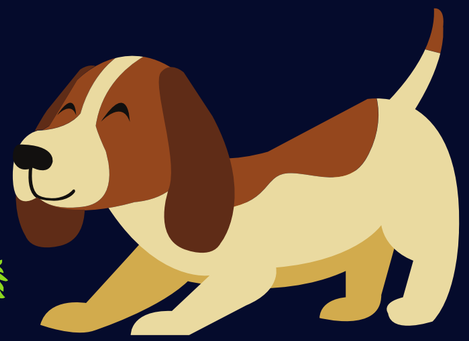
JS

ARRAYS OF BOOKS



JS





Javascript Objects

JS

Person Object

JS

{ } Syntax

JS

Object Name	Properties	Methods
	person.name = John person[name] = John person.age = 25 person[age] = 25	person.greet() person.walk() person.shout() person.run()

Javascript versions

Javascript versions

Javascript versions

JavaScript was created by Brendan Eich in 1995 and became an ECMA standard in 1997

Javascript versions

JavaScript was created by Brendan Eich in 1995 and became an ECMA standard in 1997

Organization

Javascript versions

The official name of the language is ECMAScript. ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6. Since 2016, new versions have been named by year (ECMAScript 2016 / 2017 / 2018).

Javascript versions

The official name of the language is ECMAScript. ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6. Since 2016, new versions have been named by year (ECMAScript 2016 / 2017 / 2018).

Javascript versions

The official name of the language is ECMAScript. ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6. Since 2016, new versions have been named by year (ECMAScript 2016 / 2017 / 2018).

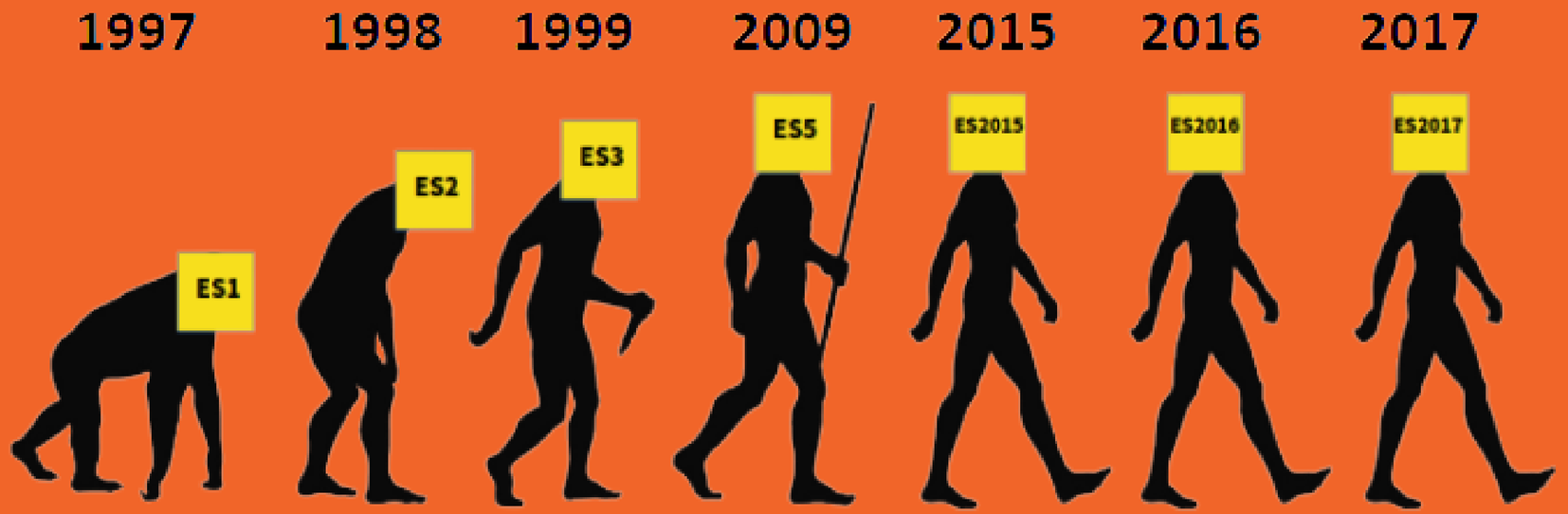
ECMA

Javascript versions

The official name of the language is ECMAScript. ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6. Since 2016, new versions have been named by year (ECMAScript 2016 / 2017 / 2018).

ECMA European Computer Manufacturers Association (ECMA)

Javascript versions



Javascript versions

Javascript versions

1997

Javascript versions

1997

ESI (ECMAScript 1) first version of JS language standard

Javascript versions

1997

2007

ESI (ECMAScript 1) first version of JS language standard

Javascript versions

1997

ESI (ECMAScript 1) first version of JS language standard

2007

ES5 (ECMAScript 5) New more features released

Javascript versions

1997

ESI (ECMAScript 1) first version of JS language standard

2007

ES5 (ECMAScript 5) New more features released

2015

ES6 (ECMAScript 5) Biggest update

Javascript versions

1997

ESI (ECMAScript 1) first version of JS language standard

2007

ES5 (ECMAScript 5) New more features released

2015

ES6 (ECMAScript 5) Biggest update

2016

Javascript versions

1997

ESI (ECMAScript 1) first version of JS language standard

2007

ES5 (ECMAScript 5) New more features released

2015

ES6 (ECMAScript 5) Biggest update

2016

2017

Javascript versions

ES6/2015

DESTRUCTURING

ES6/2015

REST PARAMETERS (...)

ES6/2015

Rest parameters are a way to pass an arbitrary number of arguments to a function

JS

ARROW FUNCTION

ES6/2015

```
() =>{}
```

ARROW FUNCTION

ES6/2015

An arrow function expression is a more concise way to write a traditional function expression, but it cannot be used in all situations.

JS

ARROW FUNCTION

ES6/2015

ES6/2015

Arrow function limitations

ARROW FUNCTION

ES6/2015

- Arrow functions do not have this keyword.

ARROW FUNCTION

ES6/2015

- Arrow functions do not have this keyword.
- Arrow functions do not have arguments.

ARROW FUNCTION

ES6/2015

- Arrow functions do not have this keyword.
- Arrow functions do not have arguments.
- Arrow functions cannot use as a constructor.

SPREAD OPERATOR (...)

The spread operator (...) is a convenient way to copy all or part of an existing array or object into another array or object.

ES6/2015

Spread vs Rest

Spread syntax "expands" an array into its elements, while rest syntax collects multiple elements and "condenses" them into a single element

Math Object

JS

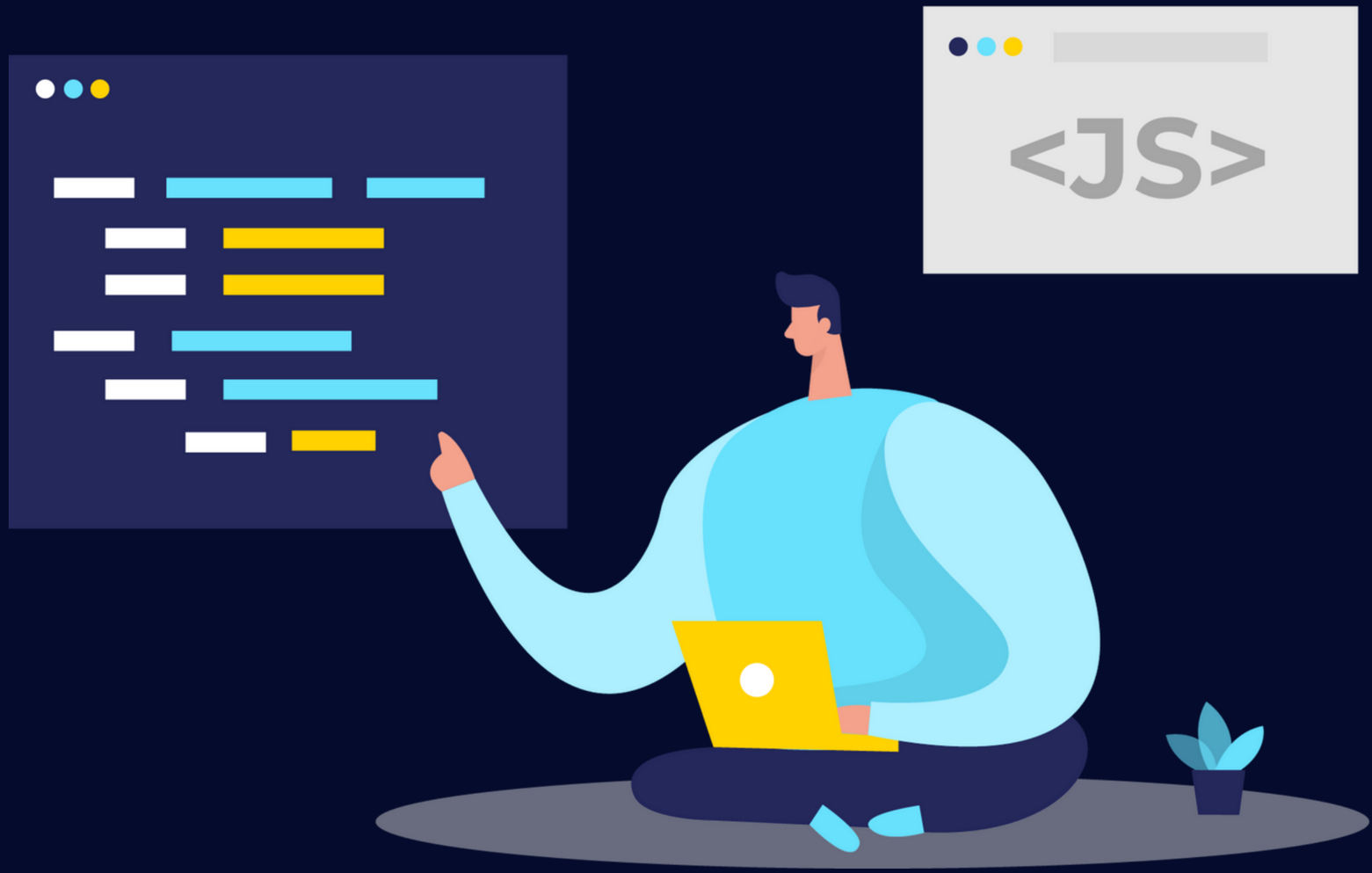
Number Methods

Number Methods

Primitive values cannot have properties and methods, but JavaScript treats primitive values as objects when executing methods and properties. This allows for methods and properties to be available to primitive values.

JS

Advanced Functions



JS

Functions

Scope

=

Variable and function visibilities

JS

I-NOVOTEK ACADEMY

JS

```
let number = 10;

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];
  }
}

```

JS

Global Scope

```
let number = 10;

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];
  }
}
```

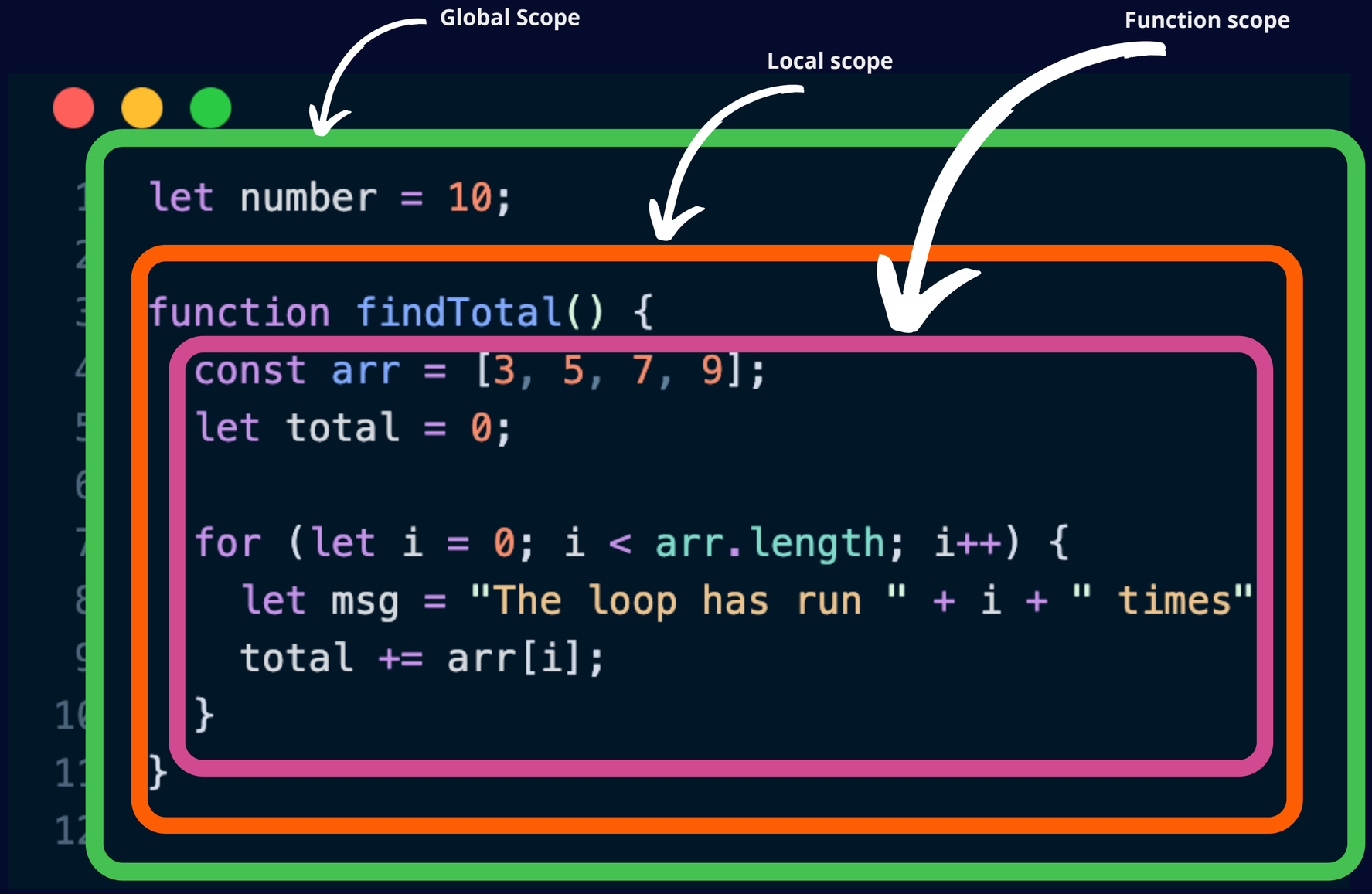
JS

Global Scope

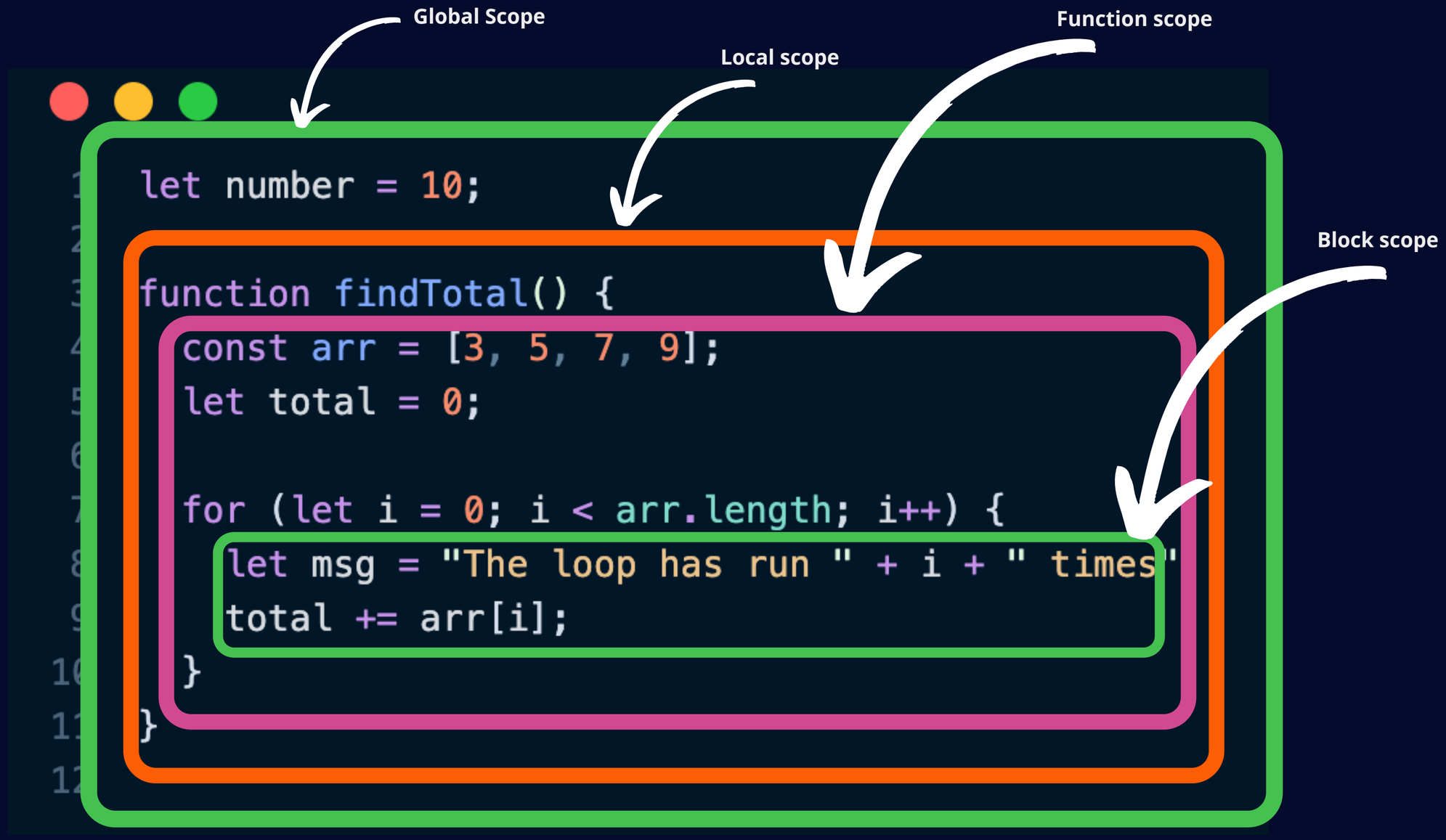
Local scope

```
let number = 10;

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];
  }
}
```

JS


Global Scope
Local scope
Function scope
let number = 10;
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];
}
}

JS


Global Scope
Local scope
Function scope
Block scope
let number = 10;
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];
}
}

JS

HISTORY OF JAVASCRIPT

HISTORY OF JAVASCRIPT

Brendan Eich

HISTORY OF JAVASCRIPT

- JavaScript was created in 1995 by Brendan Eich

Brendan Eich

HISTORY OF JAVASCRIPT

- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.

Brendan Eich

HISTORY OF JAVASCRIPT

Brendan Eich

- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.
- In 1997, JavaScript was standardised by in the ECMAScript language specification.

HISTORY OF JAVASCRIPT

Brendan Eich

- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.
- In 1997, JavaScript was standardised by in the ECMAScript language specification.
- In the early 2000s, Ajax were developed that use JavaScript to create dynamic applications

HISTORY OF JAVASCRIPT

Brendan Eich

- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.
- In 1997, JavaScript was standardised by in the ECMAScript language specification.
- In the early 2000s, Ajax were developed that use JavaScript to create dynamic applications
- JavaScript's syntax is heavily inspired by C++ and Java.

HISTORY OF JAVASCRIPT

Brendan Eich

- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.
- In 1997, JavaScript was standardised by in the ECMAScript language specification.
- In the early 2000s, Ajax were developed that use JavaScript to create dynamic applications
- JavaScript's syntax is heavily inspired by C++ and Java.
- JavaScript is an interpreted language, not a compiled language

HISTORY OF JAVASCRIPT

Brendan Eich

- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.
- In 1997, JavaScript was standardised by in the ECMAScript language specification.
- In the early 2000s, Ajax were developed that use JavaScript to create dynamic applications
- JavaScript's syntax is heavily inspired by C++ and Java.
- JavaScript is an interpreted language, not a compiled language
- JavaScript is named after Java, and many of its concepts are borrowed from the Java language

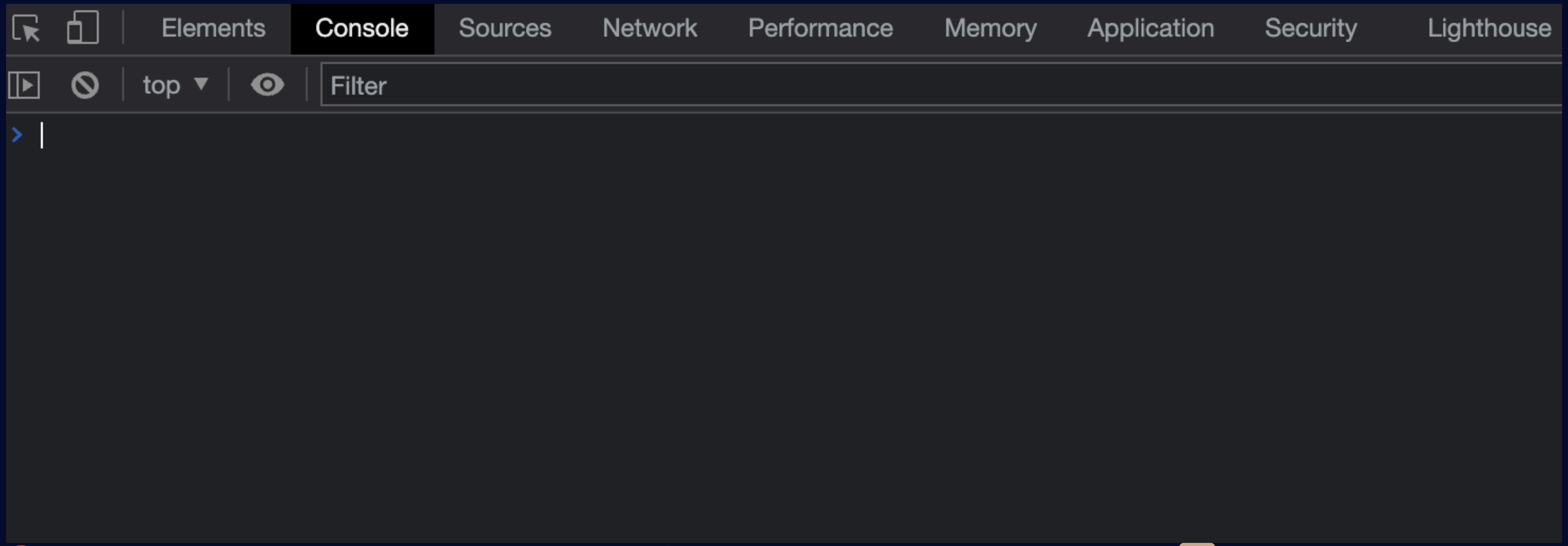
HISTORY OF JAVASCRIPT

Brendan Eich

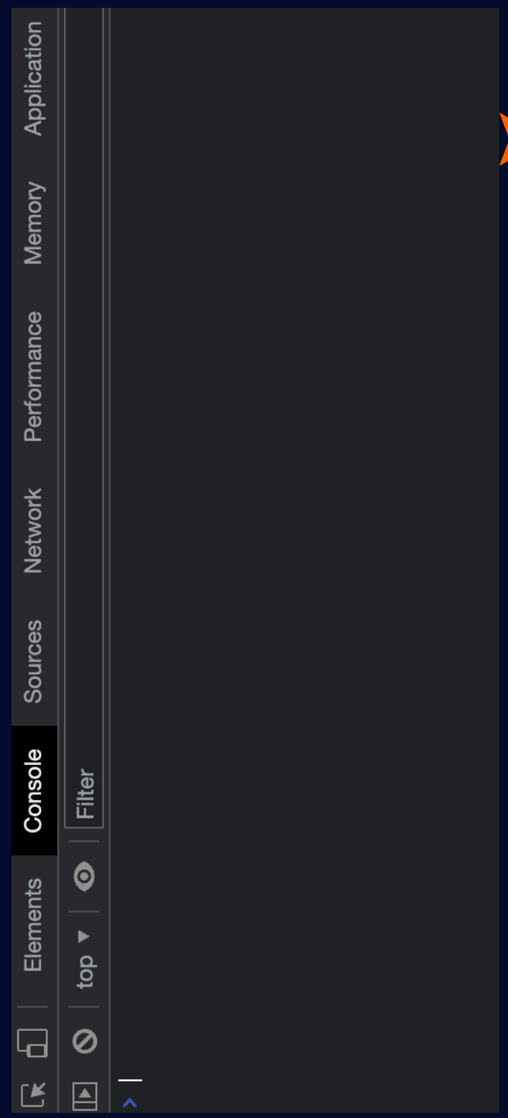
- JavaScript was created in 1995 by Brendan Eich
- It was originally named Mocha -> LiveScript ---> JavaScript.
- In 1997, JavaScript was standardised by in the ECMAScript language specification.
- In the early 2000s, Ajax were developed that use JavaScript to create dynamic applications
- JavaScript's syntax is heavily inspired by C++ and Java.
- JavaScript is an interpreted language, not a compiled language
- JavaScript is named after Java, and many of its concepts are borrowed from the Java language

JS

console.log()

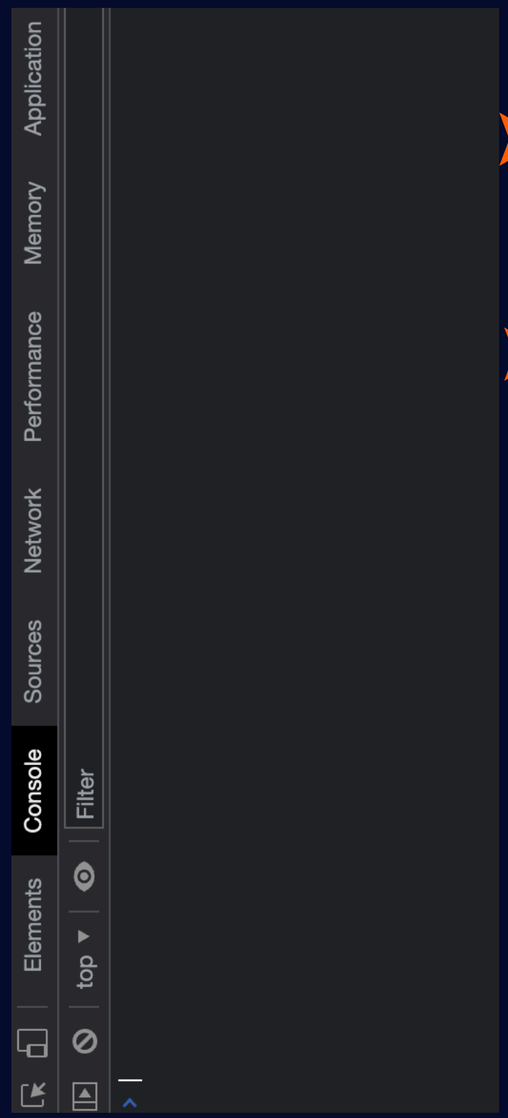


console.log()



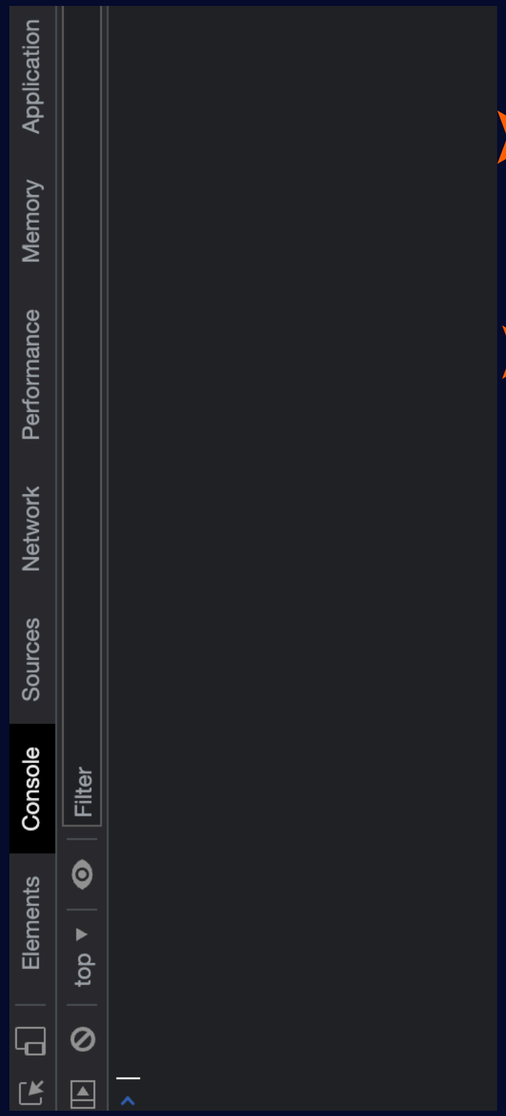
The console.log() is a function in JavaScript which is used to print any kind of variables

console.log()



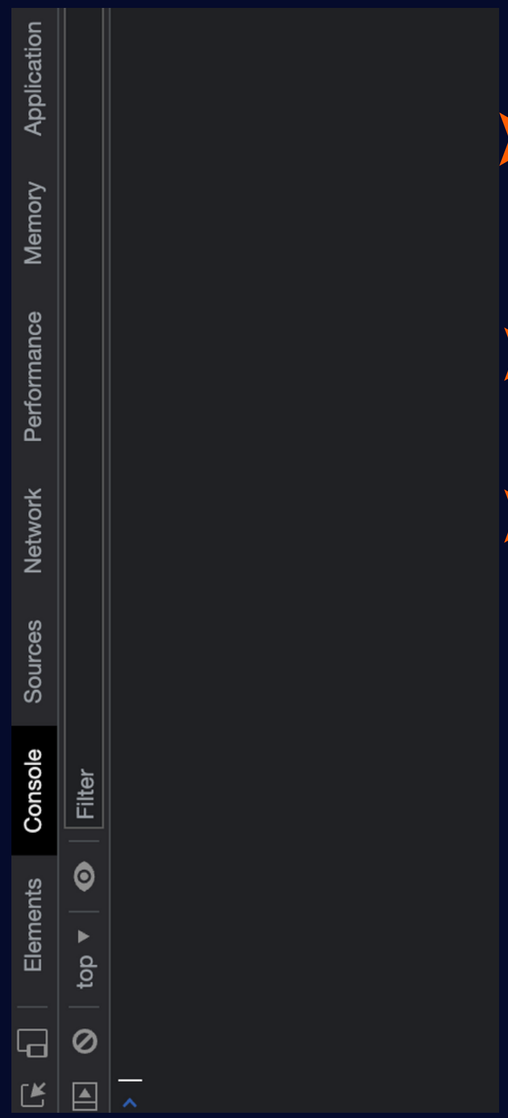
- The console.log() is a function in JavaScript which is used to print any kind of variables
- It is like a command-line interface that runs JavaScript on your JavaScript engine.

console.log()



- The console.log() is a function in JavaScript which is used to print any kind of variables
- It is like a command-line interface that runs JavaScript on your JavaScript engine.

console.log()



- The console.log() is a function in JavaScript which is used to print any kind of variables
- It is like a command-line interface that runs JavaScript on your JavaScript engine.
- read-eval-print loop (REPL). This refers to the loop that the console runs

Features of javascript

Features of javascript

It's a single threaded language

Features of javascript

It's a single threaded language

It's a dynamically typed language

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

This means that js executes the code line by line.

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

This means that the type of a variable can change during the program execution

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

It's a programming paradigm that uses objects as the primary way of representing data which uses Prototypal Inheritance

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

It's a process that frees up memory by removing objects that are no longer used in the program and it's called garbage collection.

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

Using javascript in the browser

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

Using javascript on the server

Features of javascript

- It's a single threaded language
- It's a dynamically typed language
- object-oriented language
- It's a garbage collector
- It's a client-side language
- It's a server-side language
- It's a non-blocking
- It's a high-level language

It's a code that runs concurrently in the background with the help of event loop

Features of javascript

High-level language

For high level programming languages like JavaScript, Python, Ruby, etc. resources management are done automatically

Features of javascript

High-level language

multi paradigm language

This a coding approach to structure code

1. Functional programming
2. Object oriented programming
3. Procedural programming

V8 Engine

V8 Engine

V8 Engine

V8 Engine

V8 Engine

Javascript engine is a computer program that runs javascript code

V8 Engine

V8 Engine

Javascript engine is a computer program that runs javascript code

Computers only understand machine code.

V8 Engine

V8 Engine

Javascript engine is a computer program that runs javascript code

Computers only understand machine code.

```
101001100010111
000111010111001
011010101001010
110110110101010
101010101001010
```

Code execution process

V8 Engine

Code execution process

source code

V8 Engine

Code execution process

source code

V8 Engine

Code execution process

source code

V8 Engine

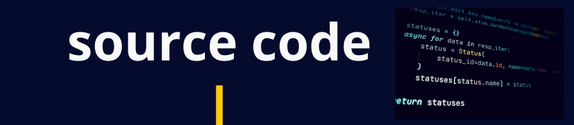
Code execution process

source code

AST (abstract syntax tree)

V8 Engine

Code execution process



source code

parser

AST (abstract syntax tree)

V8 Engine

Code execution process

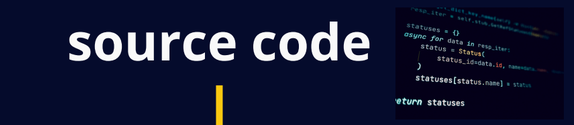
source code

parser

AST (abstract syntax tree)

V8 Engine

Code execution process



source code

parser

AST (abstract syntax tree)

machine code

V8 Engine

Code execution process

V8 Engine

source code

parser

AST (abstract syntax tree)

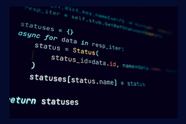
Compilation

machine code

Code execution process

V8 Engine

source code



parser

AST (abstract syntax tree)

Compilation

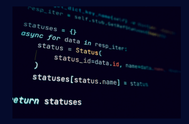
machine code

browser

Code execution process

V8 Engine

source code



parser

AST (abstract syntax tree)

Compilation

machine code

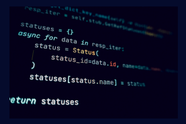
interpreter

browser

Code execution process

V8 Engine

source code



parser

AST (abstract syntax tree)

Compilation

machine code

interpreter

browser

JS Runtime (Browser)

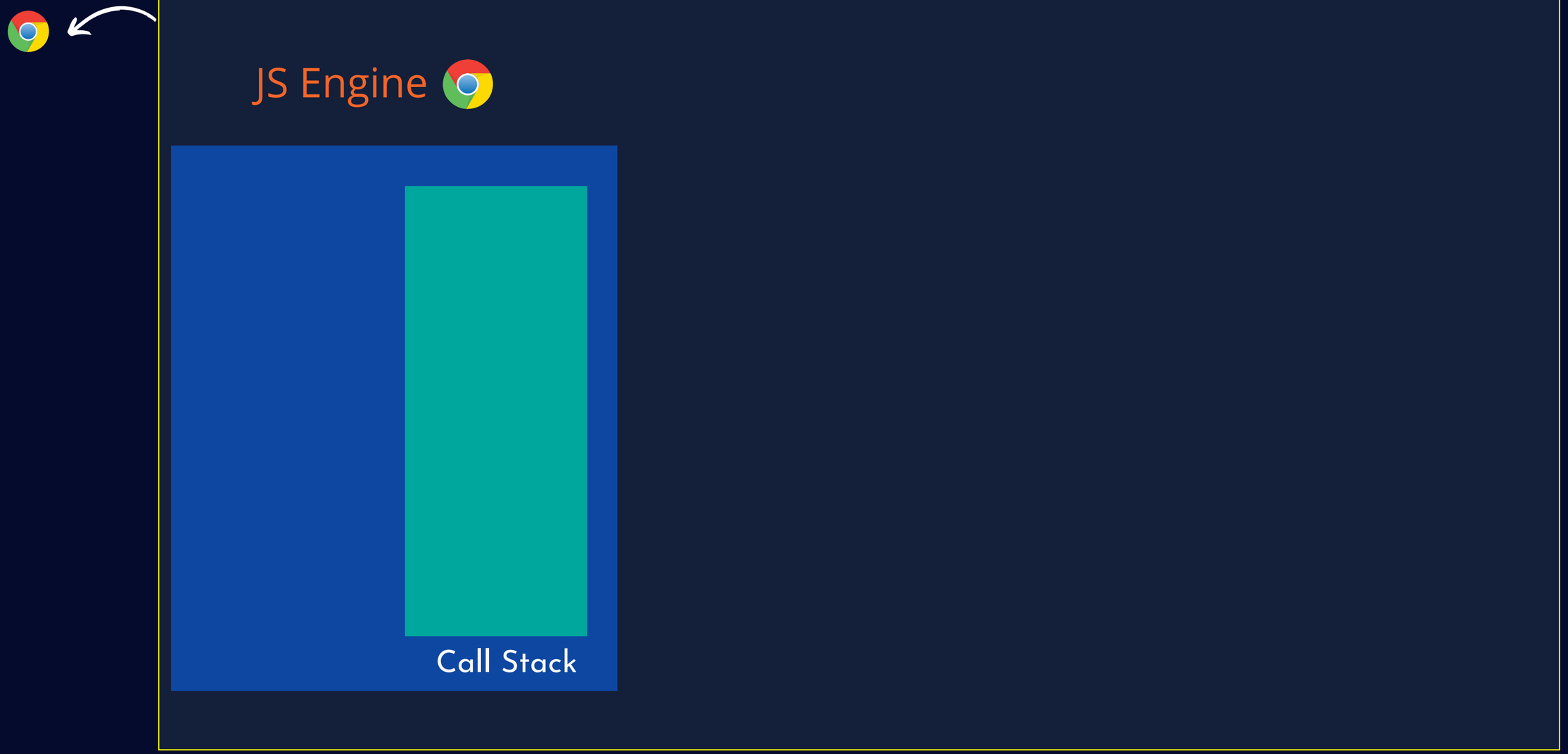
JS runtime, in a layman's point of view, it's a box that contains all the things we need to run our code.

JS Runtime (Browser)

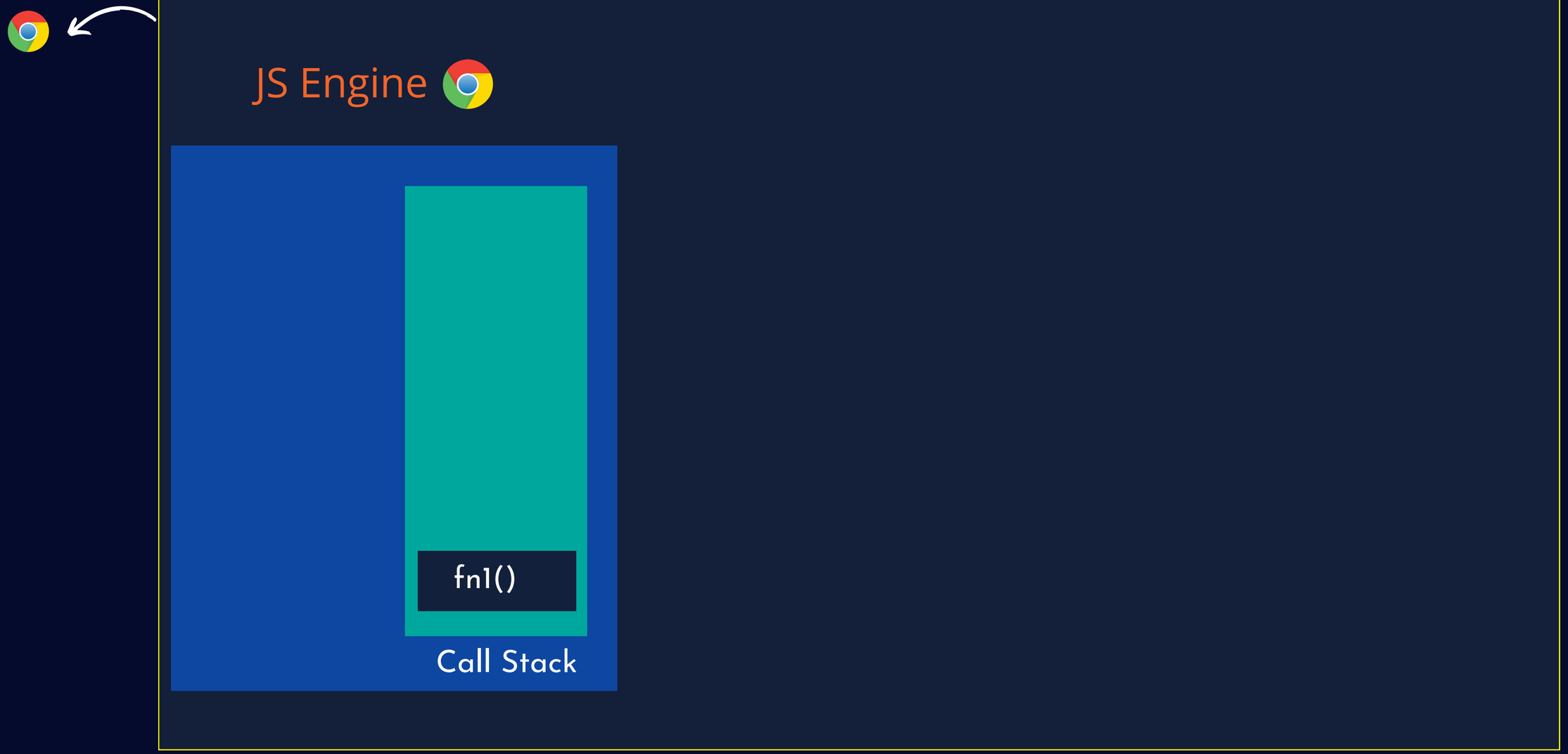
JS Runtime (Browser)

JS Engine

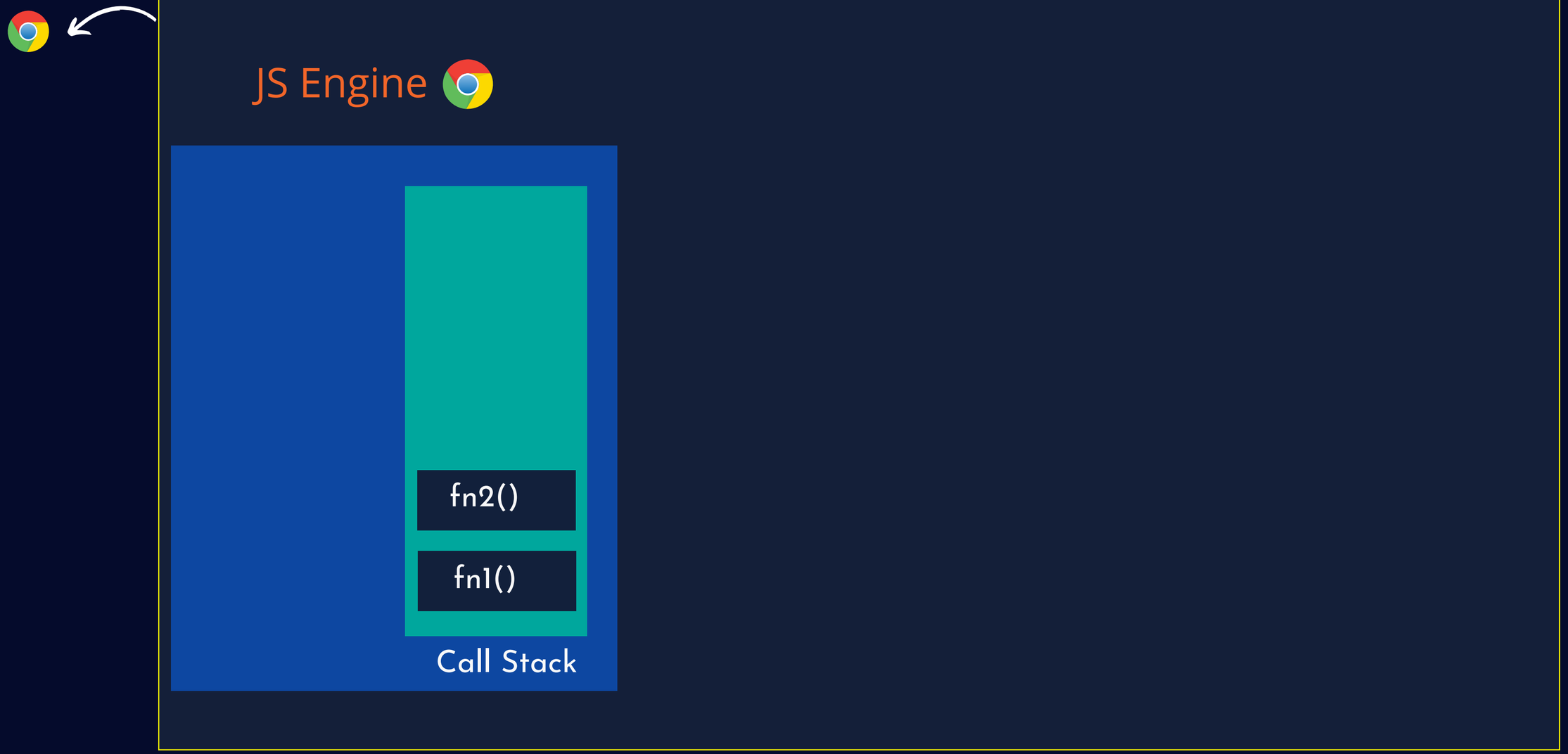
JS Runtime (Browser)



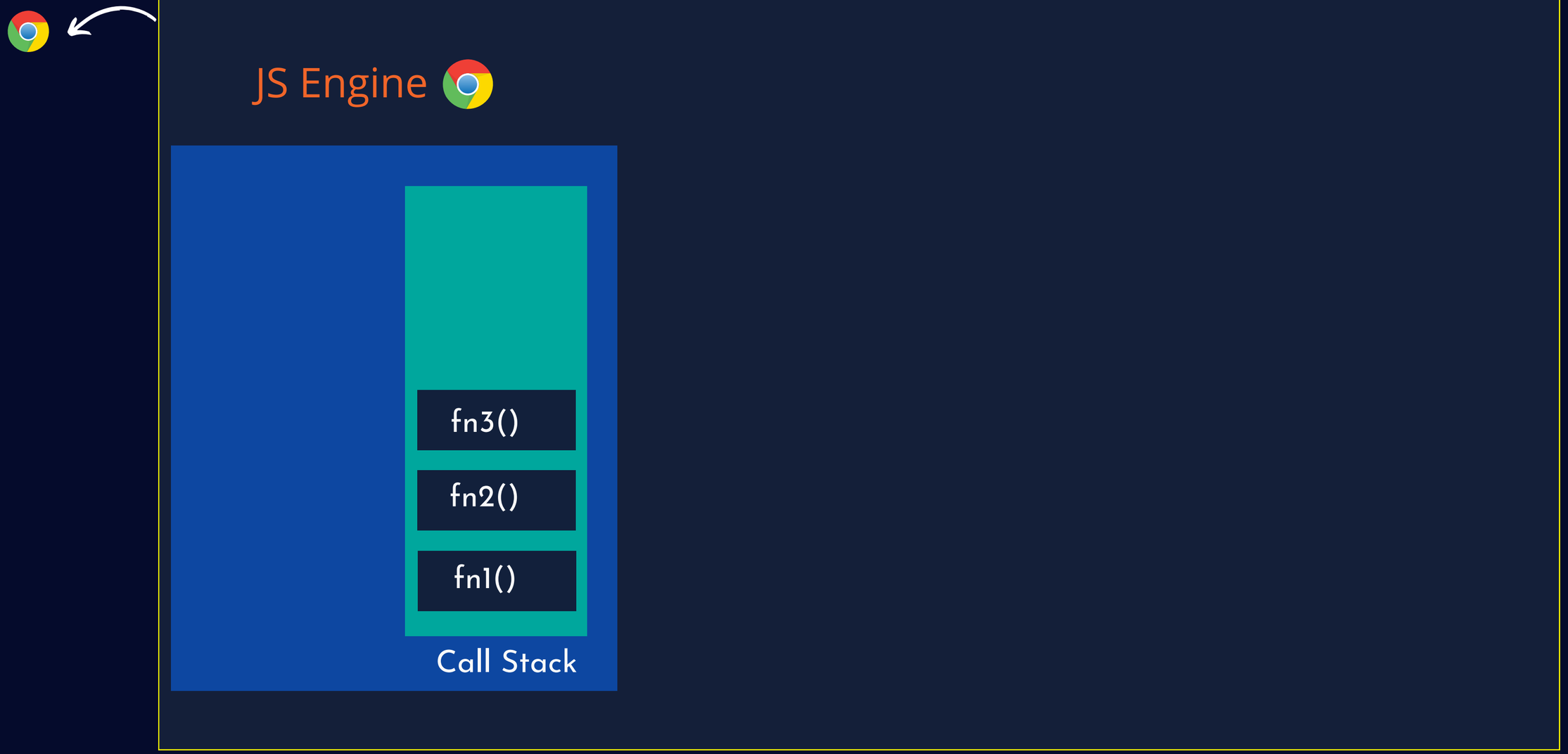
JS Runtime (Browser)



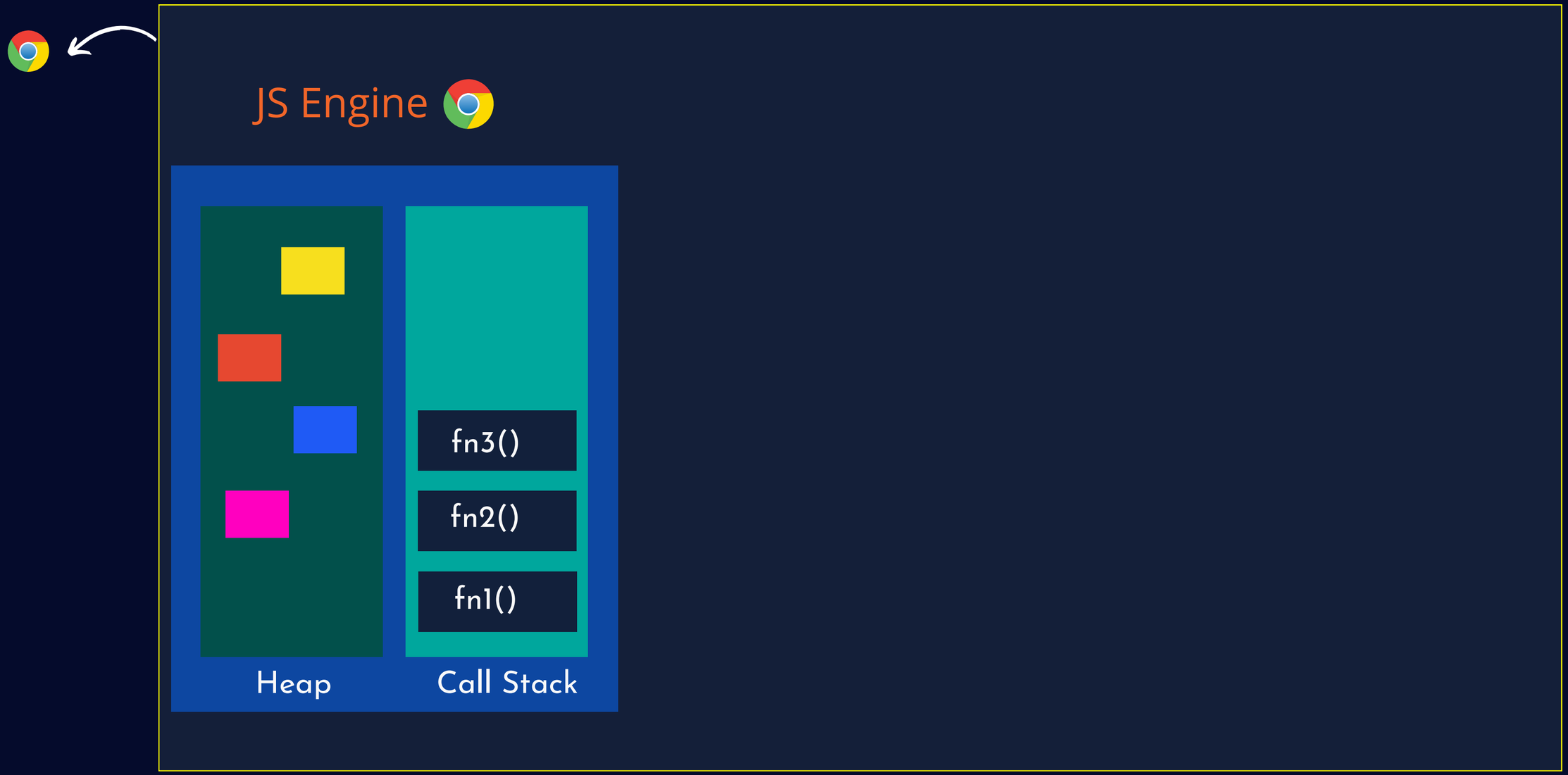
JS Runtime (Browser)



JS Runtime (Browser)

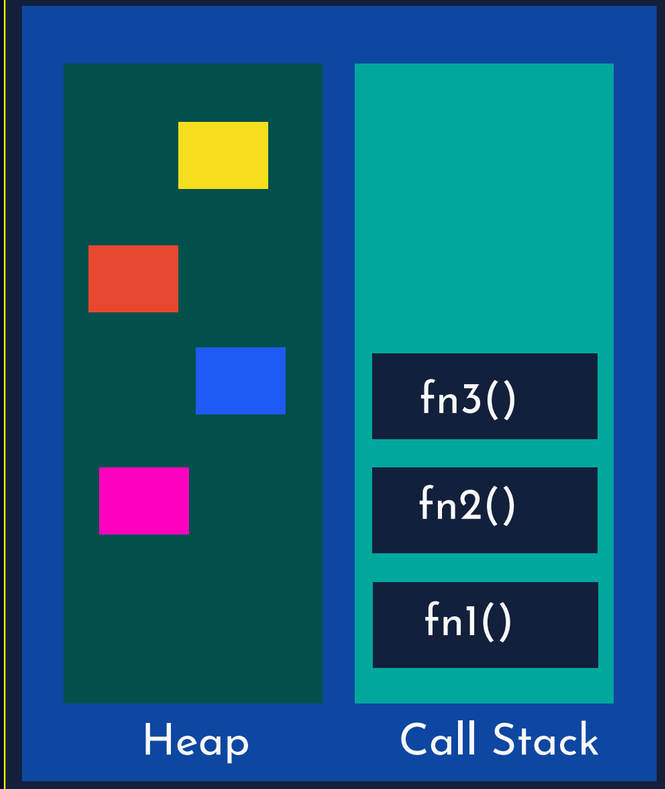


JS Runtime (Browser)

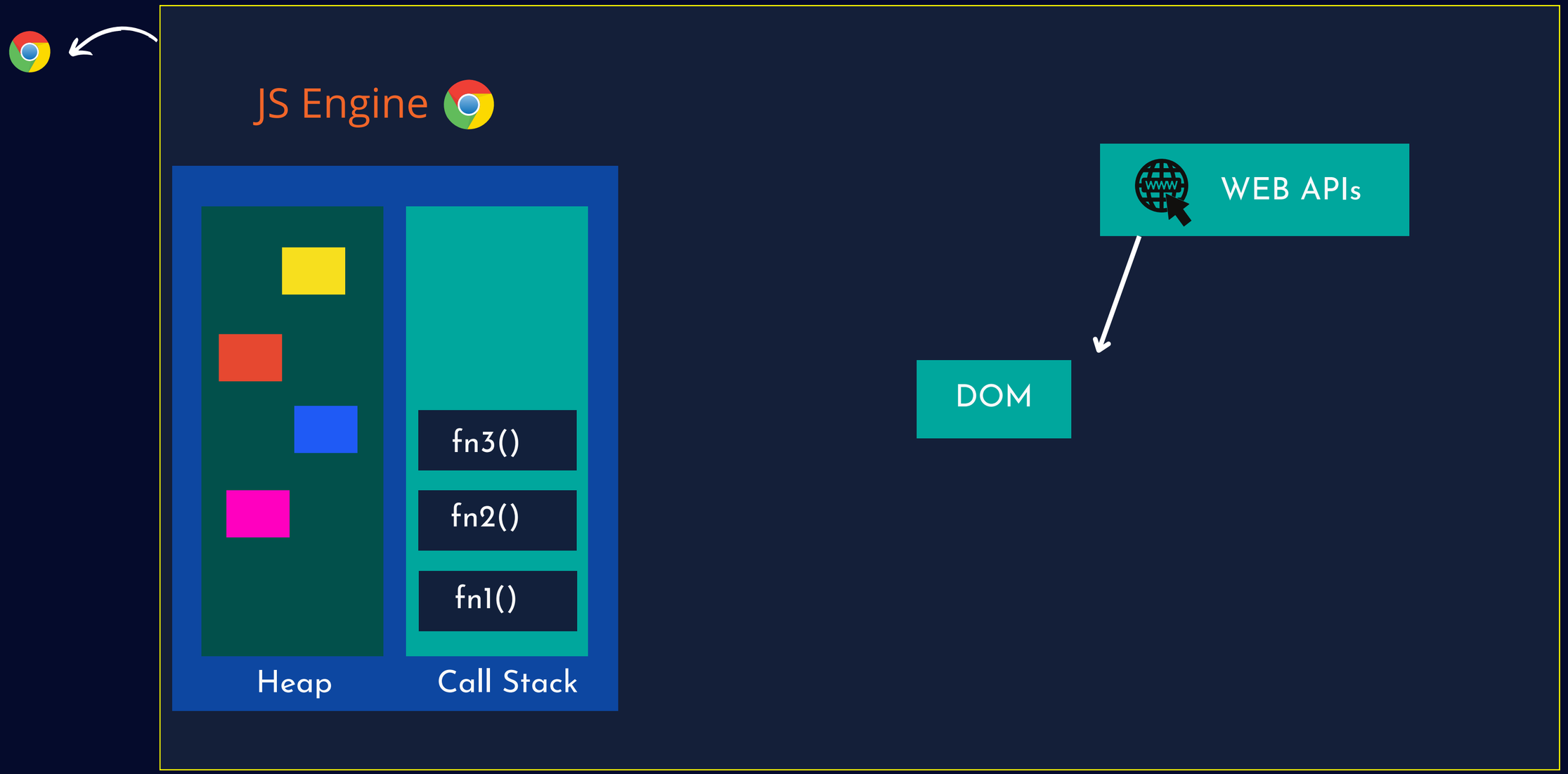


JS Runtime (Browser)

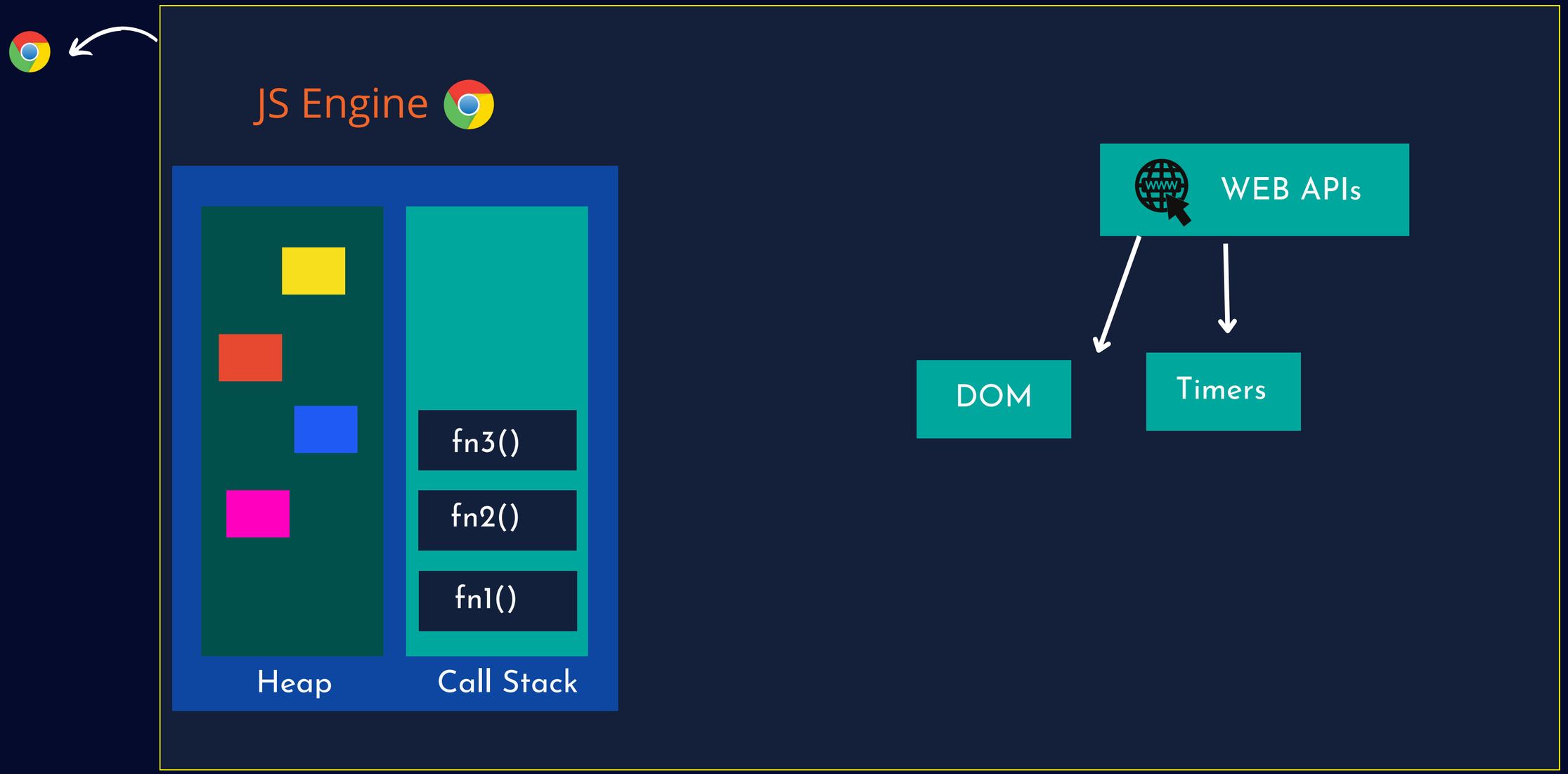
JS Engine



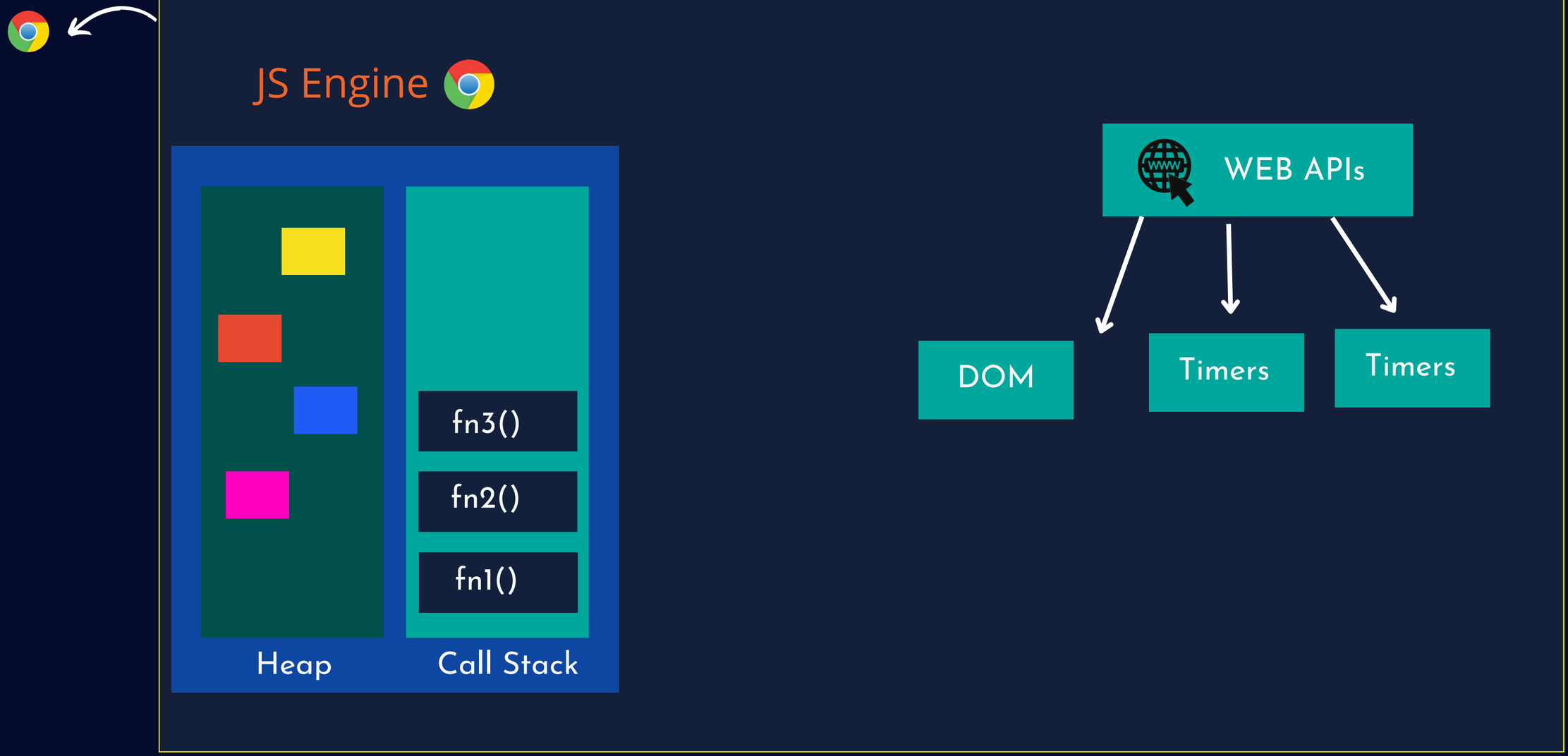
JS Runtime (Browser)



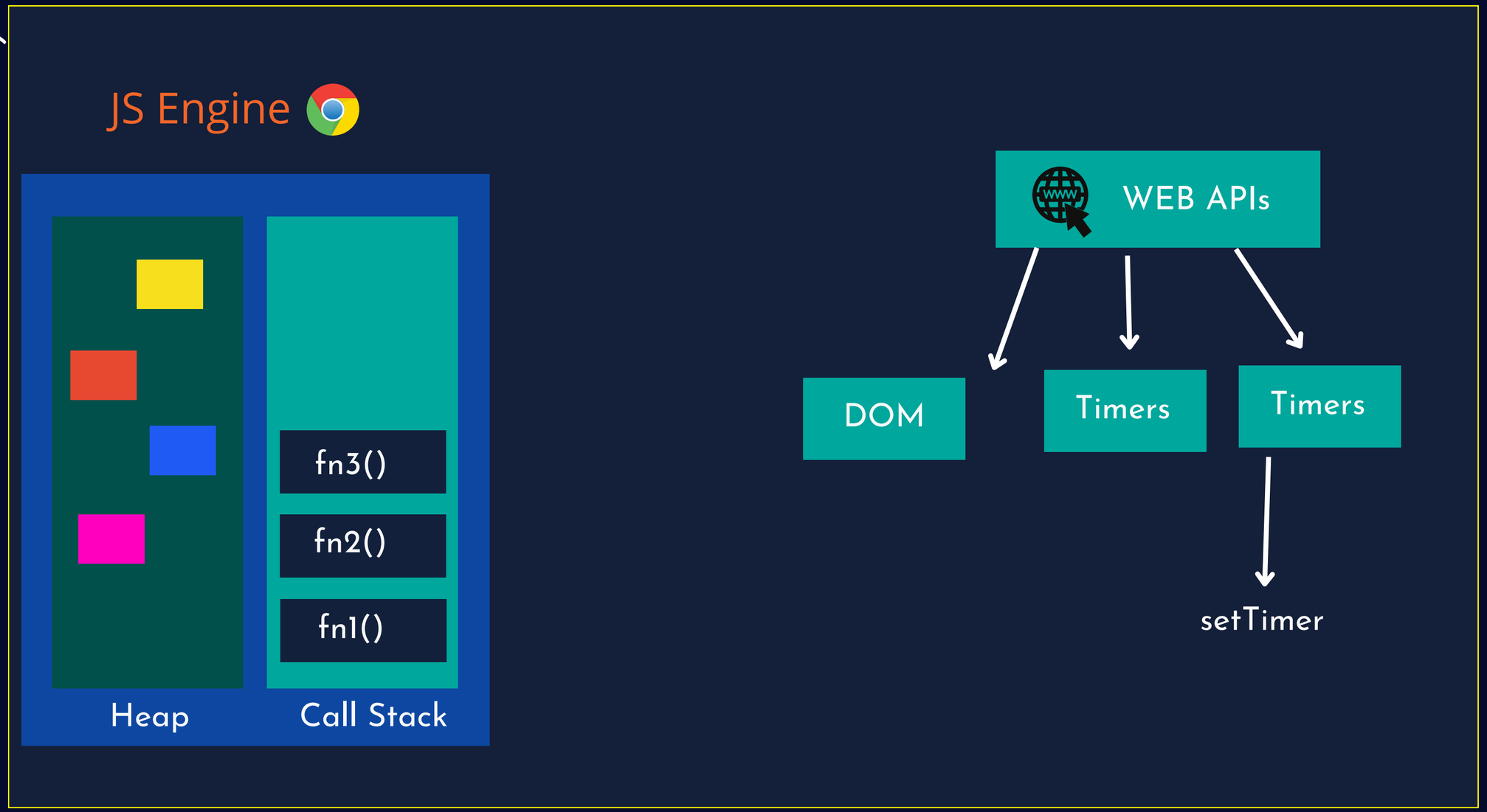
JS Runtime (Browser)



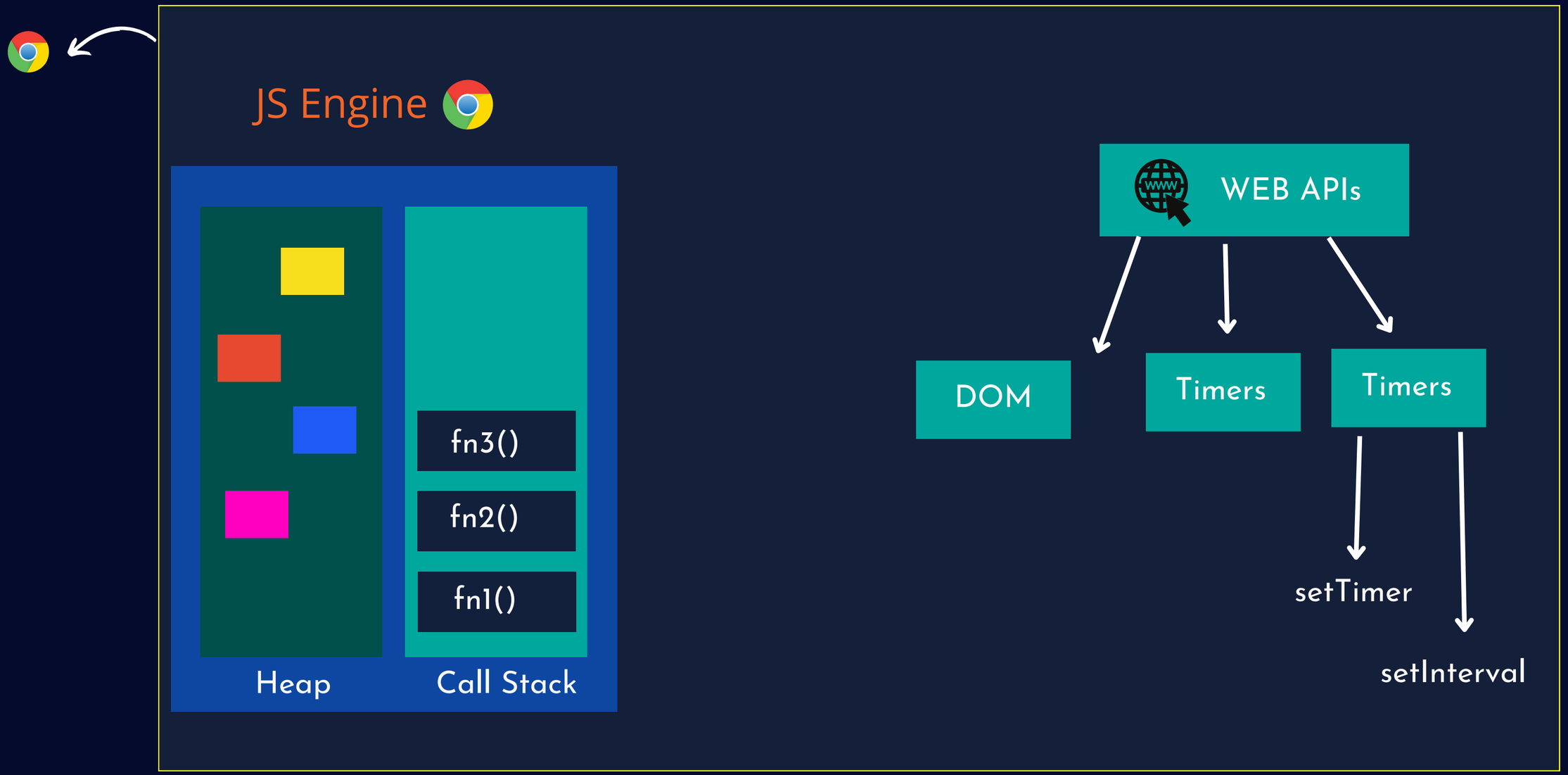
JS Runtime (Browser)



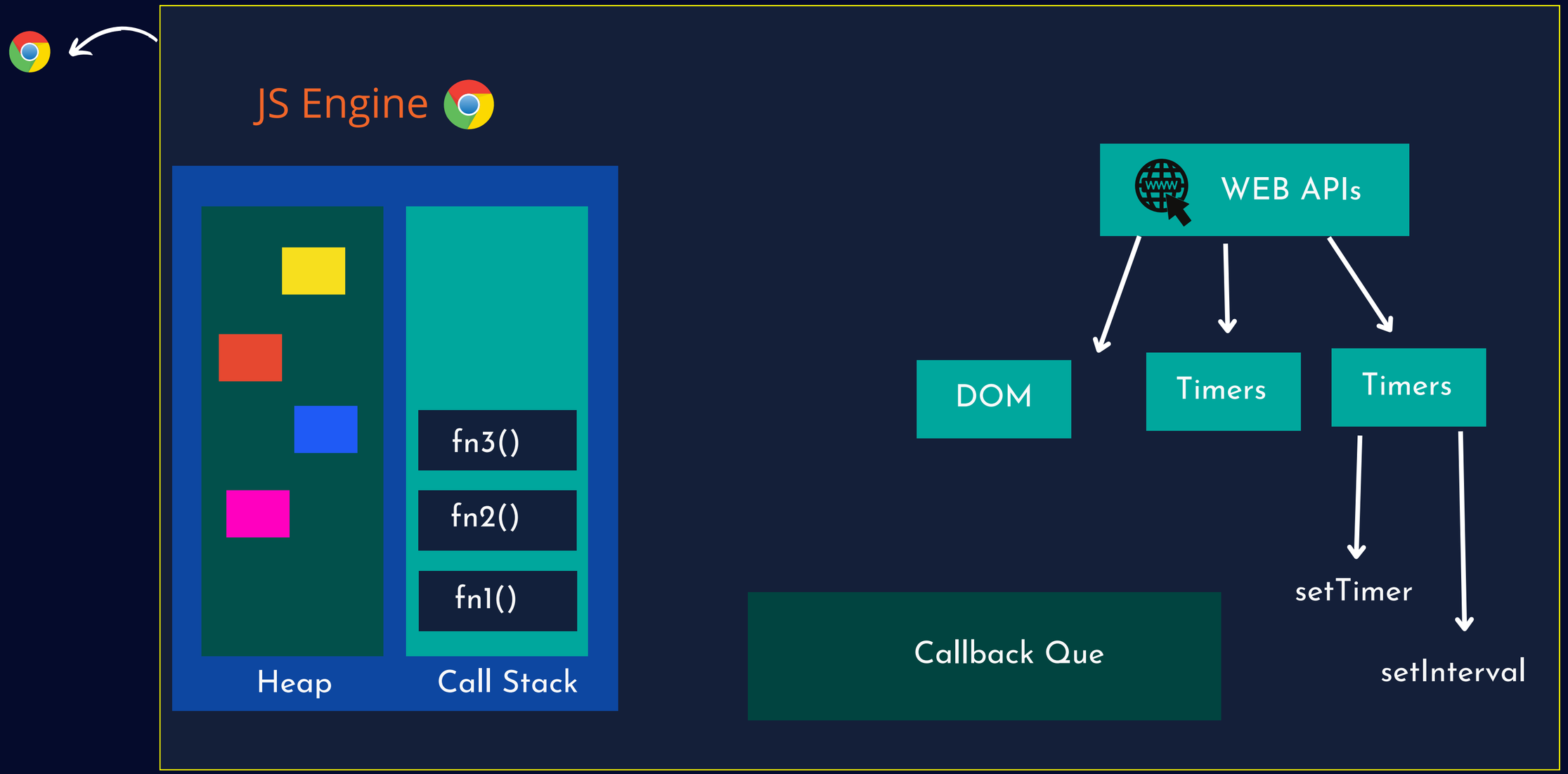
JS Runtime (Browser)



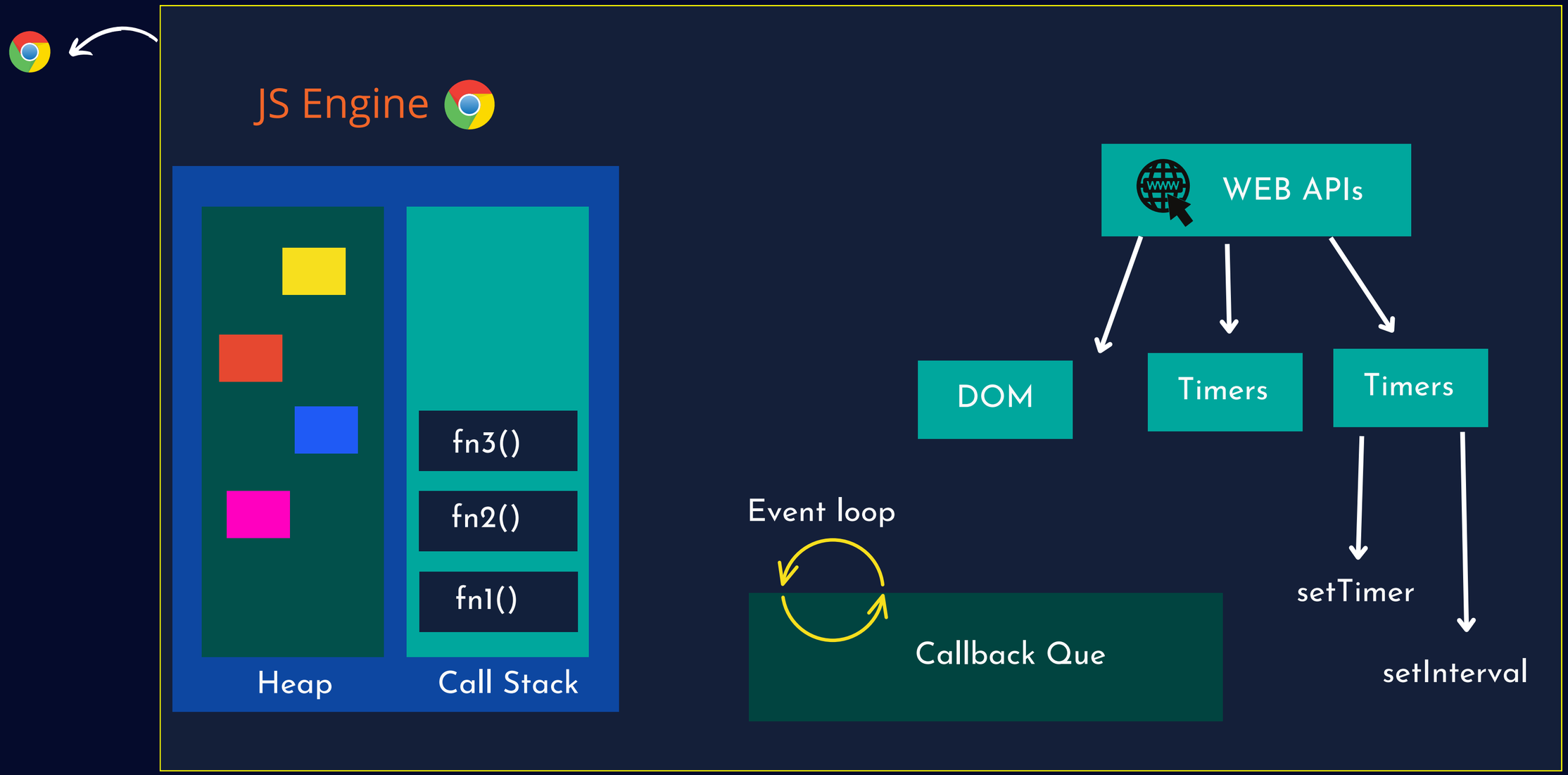
JS Runtime (Browser)



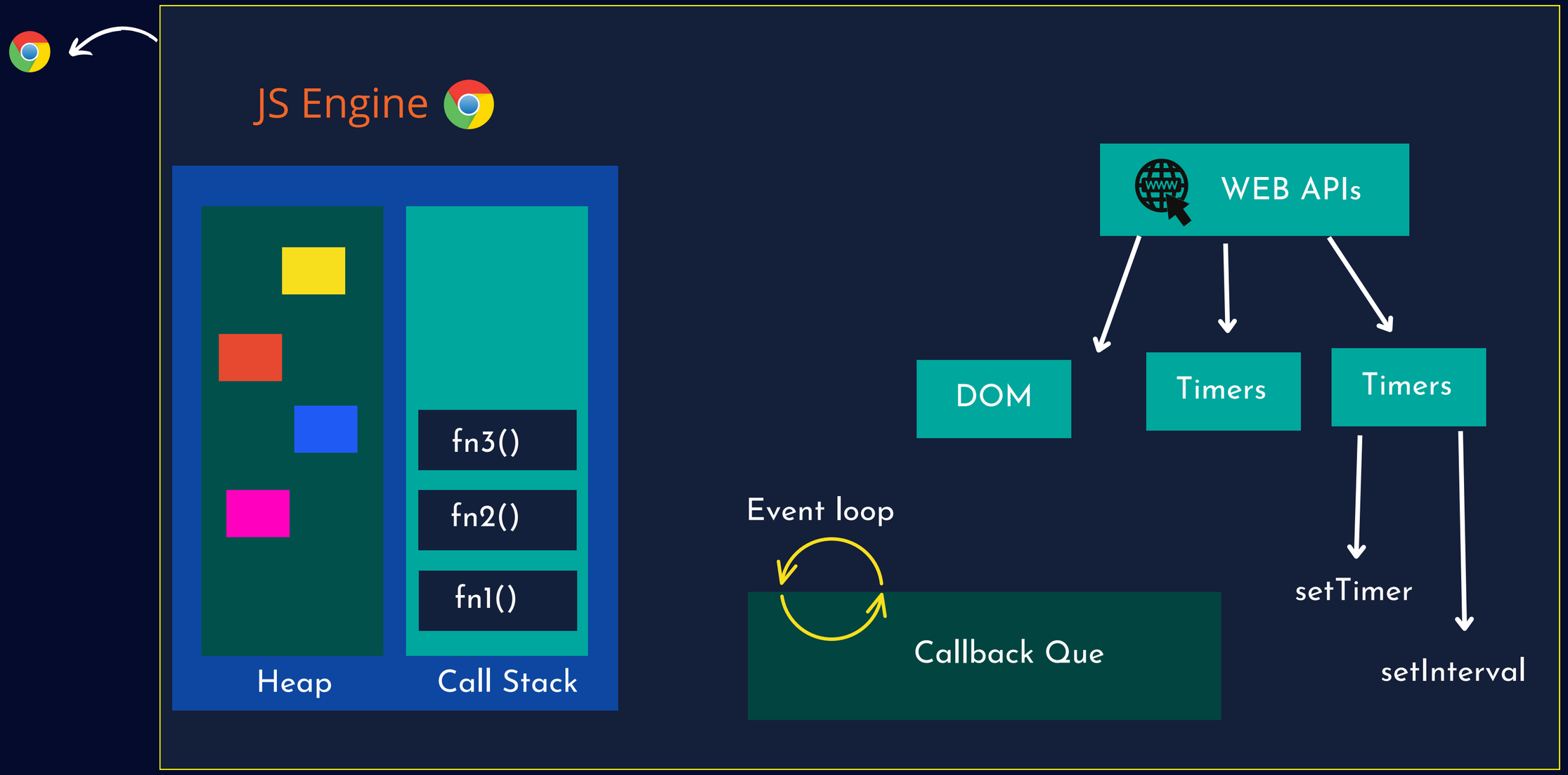
JS Runtime (Browser)



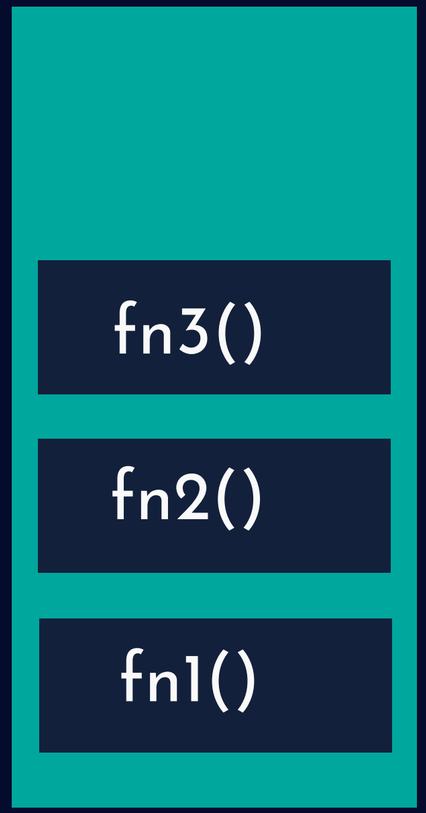
JS Runtime (Browser)



JS Runtime (Browser)



JS Callstack



JS

Synchronous
vs
Asynchronous
programming

Synchronous vs Asynchronous

Synchronous code is executed in a linear fashion, one line at a time. Asynchronous code, on the other hand, can be executed out of order or in parallel.

JS

Asynchronous

Single threaded

Code blocking

Handle one request at a time

Poor user experience

Synchronous

Multi threaded

None blocking code

Handle multiple requests at a time

Improve user experience

Synchronous programming

Single threaded

Synchronous programming

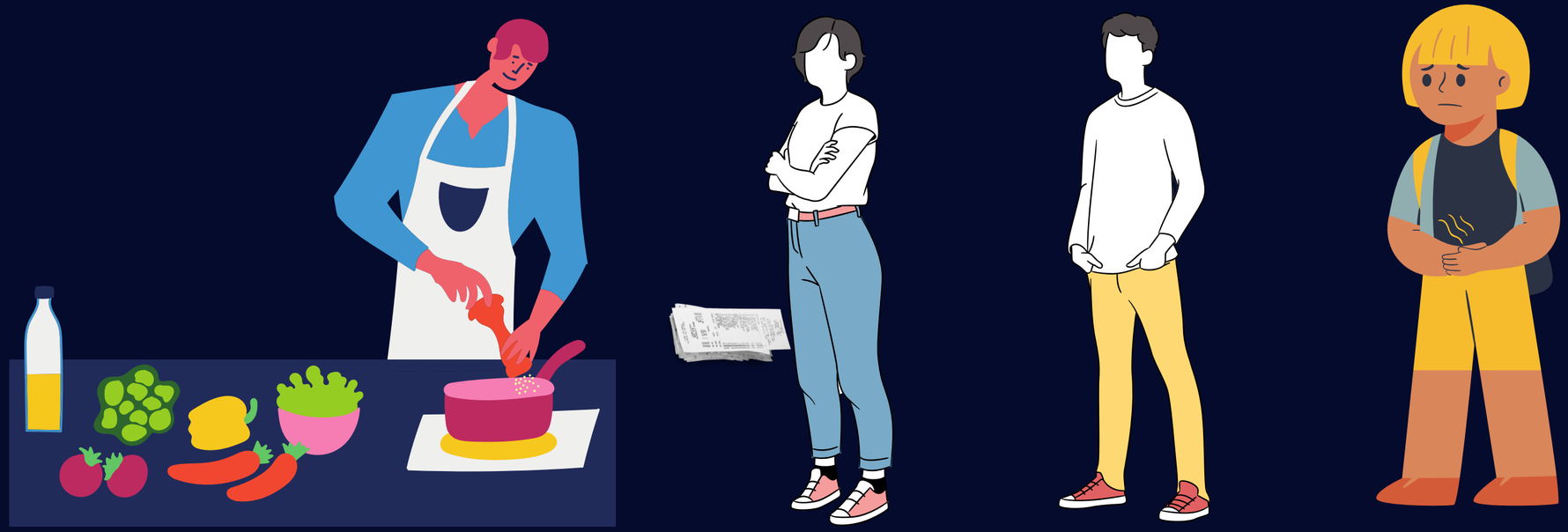
Synchronous programming

Synchronous programming

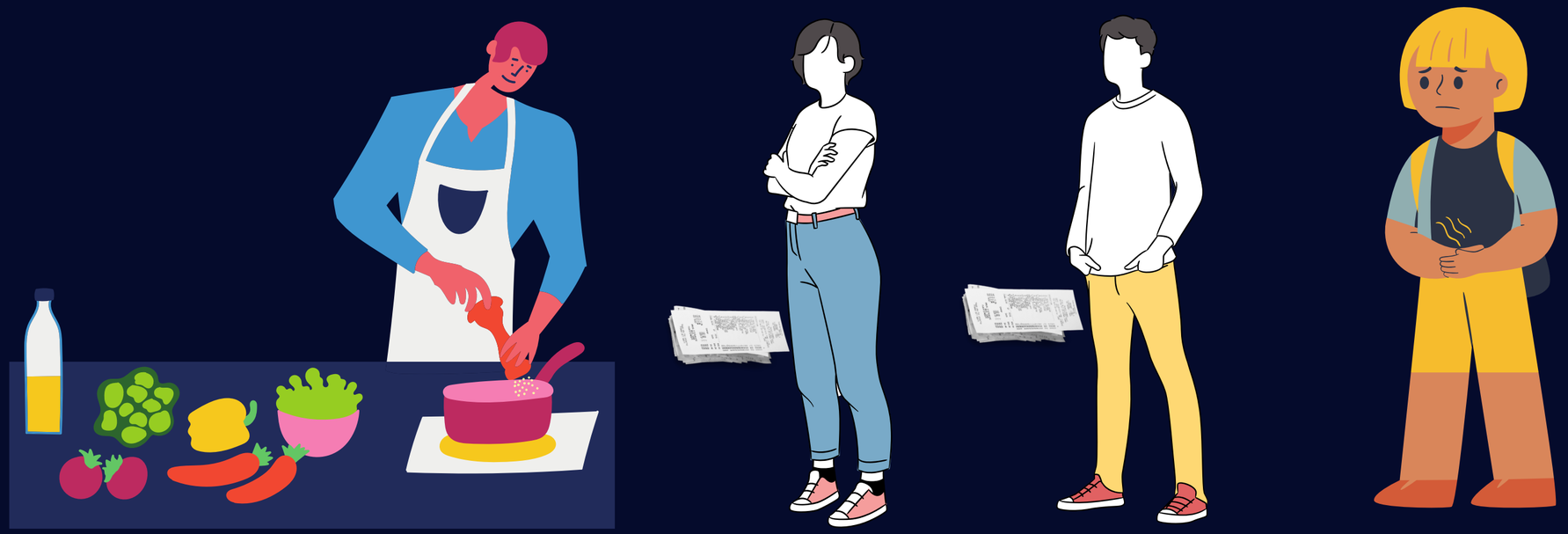
Synchronous programming

Synchronous programming

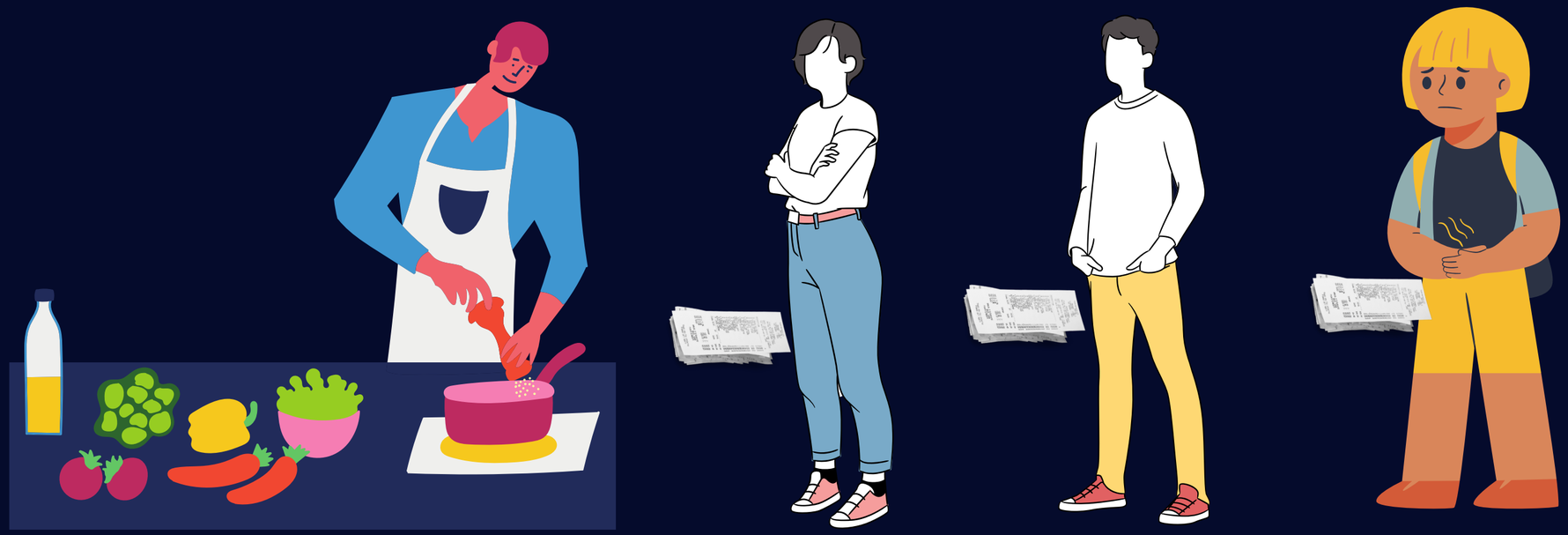
Synchronous programming



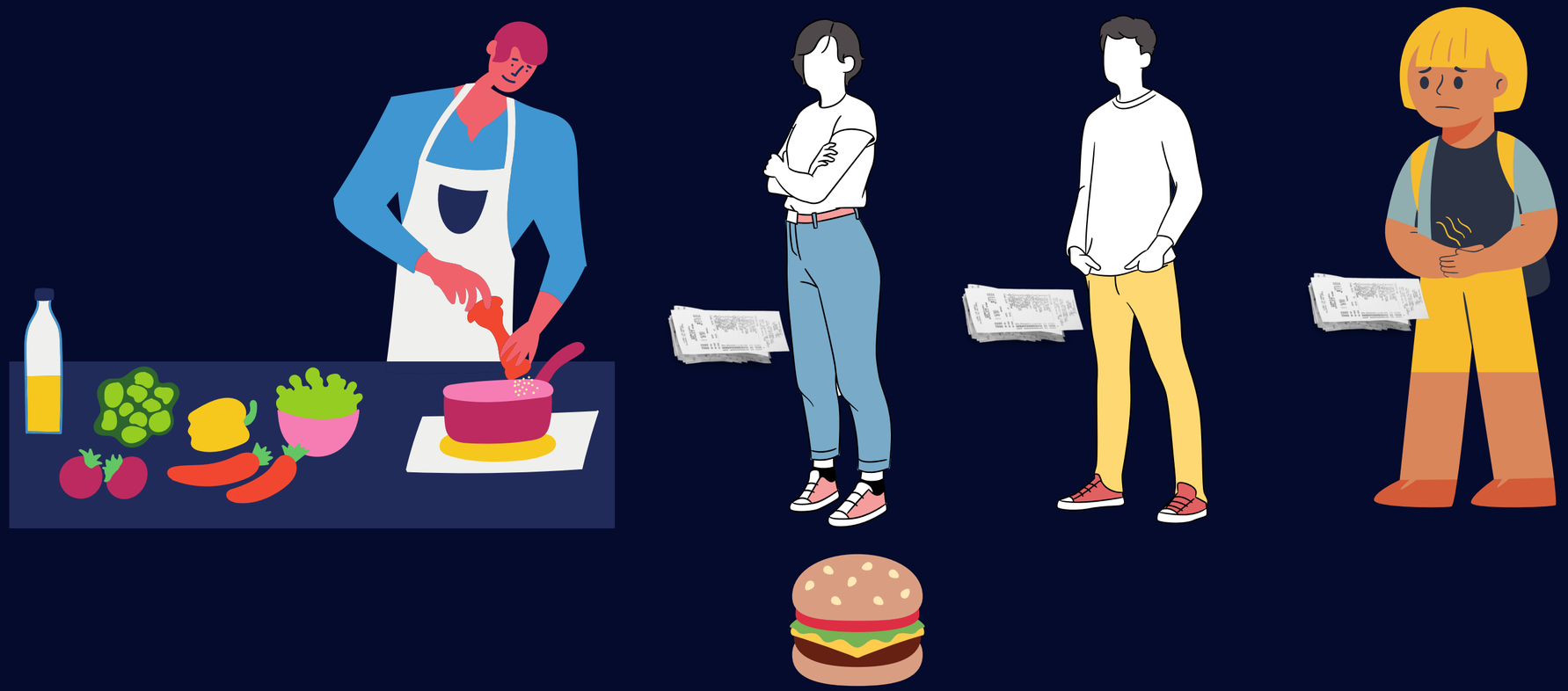
Synchronous programming



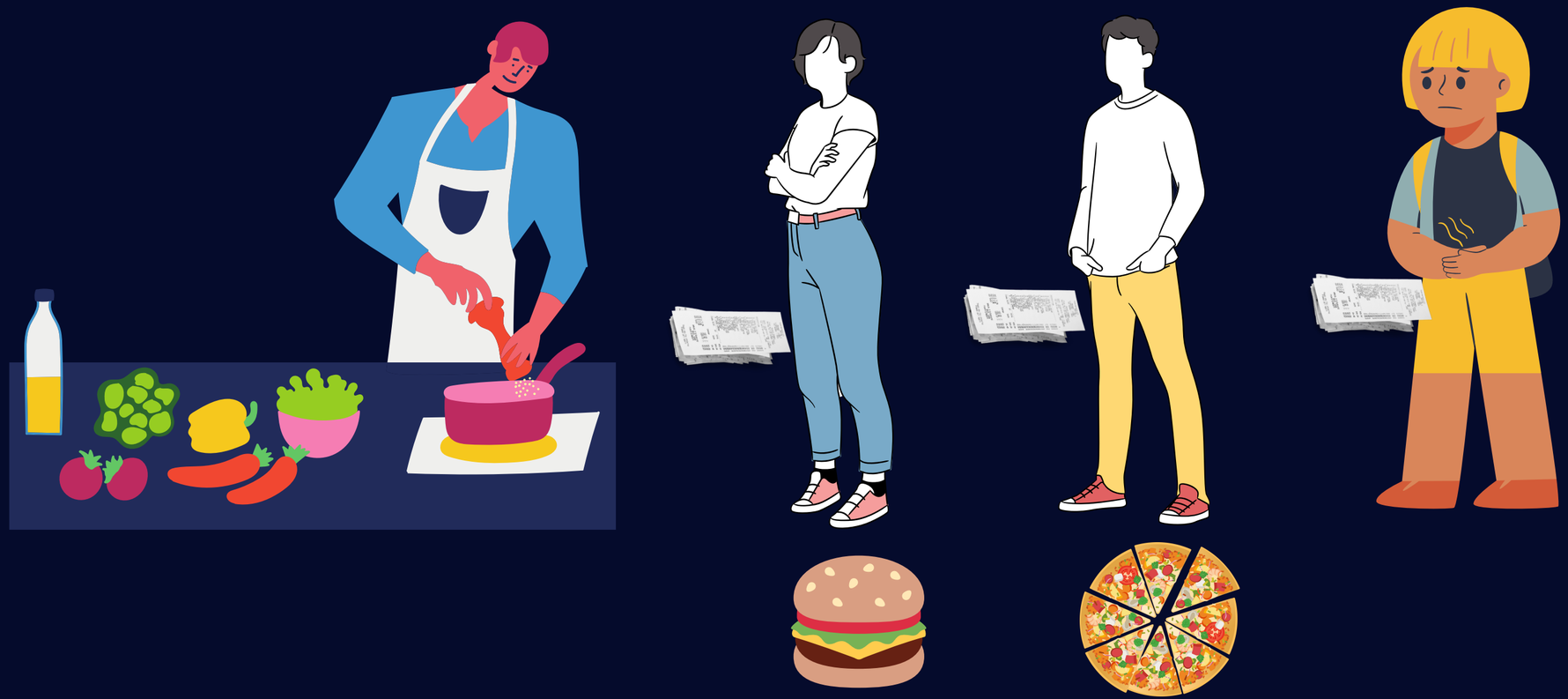
Synchronous programming



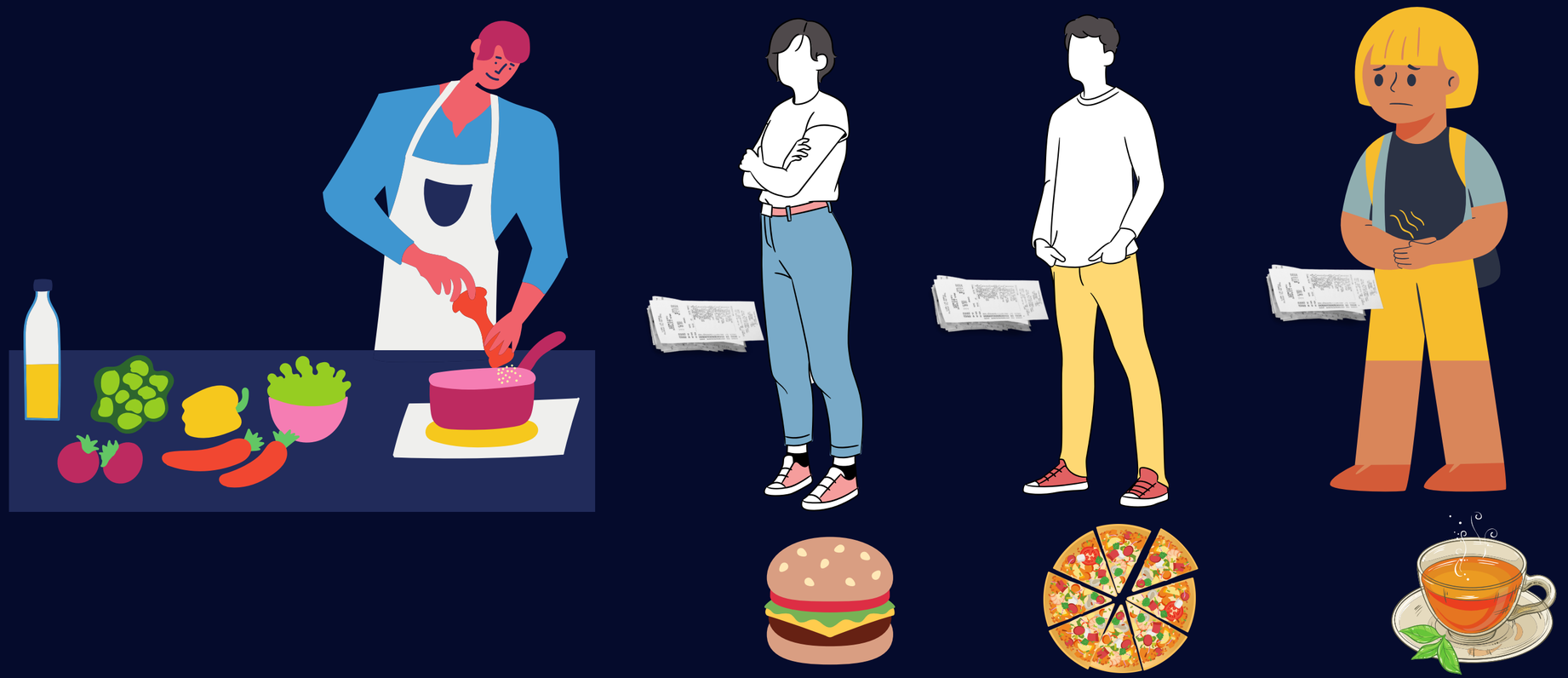
Synchronous programming



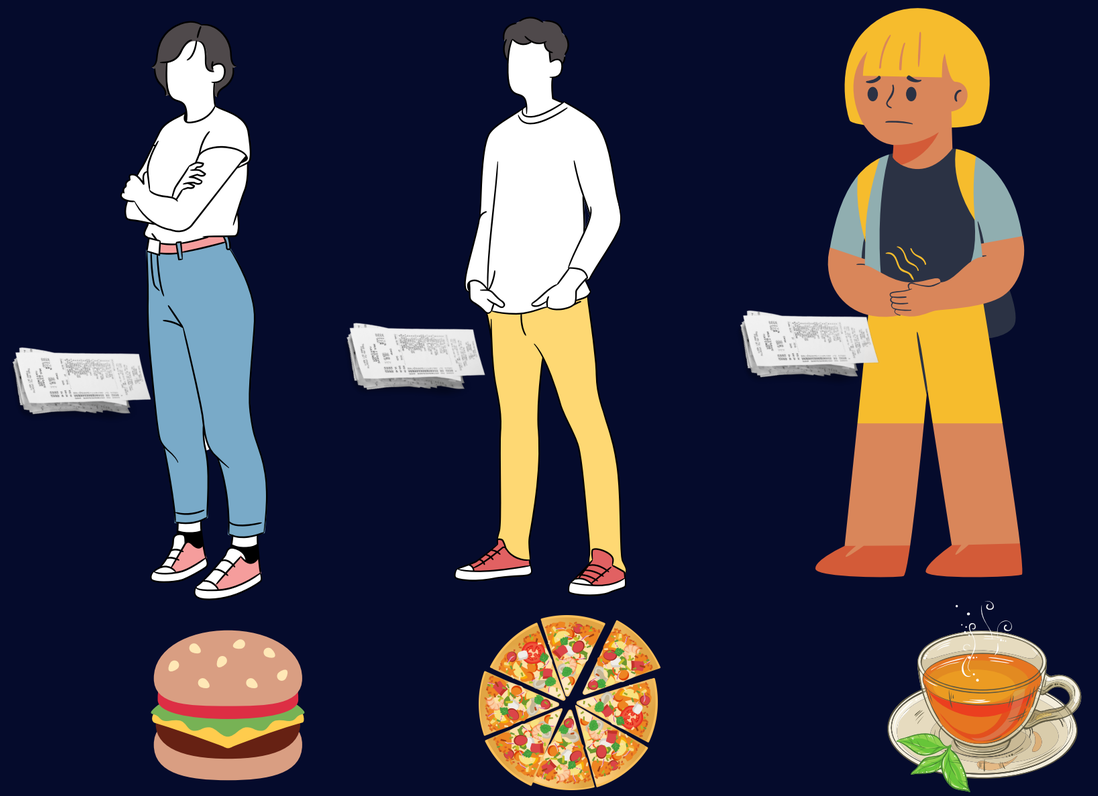
Synchronous programming



Synchronous programming



Synchronous programming



50 minutes to cook

Synchronous programming

50 minutes to cook

10 minutes to cook

JS

Synchronous programming

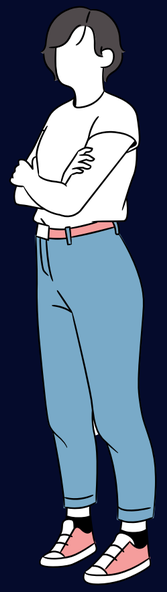
50 minutes to cook

10 minutes to cook

3 minutes to cook

JS

Synchronous programming



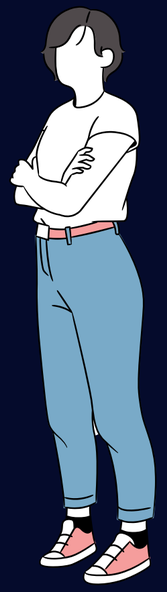
50 minutes to cook

10 minutes to cook

3 minutes to cook

JS

Synchronous programming



50 minutes to cook

10 minutes to cook

3 minutes to cook

Synchronous programming

10 minutes to cook

3 minutes to cook

Synchronous programming

10 minutes to cook

3 minutes to cook

Synchronous programming

3 minutes to cook

Synchronous programming

3 minutes to cook

Synchronous programming

Asynchronous programming

Multi threaded

50 minutes to cook

10 minutes to cook

3 minutes to cook

JS

Synchronous programming

Multi threaded

50 minutes to cook

10 minutes to cook

3 minutes to cook

JS

Synchronous programming

Multi threaded

50 minutes to cook

10 minutes to cook

3 minutes to cook

Synchronous programming

Multi threaded

50 minutes to cook

10 minutes to cook

Synchronous programming

Multi threaded

50 minutes to cook

Synchronous programming

Multi threaded

JS

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