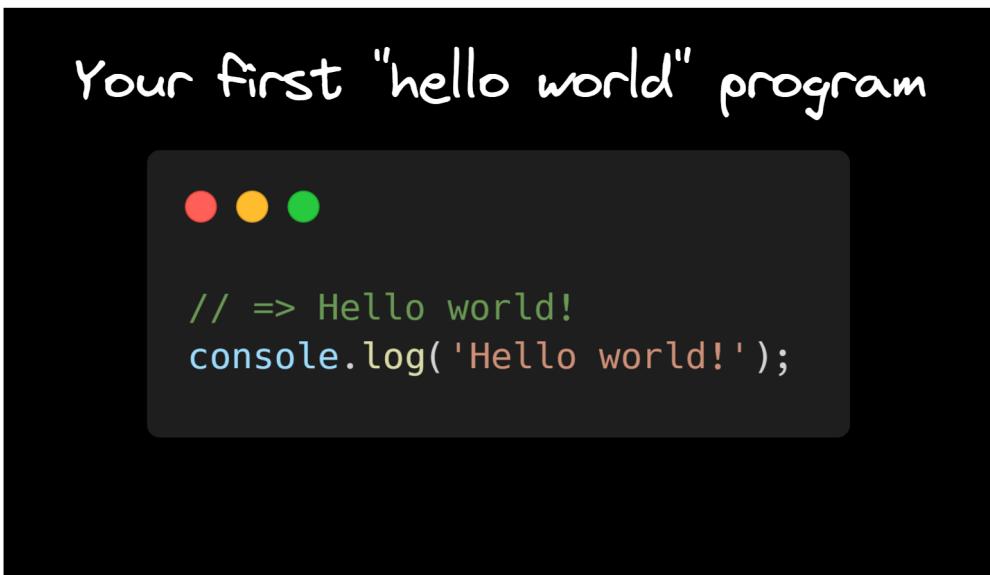


# The ultimate JavaScript cheatsheet

By Hasan Toor

JavaScript is a programming language that powers the dynamic behavior on most websites. Alongside HTML and CSS, it is a core technology that makes the web run.

# JavaScript Fundamentals



Congratulation! You just wrote your first  
JavaScript program

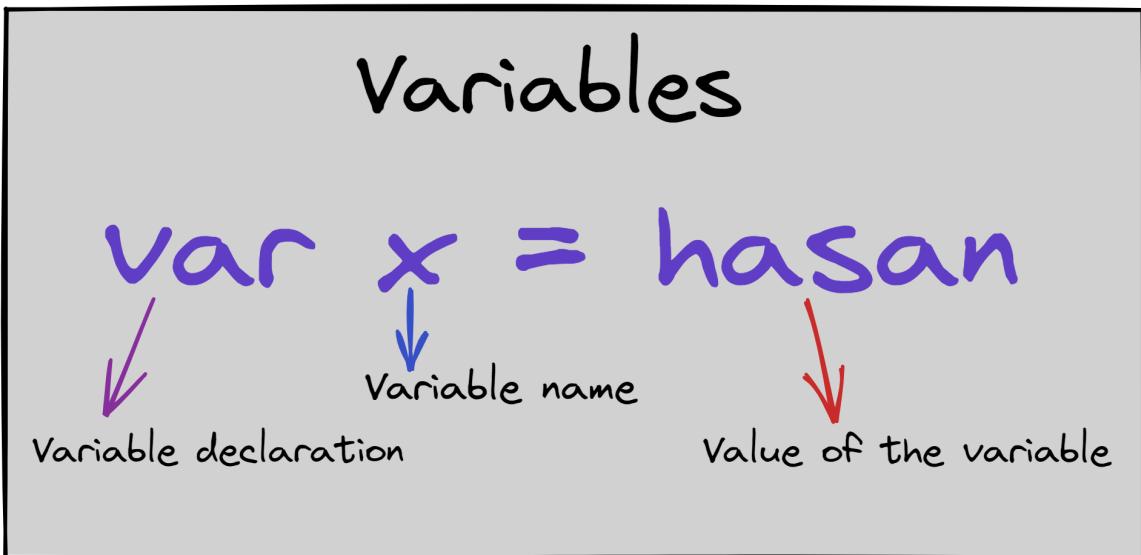
## console.log()

The `console.log()` method is used to log or print messages to the console. It can also be used to print objects and other info.

## Variable declaration

Variables in JavaScript are simply names of storage locations.

Variable declaration: var, const, let



More examples

let x = null;  
let name = "Hasan";  
const found = false;

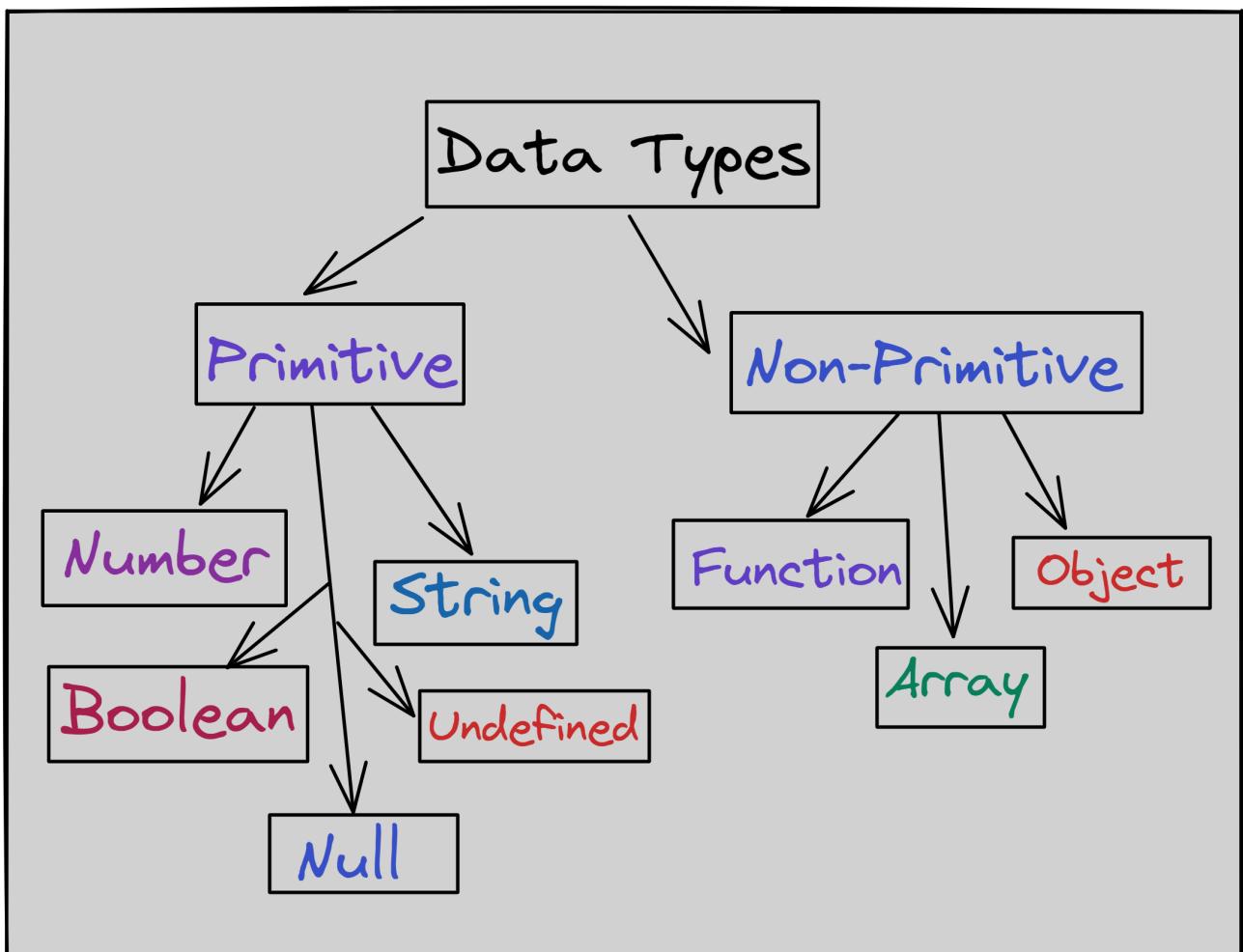
// => Hasan, false, null  
console.log(name, found, x);

var a;  
console.log(a); // => undefined

A screenshot of a terminal window on a dark background. It shows several lines of JavaScript code. The first three lines declare variables: 'name' with value 'Hasan', 'found' with value 'false', and 'x' with value 'null'. The next line uses 'console.log' to print the values of 'name', 'found', and 'x' to the console, resulting in the output 'Hasan, false, null'. The final two lines declare 'a' and then log its value to the console, which is 'undefined'.

# Javascript Data Types

Different types of values and data can be stored in JavaScript variables.



Data Types are important in a programming language to perform operations on the variables

```
● ● ●  
// Data Types examples  
var x = 10 // number variable  
var x = "hi" // string variable  
var x = true // boolean variable  
function x { // your function code here } // function variable  
var x = {} // object variable  
var x = null // null variable  
var x // undefined variable
```

## Arithmetic Operators

### Arithmetic Operators

```
5 + 5 = 10      // Addition
10 - 5 = 5      // Subtraction
5 * 10 = 50     // Multiplication
10 / 5 = 2      // Division
10 % 5 = 0      // Modulo
```

### Comments in JavaScript



```
// This line will denote a comment
```

```
/*
The below configuration must be
changed before deployment.
*/
```

## Comparison Operators

### Comparison Operators

`==` Equal to



`==` Equal value and equal type

`!=` Not equal

`!=` Not equal value or not equal type

`>` Greater than

`<` Less than

`>=` Greater than or equal to

`<=` Less than or equal to

## Logical Operators

### Logical Operators

&& Logical and

|| Logical or

! Logical not

### Bitwise Operators

& AND statement

| OR statement

~ NOT

^ XOR

<< Zero fill left shift

>> Signed right shift

>>> Zero Fill right shift

# Functions

- It is a piece of code ideally with a single purpose
- It is a wrapper around a piece of code
- It provides an abstraction to a block of code
- It provides a way to reuse functionality

## Normal Function Declaration

```
// Defining the function:  
function sum(num1, num2) {  
    return num1 + num2;  
}  
  
// Calling the function:  
sum(3, 6); // 9
```

## Arrow Function

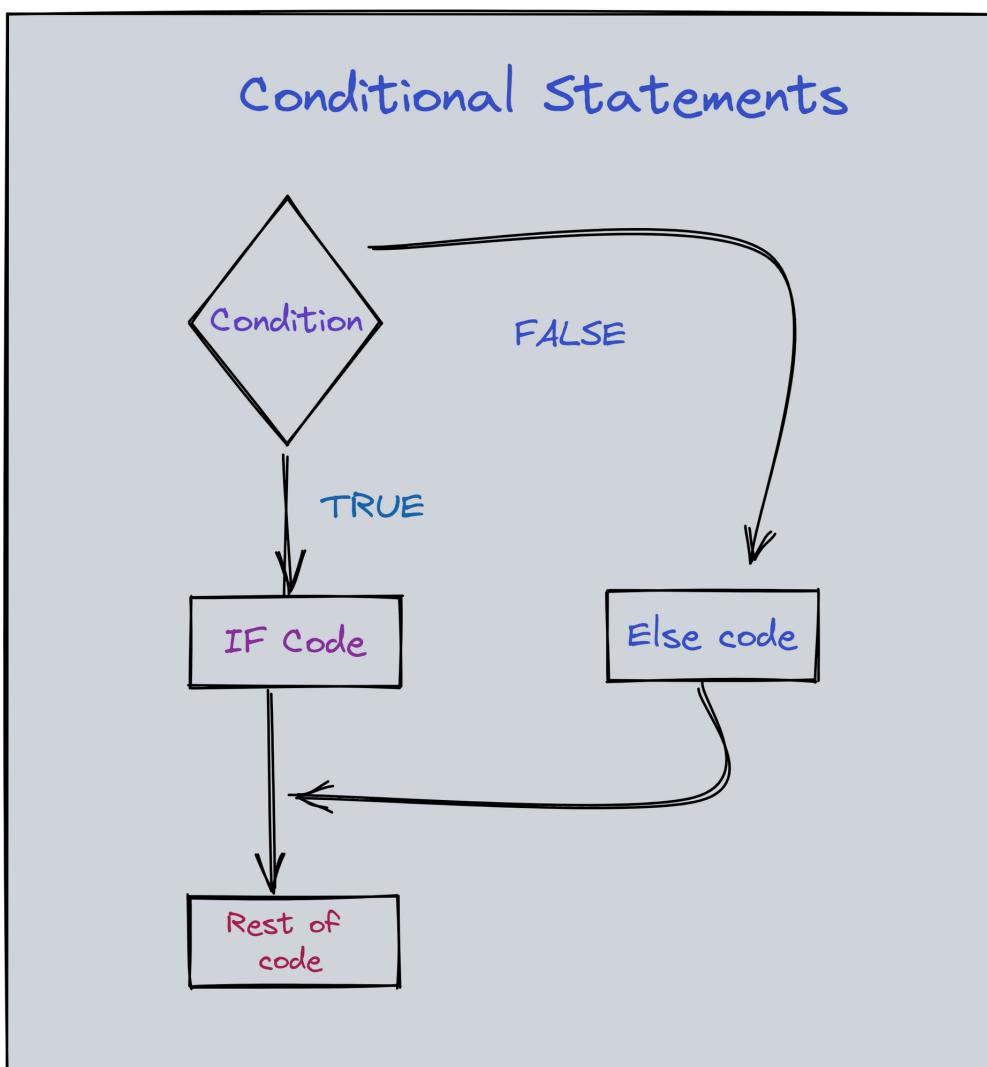
```
const name = (parameter) => {  
  
}  
// statements
```

## Function stored in a variable

```
let name = function (parameter) {  
  
}  
// statements
```

## Conditional statements

- Use **IF** to specify a block of code to be executed, if a specified condition is true
- Use **ELSE** to specify a block of code to be executed, if the same condition is false
- Use **ELSE IF** to specify a new condition to test, if the first condition is false
- Use **SWITCH** to specify many alternative blocks of code to be executed



# If - Else Statements

```
● ● ●  
if (condition) {  
  
    // code to be executed if the  
  
    condition is true  
  
} else {  
  
    // code to be executed if the  
  
    condition is false  
  
}
```

# If - Else If - Else Statements

```
● ● ●  
if (condition1) {  
  
    // code to be executed if the condition is true  
  
} else if (condition2) {  
  
    // code to be executed if the condition1 is false and condition 2 is true  
} else {  
  
    // code to be executed if condition1 is false and condition2 is false  
}
```

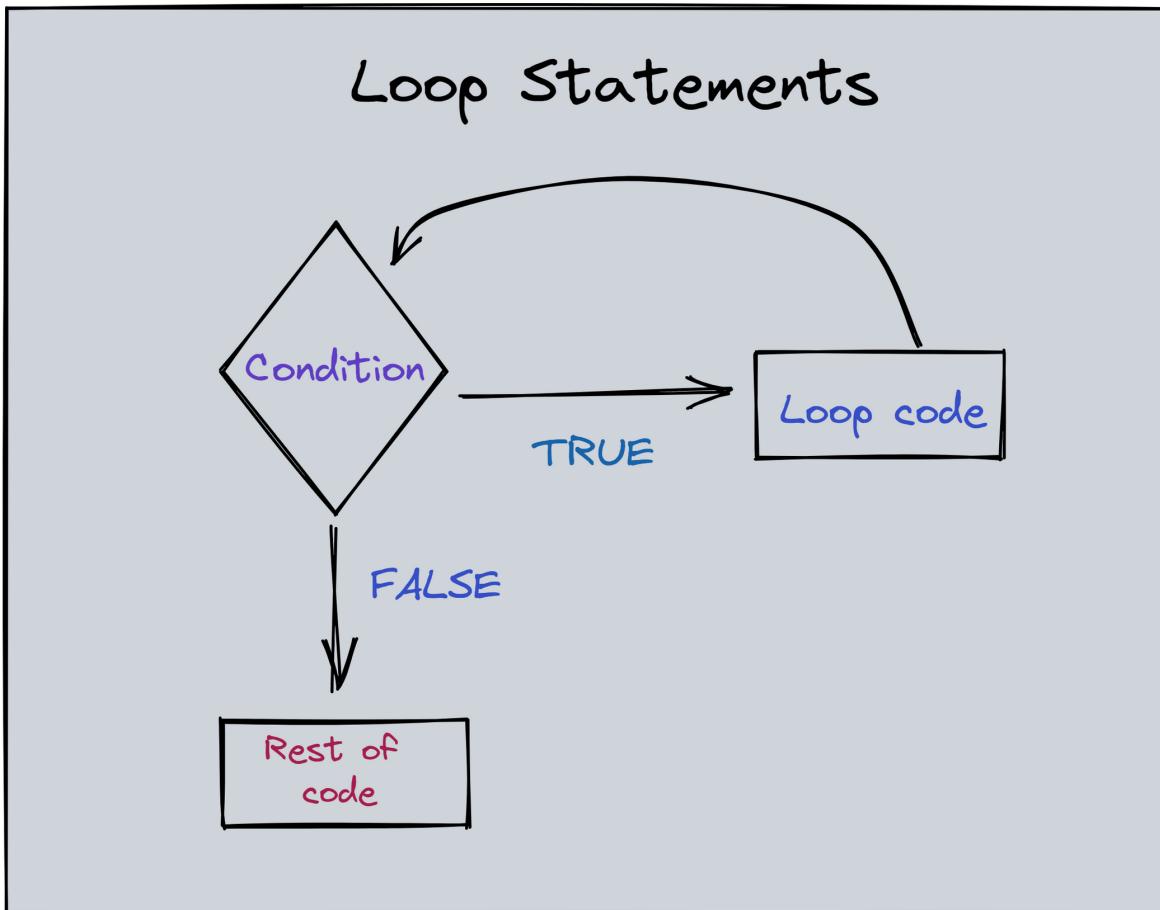
# Switch Statement

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- If there is no match, the default code block is executed.



```
switch(expression) {  
  
    case x:  
        // code block  
  
        break;  
  
    case y:  
  
        // code block  
  
        break;  
  
    default:  
        // code block  
  
}
```

# Loop Statements



## 1. While Loop



```
while (condition) {  
    // code block to be executed  
}  
  
let i = 0;  
while (i < 5) {  
    console.log(i);  
    i++;  
}
```

# Loop Statements

## For Loop

```
● ● ●  
  
for (let i = 0; i < 4; i += 1) {  
    console.log(i);  
};  
  
// => 0, 1, 2, 3
```

## Reverse Loop

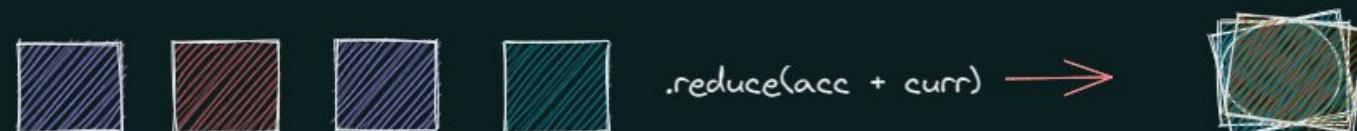
```
● ● ●  
  
const fruits = ["apple", "orange", "banana"];  
  
for (let i = fruits.length - 1; i >= 0; i--) {  
    console.log(`#${i}. ${fruits[i]}`);  
}  
  
// => 2. banana  
// => 1. orange  
// => 0. apple
```

## Do...While Statement

```
● ● ●  
  
x = 0  
i = 0  
  
do {  
    x = x + i;  
    console.log(x)  
    i++;  
} while (i < 5);  
// => 0 1 3 6 10
```

JS

# Array Methods Cheatsheet



# Javascript DOM Cheatsheet

```
< div id = " app " class = " hero dark " >
  < h1 > Hasan is a Developer < / h1 >
< / div >
```

```
// single element defined by id
const app = document.getElementById ( ' app ' )

// multiple elements ( arrays ) defined by class name
const hero = document.getElementsByClassName ( ' hero ' )

// multiple elements based on html tag
const h1 = document.getElementsByTagName ( ' h1 ' )

// first element based on selector
const hero = document.querySelector ( '. hero ' )

// multiple elements based on selector
const heroes = document.querySelectorAll ( '. hero ' )
```

```
// create html element of tag < p >
let para = document.createElement ( ' p ' )

// create text node
let text = document.createTextNode ( ' India ' )

// add text node to the para element
para.appendChild ( text )

// insert h2 before h1
app.insertBefore ( h2 , h1 )

< p > India < / p >
```

```
h1.insertAdjacentHTML ( ' beforebegin ' , ' < span > cool < / span > ' )
  // beforebegin = > placed before h1 as a sibling
  // afterbegin = > placed inside h1 as a first child
  // beforeend = > placed inside h1 as a last child
  // afterend = > placed after h1 as a sibling
```

app.classList.remove ( ' dark ' )	// remove class
app.classList.add ( ' light ' )	// add class
app.classList.toggle ( ' visible ' )	// toggle class
app.classList.contains ( ' app ' )	// true if app present
app.childNodes	// retrieve all child nodes
app.parentNode	// return parent node

## .concat()

Joins two or more arrays, and returns a copy of the joined arrays

## .every()

Checks if every element in an array pass a test

## .filter()

Creates a new array with every element in an array that pass a test

## .findIndex()

Returns the index of the first element in an array that pass a test

## .forEach()

Calls a function for each array element

## .includes()

Check if an array contains the specified element

## .indexOf()

Search the array for an element and returns its position

## .join()

Joins all elements of an array into a string

## .unshift()

This method adds new elements to the beginning of an array. This method overwrites the original array.

## .toString()

Converts an array to a string, and returns the result

## .map()

Creates a new array with the result of calling a function for each array element

## .pop()

Removes the last element of an array, and returns that element

## .push()

Adds new elements to the end of an array, and returns the new length

## .reduce()

Reduce the values of an array to a single value (going left-to-right)

## .reverse()

Reverses the order of the elements in an array

## .shift()

Removes the first element of an array, and returns that element

## .slice()

Selects a part of an array, and returns the new array

## .some()

Checks if any of the elements in an array pass a test

## .sort()

Sorts the elements of an array

## .splice()

Adds/Removes elements from an array