**01/07/2024**

JavaScript

1995 founder of java script Brendan eich

Mocha first name

Live script second name

Java Script third name

Other name ecma script

Ecama script 7 we read

|  |
| --- |
| Script  Server side |
| Web use -🡪react js  Android application use -🡪react native for  Game use |
| What is ecma? |
| Console is object and log is function  . Is access member operator. |

**02/07/2024**

|  |
| --- |
| Variable declaration in JavaScript using keyword  var,let,const  var🡪var me value of reassign kr sakte and redeclare bhi kr sakte hai  drawback to when reassign the value the change the var value;  let 🡪let ko redeclare nhi kr sakte hai lekin reassign kr sakte hai  let b=20;  let b=101; invalid  b=1055; valid  const🡪 const me value fixed rahati hai esme value ko reassign aur redeclare nhi kr sakte hai |
| Rules of identifier  It should be start lowercases or uppercase character ;  It should not be start integer;  It should not be allowed to space; |
| let a=5;  let b=5;  // <!-- //write a porgram to add two number and print in alert box -->  // alert(a+b);  //write a  porgram to find cube of a number and print on html page  document.write("cube = ",a\*a\*a); //html document pr  In javascript in relational operator  === value ke sath data type ko compareJavaScript Operators With Examples - Dot Net Tutorials  karta hai |

**03/07/2024**

|  |
| --- |
| Data type  Primitive and Non-premitive data types |
| |  |  | | --- | --- | | Primitive data type | Non-Premitive data type | | | Type | typeof return value | Object wrapper | | --- | --- | --- | | [Null](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#null_type) | "object" | N/A | | [Undefined](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#undefined_type) | "undefined" | N/A | | [Boolean](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#boolean_type) | "boolean" | [Boolean](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Boolean) | | [Number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#number_type) | "number" | [Number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number) | | [BigInt](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#bigint_type) | "bigint" | [BigInt](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/BigInt) | | [String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#string_type) | "string" | [String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String) | | [Symbol](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#symbol_type) | "symbol" | [Symbol](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Symbol) | | Array  String  Function  Object   | **Data Type** | **Explanation** | | --- | --- | | Object | It is an instance that allows us to access  members in JavaScript. | | Array | It represents a collection  of similar elements in JavaScript. | | RegExp | It represents a regular  expression in JavaScript. | | |  | | |

**04/07/2024**

|  |
| --- |
| Nested if-else |
| If(cnd){  If(cnd){  }  else  {//code  }  }  else  {  } |
|  |

**02/07/2024**

|  |
| --- |
| Loop in javascript  What is loop  Loop is work for any work many time so we use loop   1. Do-while 2. For loop 3. For in 4. For of 5. For each (important) |
| do-while  we are using do while when we want to print message 1 times so we use it.  Syntax  do  {  //code  }while(condition); |
|  |

**08,09/07/2024**

|  |
| --- |
| While loop  While(condition){  //code  //increment or decremeny  } |
| For loop  For(initialization;condition;increment or decrement)  {  //block of code  } |
| For in loop   |  | | --- | | skip | |

**11/07/2024**

|  |  |
| --- | --- |
| Array  Array is collection of multiple data type.  Syntax   |  | | --- | | var arr=[elements,elements]  elements=number,string,char; |   Why is indexing start with 0; |

**12/07/2024**

|  |
| --- |
| For of loop  No need to indexing in for of loop  For of loop print array element without indexing |
| For(let variable of arrayName)  {  //block of code  } |
| let arr=[1,2,3,4,5,6,7,8,8]  for(let a of arr)      {            console.log(a);      } |

|  |
| --- |
| For in loop  We use for in loop in object  It is use to print index |
| For(let variable of arrayName)  {  //block of code  } |
| let arr=[1,2,3,4,5,6,7,8,8]  for(let a in arr)      {            console.log(a);      } |

**13/07/2024**

|  |
| --- |
| What is function  Function is reusable  Function is time-saving  In function we can debug code esay.  Function have block of code  Function of two type  1)user defined funton  2)pre-defined function |
| Coming to pre-defined function   |  | | --- | | push(element)  when we want to add element in last then we can use push() function.  We can add multiple element push(element,element,element) | | let arr = [1, 2, 3, 4, 5];  arr.push(6);  arr.push(7);  arr.push(8);  for (let i = 0; i < arr.length; i++) {      console.log(arr[i]);  }    Output  1  2  3  4  5  6  7  8 | |

|  |
| --- |
| pop() function  when we want to remove element in last then we can use pop() function.  We can remove multiple element using pop(element,element,element) |
| let arr = [1, 2, 3, 4, 5];  arr.pop(); // Removes the last element from arr  for (let i = 0; i < arr.length; i++) {      console.log(arr[i]);  }  ouput  1  2  3  4 |
| unshift() function  when we want to add element in starting then we can use unshift() function.  We can add multiple element using unshift(element,element,element,……..) |
| let arr = [1, 2, 3, 4, 5];  arr.unshift(5,7);  for (let i = 0; i < arr.length; i++) {      console.log(arr[i]);  }  Output  Ouput  5  7  1  2  3  4  5 |

|  |
| --- |
| shift() function  when we want to remove element in starting then we can use shift() function.  We can remove multiple element using shift(element,element,element) |
| let arr = [1, 2, 3, 4, 5];  arr.shift(); // Removes the first element from arr  for (let i = 0; i < arr.length; i++) {      console.log(arr[i]);  } |
| // ouput  // 2  // 3  // 4  // 5 |

|  |
| --- |
| **slice()**  I want to remove specific element then I can use slice() function  slice function need to remove element index.  slice(start index,end index) |
| let arr = [1, 2, 3, 4, 5];  arr = arr.slice(2, 4);  console.log(arr); // Outputs: [3, 4] |
| Output  3  4 |

**15/07/2024**

|  |
| --- |
| Splice(index,No. of element) |
| Particular index ko add krana or remove krna |
| let arr = [1, 2, 3, 4, 5,6];  arr.splice(2, 3);  console.log(arr);  output  1  2  5  6 |

|  |
| --- |
| splice(start, removeCount, newItem, newItem, newItem, ...) |
| Particular index ko add krana or remove krna |
| let arr = [1, 2, 3, 4, 5,6];  arr.splice(2, 3,15,20);  console.log(arr); // Outputs: [3, 4]  output  1  2  15  20  6 |

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | arguments | await\* | boolean |
| break | byte | case | catch |
| char | class\* | const\* | continue |
| debugger | default | delete | do |
| double | else | enum\* | eval |
| export\* | extends\* | false | final |
| finally | float | for | function |
| goto | if | implements | import\* |
| in | instanceof | int | interface |
| let\* | long | native | new |
| null | package | private | protected |
| public | return | short | static |
| super\* | switch | synchronized | this |
| throw | throws | transient | true |
| try | typeof | var | void |
| volatile | while | with | yield |

Removed Reserved Words

The following reserved words have been removed from the ECMAScript 5/6 standard:

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | boolean | byte | char |
| double | final | float | goto |
| int | long | native | short |
| synchronized | throws | transient | volatile |

**16/07/2024**

Object- object is userdefined datatype in javascript.

Object is denoted by curlybraces in javascript,we can store multiple type of value in a javascript.

. = access member operator

Let obj={key:value,key:value,………..}

|  |
| --- |
| let emp={name:"Shani",id:101,age:22,sal:10050};  console.log(emp.name);  console.log(emp.id);  console.log(emp.age);  console.log(emp.sal); |
| Output  Shani  101  22  10050 |
|  |
| let emp={name:"Shani",id:101,age:22,sal:10050};  for(let k in emp)      console.log(k); |
| name  id  age  sal |

|  |
| --- |
| let emp={name:"Shani",id:101,age:22,sal:10050};  for(let k in emp){      console.log(emp[k]);  } |
| Value print karega |

**17/07/2024**

|  |
| --- |
|  |

**18/07/2024**

|  |
| --- |
| let obj = {      name: "shanideval",      age: 45  }  let obj1 = {      laction: "Bhopal"  }  Object.assign(obj, obj1)  console.log(obj)  delete obj.age;  console.log(obj) |
| Function is a block of code.  It is reusable.  Function is keyword.  We can debug function easily. |
| Function function\_name(parameter/argument){  //code  }  Function declaration  Function definition  Function calling.(identity(para/argu)) |
|  |

**18/07/2024**

|  |
| --- |
| function tbl( n) {     return n\*n\*n;  }  let n = parseInt(prompt("Enter any number:"));  console.log(tbl(n)); |
|  |

**22/07/2024**

|  |
| --- |
| Arraow function is use to one line function  Syntax  Type 1  () => {return }  Type 2  let b=(a)=>(a\*a) |
| let a=10;  let b=(a)=>{return a\*a};  console.log(b(a)); |
| let a=parseInt(prompt("Enter a number "))  let b=(a)=>(a\*a);  console.log(b(a)) |

**22/07/2024**

 **Function Declaration**

Javascript

|  |
| --- |
| function greet() {  console.log("Hello, World!");  } |

 **Function Expression**

javascript

|  |
| --- |
| const greet = function() {  console.log("Hello, World!");  }; |

 **Arrow Function**

javascript

|  |
| --- |
| const greet = () => {  console.log("Hello, World!");  }; |

 **Immediately Invoked Function Expression (IIFE)**

javascript

|  |
| --- |
| (function() {  console.log("Hello, World!");  })(); |

 **Anonymous Function**

* These are functions without a name, often used as arguments to other functions.

javascript

|  |
| --- |
| setTimeout(function() {  console.log("Hello, World!");  }, 1000); |

 **Named Function Expression**

javascript

|  |
| --- |
| const greet = function sayHello() {  console.log("Hello, World!");  }; |

 **Constructor Function**

javascript

|  |
| --- |
| function Person(name, age) {  this.name = name;  this.age = age;  }  const person1 = new Person('John', 30); |

 **Method in an Object**

javascript

|  |
| --- |
| const person = {  name: 'John',  greet: function() {  console.log("Hello, " + this.name);  }  };  person.greet(); |
|  |