

⇒ What is JavaScript?

JS is scripting language but now it is a programming language.

Why we call it programming lang. because of ECMAScript (ES). It is a standard on which JS based and works. ES are the versions of JS. ES1, ES2, ..., ES14.

Evolution started after ES6.

Let's dive into JS programming.

⇒ **Variables & Data Types**

Variable is a container that store data and that data may be varies. it may be a number, string or a character.

`var a = 7;`

`let a = 7;` → Declaring  
↳ Type.

⇒ **Variable name declaration.**

letter, digit, underscore, \$ are allowed.

reserve word cannot be used

→ Diff b/w var, let, const

1. var is globally scoped. It can be updated and re-declared.
2. let is block scope and cannot re-declared.
3. const cannot be updated and also cannot re-declared.

`var a = 12;`

{ `var a = 15;` }  $\Rightarrow$  15  
— output

`let b = 12;`

{ `let b = 15;` }  $\Rightarrow$  12  
— output

=

→ Data Types in JS

String, Symbol

Number, Null.

BigInt, Boolean, Object.

`let a = "I am String";`

`let b = Symbol("Symbol");`

`let c = 12;`

`let d = null;`

let e = BigInt("12");  
 let f = true; // false  
 let obj = {  
 id: 1,  
 name: "John",  
 mail: "m@n.com"
 }

## ⇒ Operators & conditions.

A set of Programme that  
 Performes some type of logic like  
 addition Multiply etc

### 1. Arithmetic operator

- + Addition , - Subtraction
- \* Multiply , \*\* Exponentiation
- / Division , % Modulus
- ++ Increment , -- Decrement

### 2. Assignment operator.

- $\mathbf{=}$  ,  $\mathbf{x = y}$  ,  $\mathbf{+= x = x + y}$
- $\mathbf{+=}$  ,  $\mathbf{x = x - y}$  ,  $\mathbf{*= x = x * y}$
- $\mathbf{/= x = x / y}$  ,  $\mathbf{./= x = x ./ y}$
- $\mathbf{**= x = x ** y}$

(9)

### 3. Comparison operator

- = = equal to
  - != not equal
  - == = equal value and type
  - != = not equal value or not equal
  - > greater than
  - < less than
  - >= greater than or equal to
  - <= less than or equal to
  - ? ternary operator
- ### 4. Logical operator
- && Logical and
  - || Logical or
  - ! logical not

### Comments

Single line comment //

Multy line comment /\* \*/

### Conditional Statement

Condition true executes code

block otherwise do something

else

1, if statement

2, if... else statement

3, if... else if ... else statement

(5)

```
if (condition){  
    // execute code  
}
```

```
if (condition){  
    // execute code  
}  
else {  
    // do something else  
}
```

```
if (condition){  
    // execute code  
}
```

```
else if (condition){  
    // execute code  
}
```

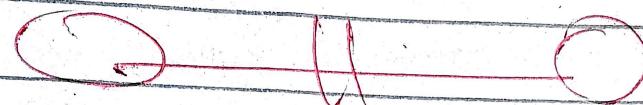
```
else if (condition){  
    // execute code  
}
```

```
else {  
    // do something  
}
```

6

## Ternary operator

This operator is used to evaluate single line condition  
(Condition) ? true : false



## Loops In javascript

A loops is a statement that is used to iterate in a sequence using key / index values Pattern

### For loop

For loop used to iterate until and limit

```
for (let i = 0; i < 10; i++) {  
}
```

### For in loop

For in loop is used to iterate over an object or an array

```
let obj = {
```

```
    id: 1,
```

```
    name: "john doe"
```

```
}
```

```
for (let b in obj) {
```

```
    console.log(obj[b]);
```

```
}
```

## For off loop

For off loop is used to iterate a string

Let str = "Some string";

for (let c of str) {

    console.log(c);

}

## While loop

While loop used to iterate same like for loop

while (condition) {

    console.log(i);

    i++

}

## do While loop

do while loop execute at least once

do {

    console.log(i);

    i++;

} while (condition);

## For each loop

for each loop iterate on every single

Let a = [2, 3, 4, 55];

a. `foreach (ele, index, ar)`  
 $\Rightarrow \{$

`Console.log(ele + " - " + index + "`  
`+ ar);`

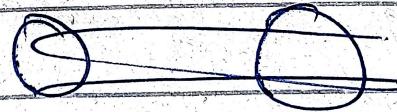
$\}$  );  
array map

Map used to manipulate array  
values we can add new  
value

Let a = [2, 3, 4, 55];

Let b = a.map (value, index)  $\Rightarrow \{$   
`return value + 1;`  
 $\}$  );

`Console.log(b);`



## String Methods & Manipulation

String is a set of char

Let str = "Hello";

\* `str.length`  $\Rightarrow$  return length

\* `str.toUpperCase();`

\* `str.toLowerCase();`

\* `str.replace();`

(9)

- 2)
  - \* str. concat () .
  - \* str. trim () .
  - \* Strings are immutable it means we cannot update string value



## Arrays manipulation

Arrays are containers that holds multiple values.

Const fruits = ["banana", "apple"]

Const nums = [4, 10, 16, 15]

Nums.length => return length

Arrays based on indexes  
and keys values

Arrays are mutables  
means we can change them

Some example of arrays

functions

\* Arr. toString ()

\* Arr. Join ("")

\* Arr. Pop ()

\* Arr. Push ()

\* Arr. Shift ()

- \* Arr. unshift ()
- \* Arr. delete ()
- \* Arr. concat ()
- \* Arr. sort ()
- \* Arr. reverse ()



## Console operation

- \* console.log ()
- \* console.warn ()
- \* console.error ()
- \* console.info ()
- \* console.clear ()

## User interaction

**Alert**

Displays a popup window

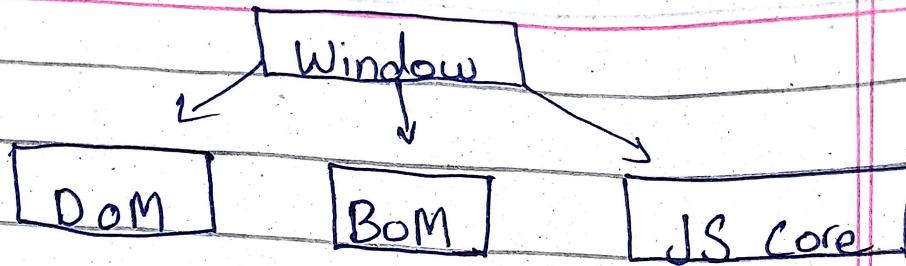
**Prompt**

Get input value from user

**Confirm**

Return true or false

**Window object BOM, DOM**



**DOM**  $\Rightarrow$  Document object model

Dom used to render or manipulate HTML content

Document . body

Document.body.style.colour

**BOM**  $\Rightarrow$  Browser object model

it Provides additional functions that are not included document like alert confirm

Prompt

**More about DOM**

it uses tree structure to manipulate documents content

First child, last child

child nodes, Sibling and Parent

\* `Document.getelement by id`

\* `Document.query selector all`

\* `Document.query selector`

\* Document.getelements by name

\* Document.getelements by tag name

\* Document.getelements by class name

Innes HTML, outer HTML,  
tax content

B Create elements on runtime

Settimeout, SetInterval

Events:

Click, mouse movement.

key up / down.

elem.onclick = function() {

do something }

⇒ addEventListener

## Synchronous / Asynchronous

Syn ⇒ Execute code line by line

Asyn ⇒ Execute code in parallel.

async function abc()

{ return 5 }

abc.then((x) => { alert(x) })

## Error Handling

13

try  $\Rightarrow$  catch()

let a = 10;

console.log(b); } Generate Error

b = 12;

a = 10;

try {

    console.log(b); }

catch(err) {

    console.log('error');

Try catch with setTimeout

Error object explanation

try  $\Rightarrow$  catch  $\Rightarrow$  finally.

$\Rightarrow$  Fetch API

let fet = fetch("URL");

fet.then(v1) {

    we can check Status }

    .then(v2) { Handle Response }

Jsonplaceholder for  
more details about fetch

⇒ Cookies:

Used to store data on user browser.

document.cookies;

⇒ LocalStorage:

To store data on browser

Diff b/w cookie & LS.

Cookies available all over the app environment while LS available only on browser.

localStorage.setItem(k, v)

LS.getItem(k);

LS.clear();

LS.removeItem();

⇒ Session Storage

Same as localStorage.

Store values for short time.

~~window.onstorage = () =>~~

    alert();

}