

Q what does ECMA Script mean?

- ⇒
- ECMAScript : European Computer manufacturers association Script)
 - ECMA is a scripting language based on JavaScript. Invented by Brendan Eich at Netscape, ECMA Script made its first appearance in the Navigator 2.0 browser. It later started appearing in succeeding browser versions of Netscape as well as other browsers.
 - ECMA Script is widely used on the world wide web especially for Client-side Scripting.

Q what is first name of JavaScript and why JS name mocha JavaScript?

- ⇒
- JavaScript is a scripting language. JavaScript was originally named Mocha, later it was renamed to LiveScript and then to JavaScript. The LiveScript to JavaScript name change came because of Netscape and Sun did a license agreement. The language was then submitted for standardization to the ECMA International organization. By that time, Netscape didn't allow the use of JavaScript name so the standardized language is named ECMAScript.
- JavaScript isn't actually an open name. Now it's a trademark of Sun (now Oracle)

Q Explain release version history of JavaScript with current version?

- ⇒
- JavaScript was invented by Brendan Eich in 1995, and became an ECMA Standard in 1997
 - ECMAScript is the official name of the language.
 - ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, & ES6
 - Since 2016, versions are named by year.
(ECMAScript 2016, 2017, 2018, 2019, 2020)

9 Difference betn Framework & Library

Framework	Library
<ul style="list-style-type: none">i) predefined code along with predefined process.ii) To write a code according to framework and tells where to write.iii) According to process code is written.iv) eg. Angular is framework developed by google <u>others</u> NodeJS & vue	<ul style="list-style-type: none">i) It is nothing but predefined code.ii) Developer can write a code independantly.iii) In library coding is already in written format.iv) React is a library developed by Facebook or meta <u>others</u> Jquery, AJAX

Q Difference betw var, let & Const with example?

Var	let	Const
<p>① If we are declared a variable from var then we can also declare it again with the same name, and if we want to re-assign its value.</p> <p>② It is function scope</p> <p>③ eg.</p> <pre> var myname = "Ratnesh"; console.log(myname); var myname = "golu"; console.log(myname); </pre> <p>→ redeclared & change value</p> <p>→ output: Ratnesh</p> <p>→ output: golu</p>	<p>① If we declare a variable with let then we cannot declare it again with the same name but can reassign its value</p> <p>② It is block scope</p> <p>③ eg.</p> <pre> let name = "Ratnesh"; console.log(name); let name = "golu"; console.log(name); </pre> <p>→ output: Ratnesh</p> <p>→ error: cannot declare again with same name but value change कर सकते हैं।</p>	<p>① if we declare variable with const then we can neither declare it again nor can reassign its value</p> <p>② It is a block scope</p> <p>eg. const = "Ratnesh"</p> <pre> const = "Ratnesh"; console.log(const); const = " </pre> <p>③ const name = "Ratnesh";</p> <pre> const name = "Ratnesh"; console.log(name); const name = "golu"; console.log(name); </pre> <p>→ output: Ratnesh</p> <p>→ error: not redeclared same name & its value</p>

Q what is variable?

↳ Name of the memory locations where we store data that is called Variable.

• Rules to write variable:

① Camel Case: eg. rollNumber, fullName

↓
first letter small second letter is capital

② Name should not start with Numbers & Special Symbol

eg. ① ~~1stName~~ ② ~~#Name~~
1Name X

③ How to write variable?

~~let a = 10;
b = 20;~~

~~c = a + b;~~

let math = 50;

let sci = 30;

total = math + sci;

console.log(total);

Rightway
to write

② Block Scope :-

↳ Syntax of block

```
if (true) {  
    // ...  
}
```

↑ parenthesis

eg. ① `if (true) {`

`var i = 10;`

`console.log(i + "
");`

`}`

// local variable
∴ local variable ki scope within block hoti hai or output 10 se outside block se access hoti hai

③

`console.log(i + "
");`

↳ output = 10

// outside block se access hoti hai

② Local Scope:

- ↳ A variable that is declared inside a function defn is local.
- ↳ It is created and destroyed everytime the function is executed and it cannot be accessed by any code outside the function.
- ↳ If a variable has not been declared with var it is created as global variable.
- ↳ If the function is called again the variable is accessed from the outside function, new function & block.

eg.

① function add(b) {

var a = 10; // local variable
return(a+b);

}

console.log(add(20));

② function add(b) {

a = 10; // global variable
return(a+b);

}

console.log(add(20));

③ function show() {

var j = "get local variable";
console.log(j + "
");
}

Call show();

* if (true) {

console.log(j + "
");

}

output: j is not defined
not accessible
(error)

// block is run

* console.log(j + "
"); // outside access

output: error (outside function access not allowed because it is a local variable)

* function disp() {

console.log(j + "
");

}

disp()

output: error (new function is not some local variable not run not error)

// new function access

Scope

① Global Scope :- a variable that is declared outside a function definition is a global variable, and its value is accessible and modifiable throughout your program.

- global variable ka scope global रहता है।
- global variable program में कहीं से भी access कर सकते हैं।
like ① inside function से ② outside function ③ New function से ④ block से भी access कर सकते हैं।

Eg. ① variable को function के अंदर से access कर रहे हैं।

```
var j = "write a global variable";
```

```
function fun1() {
```

```
    console.log(j + "<br>");
```

```
}
```

```
fun1();
```

// inside function से access कर रहे हैं।

↳ output :- write a global variable

```
console.log(j + "<br>");
```

↳ output :- write a global variable

// function के बाहर से variable को access कर रहे

// outside function variable access

⊗ Same variable को दूसरे function से access कर रहे हैं।

```
function write1() {
```

```
    console.log(j + "<br>");
```

```
}
```

```
write1();
```

// New function से variable access कर सकते हैं।

↳ output :- write a global variable

* Variable को block (eg. if-else, if,) में भी access कर सकते हैं।

```
if (true) {
```

```
    console.log(j + "<br>");
```

```
}
```

// Block से variable access

↳ output :- write a global variable