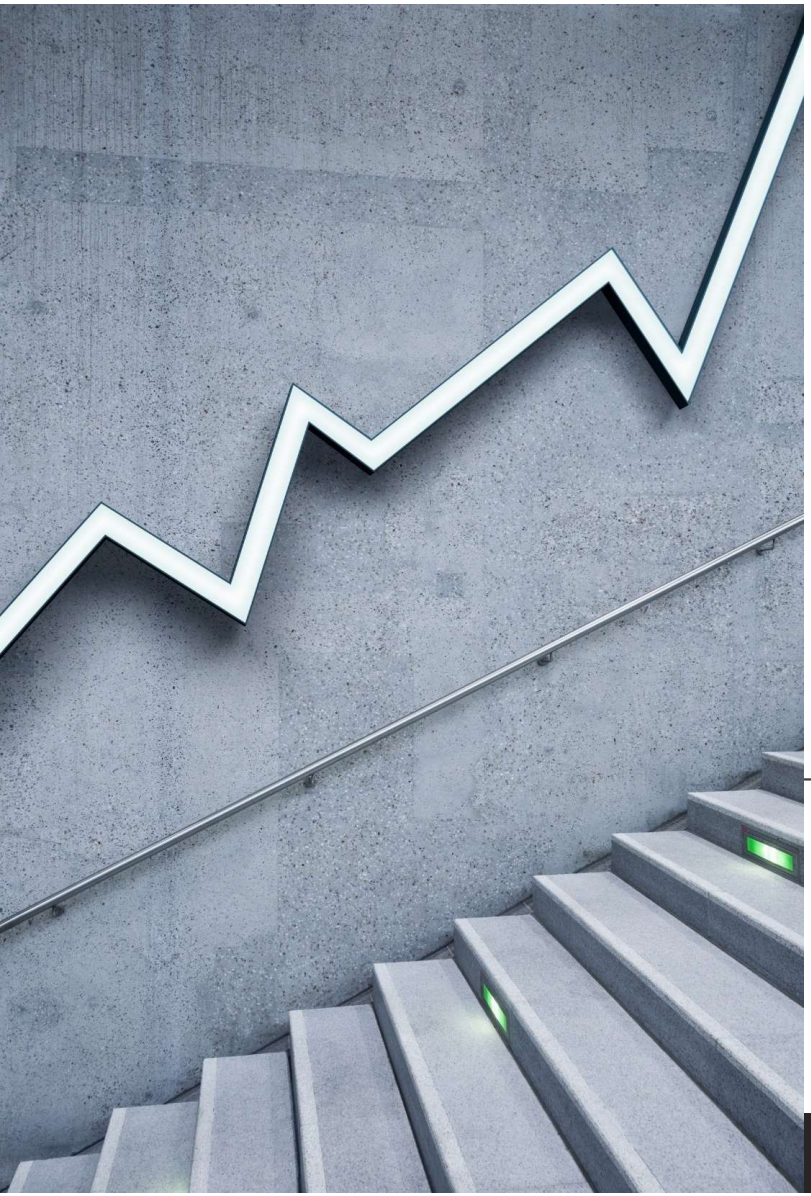

Learning JavaScript

PRAVEEN NAIR

What is JavaScript?

Used to program the behavior of web pages
JavaScript was invented by Brendan Eich in 1995.

JavaScript code is inserted between <script> and </script> tags.

Javascript was developed by Netscape

JavaScript vs VBScript (Microsoft)

Javascript supports all browser, vbscript supports IE

Originally Sun Microsystem and now Oracle

First Program

```
let a=10; // line break also works
let b=20;
let c = a + b;
console.log(c);
```

Comments // and /*

```
let name='John';
    let age=20
  /*
console.log(name)
*/
```

Printing using backtick

```
let n=2;
let s = `Price of an apple is ${n}`;
document.write(s)
```

……..

Also called template literals….try multiline

Math Operators

Addition + (also concatenates string)

Subtraction -

Multiplication *

Division /

Remainder %

Exponentiation **

Comparison Operators

Operator	Description	Comparing	Returns
==	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
>	greater than	x > 8	FALSE
<	less than	x < 8	TRUE
>=	greater than or equal to	x >= 8	FALSE
<=	less than or equal to	x <= 8	TRUE

Logical Operators

Logical NOT (!)

Logical AND (&&)

Logical OR (||)

Precedence of AND && is higher than OR ||

Conditional branching: if

```
let n = 7
if (n%2==0){
  console.log("Even Number")
}
else{
  console.log("Odd Number")
}
```

while loop

```
while (condition) {
  ...
}
```

For loop

```
for (let i = 0; i < 3; i++) {
  console.log(i);
}
```

Try break and continue

JavaScript Functions

```
function showMsg() {
  console.log( ‘Hello World!' );
}

showMsg();
```

Passing arguments

```
function sum(a, b) {
  c = a + b;
 console.log(c);
}
sum(1, 2);
```

Returning Values

```
function sum(a, b) {
  return a + b;
}
let result = sum(1, 2);
console.log( result ); // 3
```

Variables

let

const (constant, can’t be changed)

Var –

var is function scoped and let is block scoped. Variable declared by let cannot be redeclared

Variables are case-sensitive, try camelCase, titlecase, with dash

Data Types (Primitive/Value type)

1. let n=2;
2. let s = “Hello World”; //double or single quote
3. let flag = true; //true or false - boolean
4. let name; //undefined
5. let cost=null;

Type conversion

```
let value = true;
console.log(typeof value); // Boolean
value = String(value);

let numStr="34";
num = Number(numStr); // becomes a number 123

console.log(Boolean(num))

/* Values that are intuitively “empty”, like 0, an empty string, null,
undefined, and NaN, become false. Other values become true.*/
```

Data Types (Reference Type)

1. Objects
2. Arrays
3. Functions

Function Expressions

```
let sayHello = function() {
        console.log( "Hello World" );
    };

sayHello();
```

Arrow functions

```
let result = (a, b) => {
            let c = a + b
            return c
    };


let result = function(a, b) {
      let c = a + b
      return c;

  };


console.log(result(3, 2) );
```

Arrow Functions Recap

```
let a = b = 10
const fnc = () => a + b  //no braces and return is needed

let a = b = 10
const fnc = () => return a + b  //can't use return without braces

let a = 10;
const fnc = x => x + 20; //no brackets for x needed
console.log(fnc(a));

let a = (b = 10);
const fnc = () => {
  return a + b;  //return is needed if braces are used
};
```

Objects – Keyed Collections

```
let student = {
  name: "Smitha",
  age: 30
};
console.log(student.name) // student[“house address”]
console.log(student.age)
Console.log(student)
```

Objects – Spread Operator

```
const student = {
    name:"John",
    age:21
}
//spread operator
const obj = {...student,city:"NYC"}
console.log(obj)
```

Objects – return

```
let a = b = 10
    const fnc = () => ({name:'John'})  //need backet if returning obj
    console.log(fnc());
```

Array Items – foreach, map

```
let fruits = ["apple", "mango", "orange"];
    fruits.forEach((value,index,arr) => {
      console.log(value,index,arr);
    });
let fruits = ["apple", "mango", "orange"];
    fruits.map((value, index, arr) => {
      console.log(value, index, arr);
    });
```

Array Items – Spread Operator

```
const names = ["Vivek","Shivam","Aman"]

const arr = [...names,"Suresh"]

console.log(arr)
```

Array Items – filter and find

```
let score = [34, 12, 67, 89, 30];
    let result = score.filter((v) => {
      return v > 40;
    });
    console.log(result);
.....
let empnum = [1003, 1005, 1006, 1034];
    let result = empnum.find((v) => {
      return v == 1003;
    });
    console.log(result);
```

Reduce method

```
function calculateTotal() {
  let total = cart.reduce((sum, item) => sum + item.price * item.quantity, 0);
  return total;
```

Array Items – some method

```
let marks = [10,60,80,40]
    let result = marks.some((value)=>{
        return value > 30
    })
    console.log(result)
```

Array Items – every method

```
let marks = [10,60,80,40]
    let result = marks.every((value)=>{
        return value > 30
    })
    console.log(result)
```

console.table

let array = [“a”,”b”]

console.table(array)

Array of Objects

```
<script>
    const products = [
      { name: "Product 1", price: 300 },
      { name: "Product 2", price: 100 },
      { name: "Product 3", price: 500 },
    ];
    const cart = [];
    let item = products[0];
    item.quantity = 2;
    item.total = item.quantity * item.price;
    cart.push(item);
    item = products[2];
    item.quantity = 3;
    item.total = item.quantity * item.price;
    cart.push(item);
    console.log(cart);
    let orderValue = cart.reduce((sum, value) => {
      return sum + value.total;
    },0);
    console.log("Order Value is", orderValue);
  </script>
```

(IIFE)immediately invoked function expression

```
(function functionName() {
  console.log("Hello World");
})();

(function functionName() {
  console.log("Hello World");
})();
```

Array from method

```
<script>
    let arr = Array.from("aeiou");
    document.write(arr);
</script>
```

Ternary/conditional operator ‘?’

```
let isEligible = (age > 18) ? true : false;

Try multiple condition
condition1
    ? true_expression1
    : condition2
        ? true_expression2
        : else_expression2
```

Switch statement

```
let price = 40;

switch (price) {
  case 30:
    console.log( 'Too Cheap’ );
    break;

  case 40:
    console.log( 'Perfect Price’ );
    break;

  case 50:
    console.log( 'Too Costly’ );
    break;

  default:
    console.log( "I don't know the price" );

}
```

Functions (...args) vs arguments

```
<script>
 function sum(){
       let sum=0
       for (let i=0;i<arguments.length;i++){
         sum = sum + arguments[i]
       }
       console.log(sum)
     }
     sum(2,3,4,5)
.....................................................
function sum(...args) {
       let sum = 0;
       for (let i = 0; i < args.length; i++) {
         sum = sum + args[i];
       }
       console.log(sum);
     }
     sum(2, 3, 4, 5);

 </script>
```

Module Import/Export - multiple

```
 function add(x,y){
    return x+y
}
function subtract(x,y){
    return x-y
}

export {add, subtract}
.....................................
 import {add,subtract} from "./calc.js“
       let sum = add(4,5)
       console.log(sum)
       let difference = subtract(8,3)
       console.log(difference)
```