

Javascript.

Javascript :- client side scripting language (runs into user browser)

History - created By Brendan Eich at Netscape communications and initially they called it "Mocha" and later livescript before being renamed javascript

currently we are using ECMAScript 2015 (ES6)

In 2009 Ryan Dahl released node.js, which allowed javascript to run on server side, this helps javascript to run outside browser

Use-case :- it adds behavior and functionality in website

Note :- JS runs with help of browser's built JS engines, v8 (chrome), spider monkey (mozilla), chakra (edge)

Where to add JS in HTML?

① add just before closing body tag

```
<body>
<script>
    console.log("Hello")
</script>
</body>
```

② separate JS file and link in Head

```
<head>
<script src = "script.js"> </script>
</head>
```

Variable in JS - named memory locations

↳ create by using let, var, const

let a = 10

(exp need to define type)

Let v/s var :-

var is globally scoped, let local scope variable

constant :- can't be changed and not reassigned.

Variable naming rule - camel case style.

Datatype

① Primitive - string, number, Boolean, undefined (let a; console.log(a))
NULL (let a = null; \Rightarrow a doesn't have any val)
- symbol, BigInt

② Reference type - objects (have property & behavior)
Array, functions, date, Regular exp, map & set

dynamic type lang - JS is dynamic typed lang. means that variable types are not explicitly defined and can change at runtime

let a = 42 \rightarrow number literal
a = "Hello" \rightarrow string literal
a = true \rightarrow boolean literal

Operators

① Arithmetic - (+, -, *, /, %, **, \rightarrow modulo \rightarrow exponentiation operator)

② Assignment op - (++a, a--, +=, -=, /= ...)

③ Comparison - (>, <, >=, <=, ===, !==)

④ ternary - cond? val1 : val2;

⑤ logical - AND (&&), OR (||), NOT (!)

⑥ Bitwise op - Bitwise And (&), Bitwise OR (|)

ex - let a = 2, let b = 3, a & b? \Rightarrow
$$\begin{array}{r} 00000101 \leftarrow a \\ \& 00000011 \leftarrow b \\ \hline 00000010 \end{array}$$

Equality operator - $\overset{\text{loose}}{==}$ v/s $\overset{\text{strict}}{===}$
 \downarrow
compares only val \rightarrow compare val and type (Recommended)

operator precedence - use bracket wisely & BODMAS

let c = a + b * d / e;

let c = a + (b * d) / c \checkmark

Control statements:-

① IF-else

```
IF (cond) {  
  //  
}  
else IF (cond) {  
  //  
}  
else {  
  //  
}
```

②

switch case

```
let num = 2  
switch (num) {  
  case 1: //  
    break;  
  case 2: //  
    break;  
  default: //  
}
```

Rule:- Break - exits loop prematurely and start executing code just after loop

continue - used to skip the current iteration of a loop and proceed to next iteration

Loops - used for repetitive task

① for loop :-

```
for (let i = 0; i < 5; i++) {  
  console.log(i)  
}
```

② while loop -

```
let i = 0, n = 5  
while (i < 5) {  
  //  
  i++  
}
```

③ do-while loop -

```
let i = 0  
do {  
  console.log(i)  
  i++  
} while (i < 5);
```

↪ execute atleast once

④ for-in loop

- used to iterate over properties of an object, typically used with array or object to loop through their key or indices.

```
const obj = { a: 1, b: 2, c: 3 };
```

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

```
  console.log(key, obj[key]);  
}
```

⑤ for-of loop

- used with string, array, ...

```
const arr = [1, 2, 3];
```

```
for (let value of arr) {  
  console.log(value)  
}
```