

* History of JavaScript

* History - In 1995, A netscape (browser) programmer named brandan eich developed scripting language in just 10 days.

* originally name (firstName) - Mocha.
(secondName) - Livescript

At that time java is famous programming language so for marketing purpose Livescript changed in javascript.

Mocha → Livescript → Javascript

In 1997, there is another famous browser that was internet Explorer (microsoft browser). Then microsoft copied javascript features made own language named as jscript.

In browser was (netscape vs internet explorer)

netscape → JavaScript
Internet Explorer → Jscript.

EcmaScript is born---

Ecma International - Ecma International is an industry association founded in 1996, dedicated to the standardization of information and communication systems.

JavaScript + Ecma → Ecmascript

Problem Solved - We can same implement scripting language for different browser

First Ecmascript

ES1 → 1997 (first ever - standard)

ES5 → 2009 (lots of new features)

ES6 → 2015 (biggest update for js)

ES6 is also known as modern JavaScript

Ecma have a technical community known as TC39. They decide after 2015 we release JavaScript with new features every year (Annual Release)

* JavaScript Features.

Following features :-

- Features :- It's a high-level - dynamic - been inodus have no begin
- Case sensitive
- Dynamically-typed: changed in std! -
- cross-platform
- Interpreted
- object-oriented scripting language
- Backward compatible

* Output possibilities

- Write into an HTML element using innerHTML
- Write into the HTML output using `document.write()`
- writing into an alert box using `window.alert()`
- writing into the browser console - using `console.log()`

* Javascript variables

- Variables - variables stored the data which can be changed or used when we need.
 - There are keywords to declare a variable
- keywords are the predefined words in programming language.

- var	var name = 10;	Global scope
- let	let name = 10;	block scope
- const	const pi = 3.14;	can't update values

* JavaScript Declarations are hoisted.

A variable can be used before it has been declared.

e.g

```
<p id = "demo"> </p>      <script>
                           x=5; // assign value to x
elem = document.getElementById("demo");
// A Find an Element
elem.innerHTML = x; // Display x in the element
var x; // Declare x
</script>
```

* Datatype in Javascript

- There are two types of Data primitive & non-primitive
- 1) primitive
- 2) Non-primitive

* primitive datatypes are -

- Number
- Null
- string
- Boolean
- undefined
- BigInt
- symbol

* Non-primitive datatypes are -

- Array
- object
- RegExp

* Javascript Hacks

- convert string to number

put the (+) before the string.

For Example :-

```
let str = "g";  
console.log(typeof(+str));
```

- Convert number into string.

Add a empty string with the number

For Example :-

```
let num = 10;  
console.log(typeof(num + ""));
```

* undefined in JavaScript

- Accessing an uninitialized variable returns undefined.

let str;
console.log(str); //undefined

let str; str = "akash";
console.log(str); //akash

"define" = undefined

undefined = undefined

* null in JavaScript

- null means 'no value' assign to variable

- typeof null returns 'object'

- Null is treated as false value.

var details = [22, 'akash', true, null, {}]

console.log(typeof(details)); //object

* Javascript string - important of banish *

- strings - are used to store textual form of data like word, sentence. It follows zero based indexing.

let userName = "Akash";

let userName = 'patil';

let role = `front-end-developer`;

Get rid of white space in string using trim -

* Javascript string method list log -

trim()

slice()

charAt()

toString()

concat()

substring()

indexOf()

toUpperCase()

lastIndexOf()

toLowerCase()

* operators

- relational operators

($<$) , (\leq) , ($=!$) , (\geq) , ($>$) , (\neq)

* Arithmetic operators - ($+$) , ($-$) , ($*$) , ($/$) , ($\%$)

arithmetic operators add, subtract, multiply, divide, modulus

* Assignment operator - ($=$) , ($+=$) , ($-=$) , ($\times=$) , ($\div=$)

$Lsv = Rsv$ - assignment statement

a = b

$a + b$ means $a = a + b$

(2) always true

* Equality operator - (==) , (!=)

(==) - didn't check datatype (equal to)

(==) - equal value and equal type.

$81 == 800 \rightarrow \text{false}$

"flow of digits to my" $\rightarrow (81 < 800) \rightarrow \text{true}$

* Concatenation operator - ($+$)

Example

`var a=10`

`console.log("Hello " + a); | console.log('Hello ${a}');`

Template literal ↓

* Comparison operator -

($>$) , ($<$) , (\geq) , (\leq) , (\neq) , ($!$)

- ! - return true when condition become false
- ! - return false when condition become true

* Ternary operator - ($?$) alternate to if else

(condition ? " " :)

for Example (C)

```
(age >= 18) ? printf("can vote") : printf("cannot vote");
```

Ex .

```
let age = 15
```

```
let result = (age > 18) ? "You are eligible to vote." :
```

"You are not eligible to vote yet";

```
console.log(result);
```

* Logical operators - int max and min + + alignment

① (ff) → logical and

② (11) → logical or

③ (!) → logical not

* Bitwise Operators.

(&) → AND : (2, 5) \rightarrow 2 & 5 = 10

(|) → OR

(~) → NOT

(^) → XOR

(<<) → left shift

(>>) → right shift

(>>>) → unsigned right shift

* Exponentiation operator - $\text{2}^{(\ast\ast)}$; $\text{C}^{(\ast\ast)}$ +

Example :-

let $x = 10;$

base longint $\leftarrow (10)$;

let $z = x \ast\ast 2$ // result is $100 \leftarrow (11)$;

base longint $\leftarrow (1)$;

Example :-

let $a = 5;$

• operators and *;

let $b = \text{math}.\text{pow}(a, 2);$ // result \leftarrow is 25 ;

base $\leftarrow (1)$;

Term $\leftarrow (\sim)$;

ans $\leftarrow (N)$;

tuple $\leftarrow (\rightarrow)$;

tuple $\leftarrow (\leftarrow)$;

tuple $\leftarrow (\ll\ll)$;

* Conditional statements

- if statement

- use if to specify a block of code to be executed, if a specified condition is true.

for Example:

make a "good day" greeting IF the hour is less than 18:00

```
- if (hour < 18) {  
    greeting = "Good day";  
}
```

- The else statement

- use the else statement to specify a block of code to be executed if the condition is false

```
if (hour < 18)  
    greeting = "Good day"; } false
```

```
} else {  
    greeting = "Good evening"; }
```

* The else if statement

Use the else if statement to specify a new condition if the first condition is false.

Example :-

```
if (time < 10) {  
    greeting = "Good morning"; } False  
} else if (time < 20) {  
    greeting = "Good day"; }  
} else {  
    greeting = "Good evening"; }  
}
```

* The switch statement

Helps to check multiple conditions

For Example :-

```
let a = +prompt("Enter 1st NO", 0);  
let b = +prompt ("Enter 2nd NO", 0);
```

```
let input = prompt('choice operation - , + , * , / .');
```

```
switch (input) {  
    case '+':  
        console.log (`${a} + ${b}= ${a+b}`);  
        break;  
  
    case '-':  
        console.log (`${a} - ${b}= ${a-b}`);  
        break;  
  
    case '*':  
        console.log (`${a} * ${b}= ${a*b}`)  
        break;  
  
    default:  
        console.log ('wrong Data');  
        break;  
}
```