

## **JavaScript Foundation**

JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more.

Features of Javascript:

- javascript is Interpreted Language not a Compiled language.  
compiler language - C/c++ ,java ..golang etc.
- compiler - compile the code at once but interpreter compile line by line
- browser can only understand HTML/CSS/JS thanks to node.js ,js can also used for "Backend development"
- JavaScript is dynamic language during runtime it to change data type of your variable but its not good i.e. typescript comes in this case
- single threaded nature of JavaScript:  
it compile code single line by line .

## Simple Primitives:

### String:-

A string is a sequence of characters enclosed within single ( ' ') or double ( " ") quotes

### Number:-

Numbers in JavaScript can be integers or floating-point numbers. They can be positive, negative, or zero.

### Boolean:-

Boolean: A boolean represents one of two values: true or false. Booleans are typically used for logical operations and comparisons.

```
// Number
let age = 25;
let temperature = -10.5;

// Boolean
let isLoggedIn = true;
let isAdult = false;

// Example usage
console.log(greeting); // Output: Hello, world!
console.log("My name is " + name); // Output: My name is John
console.log("I am " + age + " years old"); // Output: I am 25 years old
console.log("The temperature is " + temperature + " degrees Celsius"); // Output: The
console.log("Am I logged in? " + isLoggedIn); // Output: Am I logged in? true
console.log("Am I an adult? " + isAdult); // Output: Am I an adult? false
```


## Complex Primitives:-

### Arrays:

Arrays are ordered collections of values. Each value in an array is identified by an index, starting from 0.

Arrays have built-in methods like `push`, `pop`, `shift`, `unshift`, `slice`, etc., for manipulation.


javascript

 Copy code

```
let fruits = ['apple', 'banana', 'orange'];
let numbers = [1, 2, 3, 4, 5];
```

You can access elements in an array using square brackets and the index of the element:


javascript

 Copy code

```
console.log(fruits[0]); // Output: apple
console.log(numbers[2]); // Output: 3
```

Arrays can hold different data types, including other arrays:


javascript

 Copy code

```
let mixedArray = ['apple', 10, true, ['nested', 'array']];
console.log(mixedArray[3][0]); // Output: nested
```

Arrays have built-in methods like `push`, `pop`, `shift`, `unshift`, `slice`, etc., for manipulation.

javascript

 Copy code

```
fruits.push('grape'); // Adds 'grape' to the end of the array
console.log(fruits); // Output: ['apple', 'banana', 'orange', 'grape']
```

## Objects:

Objects: Objects are collections of key-value pairs. Keys are strings (or Symbols) and values can be any data type.

```
javascript Copy code

let person = {
  name: 'John',
  age: 30,
  isStudent: false
};

let car = {
  make: 'Toyota',
  model: 'Corolla',
  year: 2020
};

You can access values in an object using dot notation or square brackets:

javascript Copy code

console.log(person.name); // Output: John
console.log(car['make']); // Output: Toyota
```


function:

- abstract out logic in your program
- take argument as input and return as output  
functions can take other function as input -  
callback

### Function Parameters:

Functions can take zero or more parameters. Parameters are placeholders for values that will be passed to the function when it's called.

javascript


 Copy code

```
function add(a, b) {  
    return a + b;  
}  
  
console.log(add(2, 3)); // Output: 5
```

### Return Statement:

Functions can return a value using the `return` statement. When a function encounters a `return` statement, it immediately exits, and the value after `return` is returned to the caller.

javascript

 Copy code

```
function multiply(a, b) {  
    return a * b;  
}  
  
let result = multiply(4, 5);  
console.log(result); // Output: 20
```

# What is Web API?

API stands for Application Programming Interface.

A Web API is an application programming interface for the Web.

A Browser API can extend the functionality of a web browser.

A Server API can extend the functionality of a web server.

For ex- `Fetch()`, `setTimeout()`, `setTimein()` etc...