Prompt

this function is used to take input value by user at run time , this function has two parameter message message and default value (optional parameter)

syntax - prompt(message,default value)

```
<script>
   var age=prompt("enter your age ",18)
   if (age>=18){alert("you can vote....")}
   else {alert("you cannot vote....")}
</script>
```

funcntion and event

Function - function is a block of code

syntax: function function_name (){ // js statement }

Arrow function= const var=()=>{}

Event - events are some specific type of situation and we can perform specific task , java script is an event handling programming lang which support various type of event

like onclick , onmousehover,onmouseout,onfocus,onblur,onchange ,onsubmit,onload,onunload

```
<button onmouseover="display()" onclick="display()"> onclick or onmouseover </button> <br> <br>
<button onmouseleave="display()">onmouseleave</button>
<script>
        function display(){ alert("hello world") }
</script>
```

Stop( ) and start() function

```
<!-- scrollamount=== is used to manage the speed of marquee -->
<body>
<marquee scrollamount="10" onmouseover="this.stop()" onmouseout="this.start()" >
 <h1> welcome to my world </h1>
</marquee> </body>
```

form

```
<script> function getvalue() {
    var x = document.f1.name.value;
    var y = document.f1.city.value;
    alert("my name is :" + x + "city :" + y);
  } </script>
</head>
<body>
 <form name="f1" method="post" action="save.html" onsubmit="getvalue();">
  name <input type="text" name="name" /><br />
  city <input type="text" name="city" />
  <input type="submit" value="save" /> </form>
```

File save.html

```
<body bgcolor="red">
<h1>save</h1>
  </body>
```

form validation

```
<script>
    function validation() {
      var x = document.f1.name.value;
      var y = document.f1.city.value;
      if (x == "") {
        alert("enter name");
        document.f1.name.focus();
        return false; }

      if (y == "") {
        alert("enter city");
        document.f1.city.focus();
        return false; }
    }
  </script>
 </head>
 <body>
  <center>
  <h1> application  form </h1>
  <form name="f1" method="post" action="save.html" onsubmit=" return validation();">
   name <input type="text" name="name" /><br />
   city <input type="text" name="city" />
   <input type="submit" value="save" />
  </form>
```

javascript variable

4 way to declare a js variable
using var ,let , const , nothing

what is variable ?
variable are container for storing data

when to use js variable
always declare js variable with var,let,const
the var keyword is used in all js code from 1995-2015
the let and const keyword were added to js in 2015
if you want your code to run in older browser ,you must use var

let
the let keyword was introduced in ES6 2015
variable defined with let cannot be redeclared
variable defined with let must be declared before use
variable defined with let have a block scope
const
the const keyword was introduced in ES6 2015
variable defined with const cannot be redeclared
variable defined with const cannot be ressigned
variable defined with const have a block scope

Ecma International (formally European Computer Manufacturers Association)
es6

```
<script>
        var a=10        //dec
        console.log(a)
         var a=50      //redec
         console.log(a)
       let  b=10        //dec
       console.log(b)
    </script>
  <script>
    let a;
    console.log(a) //undefined

    let b=null;
    console.log(b) //null

    let c=null;
    console.log(c) //null
  </script>
 <script>
    const pi=3.14
    console.log(pi) //3.14  </script>
```

```
    <script type="text/javascript">
        let msg="welcome to cybrom bhopal!!";
        msg+="we are cybrom student!!!";
        msg+="we are web developer!!!";
        document.write("<h1>"+ msg + "</h1");
    </script>
output:welcome to cybrom bhopal!!we are cybrom student!!!we are web developer!!!
```

```
<body>
    enter name <input type="text"  id="name"><br>
    enter city <input type="text" id="city"><br>
    <button onclick="c()">click me</button>
    <h1 id="ans"></h1>
    <script>
    function c(){
     var name=document.getElementById("name").value;
     var city=document.getElementById("city").value;
    document.getElementById("ans").innerHTML="my name is " + name +"," + "my city is :" +city
     }
    </script>
</body>
```

enter name amarth
enter city vidisha
click me

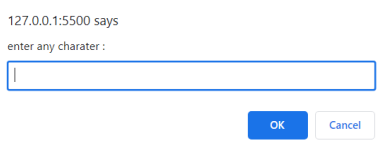
my name is amarth,my city is :vidisha

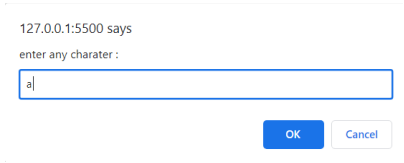
<u>If ,else if , else statement</u>

syntax:

```
if (){
   //statement
}

else  if(){
   //statement
}

else
{
//statement
}
```

```
<script>
    var a=prompt("enter any charater : ")
    if(a=='a'||a=='A'){
        document.write("<h1> this is vowel  ")
    }

     else if(a=='e' || a=='E'){
        document.write("<h1> this is vowel ")
     }

     else if(a=='i' || a=='I'){
        document.write("<h1> this is vowel ")
     }
     else if(a=='o' || a=='O'){
        document.write("<h1> this is vowel ")
     }
     else if(a=='u' || a=='U'){
        document.write("<h1> this is vowel ")
     }

    else{
        document.write("<h1> this is consonent")
    }

</script>
```





Output: this is vowel

---

switch statement

```
switch(){
   case level1:
   //statement
   break;
   case level2:
   //statement
   break;
   default:
   //statement
   }
```

```html
<body>
  enter number 1 <input type="text" id="n1" /><br />
  enter number 2 <input type="text" id="n2" /><br />
  enter choice <input type="text" id="ch" />(-,+,*,/) <br />
  <button onclick="myFunction()">click me</button>
  <h1 id="ans"></h1>
  <script>
    function myFunction() {
      var n1 = parseInt(document.getElementById("n1").value);
      var n2 = parseInt(document.getElementById("n2").value);
      var ch = document.getElementById("ch").value;

      switch (ch) {
        case "+":
          document.getElementById("ans").innerHTML = n1 + n2;
          break;
        case "-":
          document.getElementById("ans").innerHTML = n1 - n2;
          break;
        case "*":
          document.getElementById("ans").innerHTML = n1 * n2;
          break;
        case "/":
          document.getElementById("ans").innerHTML = n1 / n2;
          break;
        default:
          document.getElementById("ans").innerHTML = "invalid";
          break;
      }
    }
  </script>  </body>
```

enter number 1 10

enter number 2 10

enter choice - (-,+,*,/)

click me

0

Looping statement in JS :-

Looping statement are iterative statement ,they repeat statment or group of statement accordind to condition .
js provide various type of looping statement

1.for loop
2.while loop
3.do while loop
4.for in loop
5.for of loop
6.foreach

1.for loop
syntax:

```
for(initilization;condition;inc/dec)
for (let i=0;i<10;i++){
   // statment
}
```

2.while loop

```
while(condtion){

   // statment
   a++ ,a-- inc or dec
}
```

3.FOR IN LOOP : FOR IN STATEMENT LOOP THORUGH THE PROPERTIES OF AN OBJECT
SYNTAX :

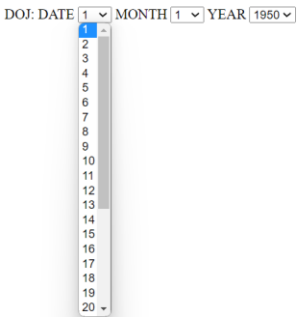
```
FOR(KEY IN OBJECT){
   // statement
.
}
```

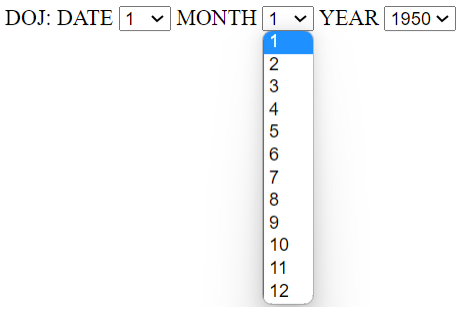
4. for of loop : for of statement loops though the value of an iterable object
it lets you loop over iterable data structures such as arrays, string,maps,nodelist,and more

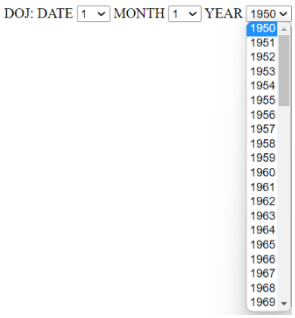
syntax :

```
for(value of iterable){
   // statement
}
```

```
<body>
    DOJ:
    DATE
    <select>
        <script>
            for (let i = 1; i <= 31; i++) {
                document.write("<Option>" + i + "</Option>")
            }
        </script>
    </select>
    MONTH
    <select>
        <script>
            for (let i = 1; i <= 12; i++) {
                document.write("<Option>" + i + "</Option>")
            }
        </script>
    </select>
    YEAR
    <select>
        <script>
            for (let i = 1950; i <= 2023; i++) {
                document.write("<Option>" + i + "</Option>")
            }
        </script>
    </select>
</body>
```







```
    <H1>FOR IN LOOP</H1>  for in loop return key
     <script>
        // object
        const student={
            "roll_number":1000,
             "name":"amarth",
             "city":"bhopal",
             "fees":45000
        }
        for (const key in student) {
           document.write(key+" ,")
         }
    </script>    output: roll_number ,name ,city ,fees ,
```

```
    <script>
        // object
        const student={
            "roll_number":1000,
             "name":"amarth",
             "city":"bhopal",
             "fees":45000       }
         for (const key in student) {
           document.write("<h1>"+key + " : "+ student[key] +"</h1>" )
         }
    </script>
Output:
roll_number : 1000
name : amarth
city : bhopal
fees : 45000
```

```
<script>
    //array
    const name=["raj","amarth","patel","hello"]
    console.log(name[0])

    for (var keey in name) {
     document.write(keey+name[keey]+" ")  output: 0raj 1amarth 2patel 3hello
    }

    var a="welcome"
    for (var v in a){
        console.log(v) //012345
    }
</script>
```

```
For of loop
    <script>
        const a = ["a","b" ,"c"  ];
        for(var i of a){
            document.write( a )  //a,b,ca,b,ca,b,c
        }
    </script>

    <script>
        const name="amarth"
        for (var i of name) {
            document.write("<h1>" , i , "</h1>")
        }
    </script>
```

a
m
a
r
t
h

```
While loop :
<body>
enter any number <input type="text" id="aa">
<button onclick="run();">click</button>
    <script>
       function run(){
         var a = parseInt(document.getElementById("aa").value);
         var i=1;
            while (i<=10) {
                document.write(i * a+” ”) // 4 8 12 16 20 24 28 32 36 40
                i++
            }
        }
    </script>
</body>
```

Functions in JS:--

- functions are used to perform specific task, they are program or sub program.
- to create a function in js we can use function keyword follow with function name.
- Functions are the building blocks of JS.

syntax:--

```
function functionName(){
   //code to be executed
   }
```

- functionName is the name of the function.
- () is the parameter list.
- {} is the function body.
- functionName() is the function call.
- functionName is the identifier for the function.
- functionName() is the invocation of the function.

--js functions also have parameters. parameters are the arguments pass inside the paranthesis.

syntax :--

```
function functionName(parameter1,parameter2,parameter3...){
   //statements
}
```

JS function also return a value. To return a value by function we can use return keyword.

syntax:--

```
function functionName(parameter1,parameter2,parameter3...){
   //statemnts
   return value;
   }
```

```
<script>
function fun1(){
 document.write("<h1> welcome to cybrom !</h1>")
 }
 document.write("<h3> welcome to bhopal !</h3>")
 fun1();
document.write("<h3> welcome to bhopal !</h3>")
fun1();   </script>
```

welcome to bhopal !

welcome to cybrom !

welcome to bhopal !

welcome to cybrom !

```
<head>
    <script>
        function myFunction(fnm,snm) {
            document.write("<h1> my name is "+fnm+"  "+snm+ "  from bhopal!</h1>")
        }
    </script>
</head>
<body>
    <script>
        myFunction("sudhanshu","kumar");
        myFunction("sunil","thakur");
    </script>
</body>
```

my name is sudhanshu kumar from bhopal!

my name is sunil thakur from bhopal!

```
    <script>
        function mysqr(myno) {
            var ans=myno*myno;
            return ans;
        }
        function myadd(no1,no2) {
            var add=no1+no2;
            return add;
        }
        function mycube(no1) {
            var cube=no1*no1*no1;
            return cube;
            }
    </script>
</head>
<body>
    <script>
        var myans=mysqr(3);
        document.write("<h1> square is : ",myans,"</h1>"); //9
        var myans1=myadd(333,454);
        document.write("<h1> add is : ",myans1,"</h1>"); //
        var myans2=mycube(35);
        document.write("<h1> cube is : ",myans2,"</h1>");

    </script>
</body>
```

square is : 9

add is : 787

cube is : 42875

arrow function

arrow function were introduced in 2015
before arrow function
hello=function(){ return "hello wolld";}

with arrow function
hello=()=>{ return "hello wolrd"}

arrow function return value by defalut:
hello=()=>"hello wolrd";
note:this works only if the function have only one parameter

if you have paramter , you can pass inside the parentheses
arrow function with parameter
hello=(val)=>"hello"+ val

infact if you have only one parameter , you can skip parentheses as well.
arrow function without parentheses
hello=val=>"hello"+ val;

```
<body>
<h1 id="demo"> welcome to cyborm </h1>
<button onclick="display()">click me</button>
<script>
    // old function
function display(){
        myname=function (){return "hello world"}
        document.getElementById("demo").innerHTML=myname();}

        // arrow function
function display2 (){
     const name=()=>{return "hello world"}
    document.getElementById("demo").innerHTML=name();}

//this run only if arrow function have only one parameter
function display3 (){
    const  name3=()=>"hello world";
    document.getElementById("demo").innerHTML=name3();
}

 // arrow function with parameter
function display4 (){
    const  name4=(name,surname)=>"hello world"+" "+name+" "+surname;
    document.getElementById("demo").innerHTML=name4("amarth","patel");}
```

```
 // arrow function with single parameter you can leave parantheses
 function display5 (){
    const  name5=name=>"hello world"+" "+name;
    document.getElementById("demo").innerHTML=name5("amarth");
}
</script>
```

array.

an array us a special variable ,which can hold more than one value
syntax:
const array_name=[va1,val2,val3.....]

we can access array variable value by using its index number like 0,1,2,3....

in js array value can be same or mixed type.

the length properties is used to find the length of array

```
<script>
const name=["raaju","ramu",111 ,12,10544,10.12];
document.write("<h1>", name.length , "</h1>") //6
document.write("<br>")
document.write(name[0])  //raaju
document.write("<br>")
document.write(name[3]*10)  //120
</script>
```

array method

tostring() - coverting array into string tostring() coverts an array to a string of(comma separated) array values

```
    <h1 id="demo1"> welcome </h1>
    <h1 id="demo2"> welcome </h1>
    <button onclick="display()"> click </button>
   <script>
function display() {
const name=["raj","mohna","pankay","ramu"]
document.getElementById("demo1").innerHTML=name +" "+ typeof(name);//raj,mohna,pankay,ramu object

//tostring()
const myname=name.toString();
document.getElementById("demo2").innerHTML=myname +" "+ typeof(myname);//raj,mohna,pankay,ramu string
</script>
```

join() method also joins all array element into a String. it behave just like toString() but the in addition you can specify separator.

```
<body>
    <h1 id="demo1"> welcome </h1>
    <h1 id="demo3">welcome</h1>
    <button onclick="display()"> click </button>
    <script>
function display() {
const name=["raj","mohna","pankay","ramu"]
document.getElementById("demo1").innerHTML=name +" "+ typeof(name);//raj,mohna,pankay,ramu object
//join
const mynamee=name.join("**");
document.getElementById("demo3").innerHTML=mynamee +" "+ typeof(mynamee); //raj**mohna**pankay**ramu string
        }
    </script>
```

pop() method removes the last element from an array

```
<h1 id="demo4">welcome</h1>
<h1 id="demo5">welcome</h1>
<button onclick="display2()"> click </button>
<script>
//pop()
 function display2() {
  const name=["raj","mohna","pankay","ramu"]
 name.pop();
 a=name.pop()
 document.getElementById("demo4").innerHTML=name;  //raj,mohna,pankay
 document.getElementById("demo5").innerHTML=a;  //pankay
        }
</script>
```

```
shift()
remove the first arry element and shifts all other element to a lower index -->

<h1 id="demo4">welcome</h1>
<h1 id="demo5">welcome</h1>
<button onclick="display2()"> click </button>
<script>
 function display2() {
        const name=["raj","mohna","pankay","ramu"]
        var a=name.shift("raj")
        document.getElementById("demo4").innerHTML=name;  //mohna,pankay,ramu
        document.getElementById("demo5").innerHTML=a;  //raj
      }
</script>
```

```
unshift()
add the first arry element and unshifts all other element at the beginning ( lower index) -->
<h1 id="demo4">welcome</h1>
<h1 id="demo5">welcome</h1>
<button onclick="display2()"> click </button>
<script>
 function display2() {
        const name=["raj","mohna","pankay","ramu"]
        var a=name.unshift("yaaa")
        document.getElementById("demo4").innerHTML=name;  //yaaa,raj,mohna,pankay,ramu
        document.getElementById("demo5").innerHTML=a;  //5
      }
</script>
```

```
js array delete is a command
warning
array elemet can be delete using the js operator delete
using delete leave undefined holes in array
use pop or shift() instead -->
<h1 id="demo4">welcome</h1>
<h1 id="demo5">welcome</h1>
<button onclick="display2()"> click </button>
<script>
 function display2() {
        const name=["raj","mohna","pankay","ramu"]

        document.getElementById("demo4").innerHTML=name; //raj,mohna,pankay,ramu
        delete name["0"];
        document.getElementById("demo5").innerHTML=name;  //,mohna,pankay,ramu
      }
</script>
```

```
concat()
create a new array by merging existing array
<h1 id="demo4">welcome</h1>
<h1 id="demo5">welcome</h1>
<button onclick="display2()"> click </button>
<script>
 function display2() {
        const name=["raj","mohna","pankay","ramu"]
        const name2=["seema","zoya"]
        const name1=["ddddddddeema","zoya"]
        const name3=name.concat(name2,name1,"my name is amarth")
        document.getElementById("demo4").innerHTML=name3;
//raj,mohna,pankay,ramu,seema,zoya,ddddddddeema,zoya,my name is amarth
      }
</script>
```

```
the splice() method adds new item to an array
<body>
  <body>
    <h1 id="demo4">welcome</h1>
    <h1 id="demo5">welcome</h1>
    <button onclick="display2()"> click </button>
    <script>
     function display2() {
            const name=["raj","mohna","pankay","ramu"]
            document.getElementById("demo4").innerHTML=name;
            name.splice(2,0,"hello","heyaa")
    document.getElementById("demo5").innerHTML=name; //raj,mohna,hello,heyaa,pankay,ramu
          }
    </script>
```

```
the slice() method  slices out a piece of an array
<body>
     <body>
         <h1 id="demo4">welcome</h1>
         <h1 id="demo5">welcome</h1>
         <button onclick="display2()"> click </button>
         <script>
          function display2() {
                     const name=["raj","mohna","pankay","ramu"]
                     document.getElementById("demo4").innerHTML=name;
                     var a= name.slice(2)
                     document.getElementById("demo5").innerHTML=a; //pankay,ramu
                 }
         </script>
```

```
    <script>

        const name=["raj" , "mohan" , "pankaj" , "seema" , "jalaj" ,"nitin"];
        document.getElementById('demo1').innerHTML=name; //raj,mohan,pankaj,seema,jalaj,nitin

        const myname=name.slice(2);    // slice array method
        document.getElementById('demo2').innerHTML=myname; //pankaj,seema,jalaj,nitin


    {    const name=["raj" , "mohan" , "pankaj" , "seema" , "jalaj","nitin"];
        document.getElementById('demo1').innerHTML=name;

        const myname=name.slice(2,4);
        document.getElementById('demo2').innerHTML=myname;//pankaj,seema}
</script>
```

```
Sort()

    <script>
         {
        const name=["raj" , "mohan" , "pankaj" , "seema" , "jalaj","nitin"];
        document.getElementById('demo1').innerHTML=name;

         name.sort();    // sort array method
        document.getElementById('demo2').innerHTML=name; //jalaj,mohan,nitin,pankaj,raj,seema
    }
</script>
```

```
Reverse ()
    <script>
    {       const name=["raj" , "mohan" , "pankaj" , "seema" , "jalaj","nitin"];
            document.getElementById('demo1').innerHTML=name;

             name.sort();    // sort array method
            document.getElementById('demo2').innerHTML=name;
//jalaj,mohan,nitin,pankaj,raj,seema

             name.reverse();    //  Revers array method
            document.getElementById('demo3').innerHTML=name;
//seema,raj,pankaj,nitin,mohan,jalaj
        }

</script>
```

```
For each
    <script>
    {
         {
        var tot=0;

        const num=[2,3,4,5,6,7,8,9,10];
        num.forEach(mycal);  //for each() method
        function mycal(val)
        {
            tot=tot+val;
        }
        document.getElementById('demo2').innerHTML=tot; //56

 } }
</script>
```

```
    <script type="text/javascript">
        function Display()
        {  const name=["raj","nitin","aval","pamkaj","seema","sunita"];
            var txt="<ol>";
            name.forEach(Mylist);
            function Mylist(val)
            {
                txt+="<li>"+val+"</li>";
            }
            txt+="</ol>";
            document.getElementById('demo').innerHTML=name;
            document.getElementById('demo1').innerHTML=txt;
        }
    </script>
</head>
<body>
    <h1 id="demo">Welcome!!!</h1>
    <h1 id="demo1">Welcome!!!</h1>
    <button onclick="Display();">click here!!</button>
```

raj,nitin,aval,pamkaj,seema,sunita

1. **raj**
2. **nitin**
3. **aval**
4. **pamkaj**
5. **seema**
6. **sunita**

click here!!

Map() important
map is a method work in array ,we pass parameter as function , return new array
call bACK function
variable.map(function function_name(){ })

```
    <script type="text/javascript">
        function Display()
        {   const name=["raj","nitin","aval","pamkaj","seema","sunita"];
            const myname=name.map((val)=>"<li>"+ val + "</li>");
            document.getElementById("demo1").innerHTML=myname;
        }
    </script>
</head>
<body>
    <h1 id="demo1">Welcome!!!</h1>
    <button onclick="Display();">click here!!</button>
</body>
```

- raj
,
- nitin
,
- aval
,
- pamkaj
,
- seema
,
- sunita

click here!!

```
  <body>
    <h1>map function</h1>
    <button onclick="display()">click me</button>
    <h3 id="demo"></h3>

    <script>
      function display() {
        const name = ["raju", "mohan", "pankaj", "sandeep", "jalaj", "ranu"];
        let data = "<ul type='circle'>";
        name.map(function (val) {
          data += "<li>" + val + "</li>";
        });
        data =data + "</ul>";
        document.getElementById("demo").innerHTML = data;
      }
    </script>
  </body>
```

map function

click me

- raju
- mohan
- pankaj
- sandeep
- jalaj
- ranu

```
<h1>map function  </h1>
<button onclick="display();" >click me</button>
<p id="demo">welcome</p>

<script>
  function display() {
    const name = ["raju", "mohan", "pankaj", "sandeep", "jalaj", "ranu"];
    const myname  =name.map((keyy)=>"<h1>" + keyy +"</h1>" );
    document.getElementById("demo").innerHTML = myname;
  }
</script>
```

map function

click me

raju

,

mohan

,

pankaj

,

sandeep

,

jalaj

,

ranu

```
<h1>map function  </h1>
<button onclick="display();" >click me</button>
<p id="demo">welcome</p>

<script>
    // object of arrAY
  function display() {
    const name =[
    {"roll_no":120, "name":"amarth" ,"city":"bhopal" ,"fees":1000},
    {"roll_no":100, "name":"ramu" ,"city":"dilhi" ,"fees":1000},
    {"roll_no":999, "name":"salman" ,"city":"mumbai" ,"fees":00},
    {"roll_no":999, "name":"hello" ,"city":"ASSAM" ,"fees":990}
  ]
    const myname  =name.map((keyy)=>"<h1>" + keyy.roll_no +" "+ keyy.name+" "+keyy.city+
" "+keyy.fees +"</h1>" );
    document.getElementById("demo").innerHTML = myname;
  }
</script>
```

map function

click me

120 amarth bhopal 1000

,

100 ramu dilhi 1000

,

999 salman mumbai 0

,

999 hello ASSAM 990

```
  <body>
    <h1>map function  </h1>
    <button onclick="display();" >click me</button>
    <p id="demo">welcome</p>
    <script>
        // object of arrAY
      function display() {
        const name =[
        {"roll_no":120, "name":"amarth" ,"city":"bhopal" ,"fees":1000},
        {"roll_no":100, "name":"ramu" ,"city":"dilhi" ,"fees":1000},
        {"roll_no":999, "name":"salman" ,"city":"mumbai" ,"fees":00},
        {"roll_no":7894, "name":"HELLO" ,"city":"ASSAM" ,"fees":990}
      ]

var table="<table border='1' >";
    name.map((key)=>{
            table+="<tr>";
            table+="<td>"+key.roll_no+"</td>";
            table+="<td>"+key.name+"</td>";
            table+="<td>"+key.city+"</td>";
            table+="<td>"+key.fees+"</td>";
            table+="</tr>";
    }   )
    table+="</table>";
    document.getElementById("demo").innerHTML=table;

      }
    </script>
  </body>
```

map function

click me

120	amarth	bhopal	1000
100	ramu	dilhi	1000
999	salman	mumbai	0
7894	HELLO	ASSAM	990

filter ()
filter methods takes each element in an array and it applies a conditional statement against it.
if this condition return true , the element gets pushed to output array.
if condtion retuen false ,the element does not get pushed to output array.

```
var new_array=arr.filter( function callback (element) {
   //retuen true or false
})
```

This syntax for filter is similar to map,except the callback function should return true to keep th element or false otherwise . in the csllback only element is required.

```
<body>
<button onclick="display()">cilck me</button>
<h1 id="demo">Welcome</h1>
<h1 id="demo1">welcome.</h1>

<script>
   function display() {
      const num=[34,1,5,4,6,79,4,8,4,7,5,7,1,7,121,84,84,84]
      const  mynum=num.filter(function(val) {
         if (val%2==0){return true}
         else{ return false}
      })

      document.getElementById("demo").innerHTML=num
      document.getElementById("demo1").innerHTML=mynum            // 34,4,6,4,8,4,84,84,84
}
</script>
```

```
    <h1>with arrow function</h1>
<button onclick="display()">cilck me</button>
<h1 id="demo">Welcome</h1>
<h1 id="demo1">welcome.</h1>
<script>
    function display() {
        const num=[34,1,5,4,6,79,4,8,4,7,5,7,1,7,121,84,84,84]
        const  mynum=num.filter(val=>val%2==0)
        document.getElementById("demo").innerHTML=num
        document.getElementById("demo1").innerHTML=mynum
    }
</script>
```

reduce()
method reduce an array of value down to just one value.
to get the output value it run a reducer function on each element of an array

syntax
arr.reduce(callback,initial value)
the callback argument is a function that will be called once for every item in the array.
this function takes four argument ,but often only first two are used.

```
<button onclick="display()">cilck me</button>
<h1 id="demo">Welcome</h1>
<h1 id="demo1">welcome.</h1>
<script>
  function display() {
    const sal=[1000,1000,1000,1000,1000]
    const  mysal=sal.reduce(function(val,int){return val+int },0)
    document.getElementById("demo").innerHTML=sal
    document.getElementById("demo1").innerHTML=mysal
  }
</script>
```

```
<script>
    function display() {
        const sal=[1000,1000,1000,1000,1000]
        const  mysal=sal.reduce((val,int)=>{val+int},0)
        document.getElementById("demo").innerHTML=sal
        document.getElementById("demo1").innerHTML=mysal
    }
</script>
```

string

js string are for storing and manipulating text.
string is zero or more character written inside quotes

the solution ti avoid this mistake , is to use backlash escape character
the backlash (\) escape character turns special character into string character

code	result	description
\'	'	single quotes
\"	"	double quotes
\\	\	backlash

```
    <script>
        let s1="welcome"
        let s2='welcome'
        let s3="hello 'wolrd' "
        let s4= ' hello "wolrd"  '
        let s5= 'don\'t  ,  \'hello wolrd\''
        let s6= 'don\'t ,   \'hello\\wolrd\'    ,  hahahah\\hahaha    '
        let s7 ="hello boyssssssssssssss"
        document.write("<h1>",s1,"</h1>")
        document.write("<h1>",s2,"</h1>")
        document.write("<h1>",s3,"</h1>")
        document.write("<h1>",s4,"</h1>")
        document.write("<h1>",s5,"</h1>")
        document.write("<h1>",s6,"</h1>")
        document.write("<h1>","length function :",s7.length,"</h1>")
    </script>
```

welcome

welcome

hello 'wolrd'

hello "wolrd"

don't , 'hello wolrd'

don't , 'hello\wolrd' , hahahah\hahaha

length function :24

extracting string parts

there are 3 methods for extracting a part of a string:

slice(start,end) , substring(start,end),substr(start,length)

```
<body>
  <h2>JavaScrip  String Methods</h2>
  <p id="demo"></p>
  <script>
    var str = "Hello we are cybrom student";
    //document.getElementById("demo").innerHTML = str.slice(5);       //we are cybrom student
    //document.getElementById("demo").innerHTML = "<h1>"+str.slice(0,5)+"</h1>";  //Hello
   // document.getElementById("demo").innerHTML = "<h1>"+str.substring(0,5)+"</h1>";  //Hello
    document.getElementById("demo").innerHTML = "<h1>"+str.substr(5,8)+"</h1>";  //we are
    </script>
```

replace:

```
1.The replace() method replace a specified value with another value in a string. return new
string
2.Replace method only replace first match.
3.To replace all match use a regular expression with /g flag (global match)
4.To replace case insensitive ,use a regular expression with /i flag
<body>
    <h2>JavaScrip  String Methods</h2>
    <p id="demo"></p>
    <script>
        var str = "Hello we are cybrom student  cybrom ";
        var str1 = "Hello we are cybrom student  CYBROM ";
 //document.getElementById("demo").innerHTML =
"<h1>"+str.replace("cybrom","amarth")+"</h1>";  //Hello we are amarth student cybrom


// document.getElementById("demo").innerHTML =
"<h1>"+str.replace(/cybrom/g,"amarth")+"</h1>";  //Hello we are amarth student amarth


document.getElementById("demo").innerHTML =
"<h1>"+str1.replace(/cybrom/gi,"amarth")+"</h1>";  //Hello we are amarth student amarth

        </script>
```

Converting to upper and lower case

```
1.a string is converted to upper case with toUpperCase()
2.a string is converted to lower case with toLowerCase()


<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
    <h2>JavaScrip  String Methods</h2>
    <p id="demo"></p>
    <p id="demo1"></p>
    <script>

var str1 = "Hello we are cybrom student  CYBROM ";

document.getElementById("demo").innerHTML = "<h1>"+str1.toLowerCase()+"</h1>";
//hello we are cybrom student cybrom

document.getElementById("demo1").innerHTML = "<h1>"+str1.toUpperCase()+"</h1>";
//HELLO WE ARE CYBROM STUDENT CYBROM
        </script>
```

exacting string character

```
there are 3 method
1.charAT(POSITION)
2.charCodeAt(position) return ascii scharacter
3.Property access[] :
property might be a little unpredicatable
1.it makes string look like array (but they are not)
2.if no character is found ,[] return undefined ,while charAt() return an empty String
3.it is read only. str[]="A" gives no error (but does not work)
<body>
    <h2>JavaScrip  String Methods</h2>
    <p id="demo"></p>
    <p id="demo1"></p>
    <p id="demo2"></p>

    <script>

var str1 = "Hello we are cybrom student  CYBROM ";

document.getElementById("demo").innerHTML = "<h1>"+str1.charAt(0)+"</h1>";  //h

document.getElementById("demo1").innerHTML = "<h1>"+str1.charCodeAt(0)+"</h1>";//72

document.getElementById("demo2").innerHTML = "<h1>"+str1[0]+"</h1>"; //H

 </script>
```

string split()

a string can be converted to an array with the split() method
example.
text.split(",") //split on commas
text.split(" ") //split on spaces
text.split("|") //split on pipe
if the seprator is omitted,the returned array will contain the whole string in index[0]
if the separator is "", the returned array will be an array of string characters

```
<body>
    <p id="demo"></p>
   <p id="demo1"></p>
  <script>
var str1 = "Hello, we ,are, cybrom";
document.getElementById("demo").innerHTML = "<h1>"+str1+" "+ typeof(str1) +"</h1>";
 //Hello, we ,are, cybrom string

var a=str1.split(",")
document.getElementById("demo1").innerHTML = "<h1>"+a+" "+ typeof(a)  +"</h1>";  //Hello, we,are, cybrom object
</script>
```

```
  <body>
    <p id="demo"></p>
    <p id="demo1"></p>
    <script>
      var str1 = "Hello we are cybrom";
    document.getElementById("demo").innerHTML = "<h1>" + str1 + " " + typeof str1 + "</h1>";
                                                  //Hello we are cybrom string
     var a = str1.split(" ");
document.getElementById("demo1").innerHTML ="<h1>" + a + " " + typeof a + "</h1>";
                                                //Hello, we,are, cybrom object
        console.log(str1)
        console.log(a)
    </script>
```

string indexof()

this method returns the index of (position of) the first occurance of a specified text if a string:

```
  <body>
    <p id="demo1"></p>
    <script>
      var str1 = "Hello we are cybrom ";
      var a = str1.indexOf("cybrom");
 document.getElementById("demo1").innerHTML = "<h1>" + a + " " + typeof a + "</h1>"; //13
       </script>
```

lastIndexOf()

this method returns the index of (position of) the last occurance of a specified text if a string: -->

```
  <body>
    <p id="demo1"></p>
    <script>
      var str1 = "Hello we are cybrom ANA AMARTH cybrom ";

      var a = str1.lastIndexOf("cybrom");
    document.getElementById("demo1").innerHTML = "<h1>" + a + " " + typeof a + "</h1>"; //31
    </script>
```

includes():

this method returns true if a string contain a specified value

```
  <body>
    <p id="demo1"></p>
    <p id="demo2"></p>
    <script>
      var str1 = "Hello we are cybrom ANA AMARTH cybrom ";
       var a = str1.includes("cybrom");
      var b=str1.includes("lucky")

document.getElementById("demo1").innerHTML = "<h1>" + a + " " + typeof a + "</h1>";
//true  Boolean

document.getElementById("demo2").innerHTML = "<h1>" + b + " " + typeof b + "</h1>";
//false  boolean
            </script>
```

template literals:

synonyms
template literals
template string
string template
back-tics syntax
template literals use back-tics(``) rather than the quotes ("") to define a string:
ex.le text=`hello wolrd!`;

quotes inside string:
with the template literal you can used singlt or doublr quotes inside a string

```
  <body>
    <p id="demo"></p>
    <p id="demo1"></p>
    <p id="demo2"></p>
    <script>
var str1 = `hello world`;
document.getElementById("demo").innerHTML = "<h1>" + str1 + " , " + typeof str1 + "</h1>";
/                                                                  /hello world , string

var s=`"hello world" 'hey'`;
document.getElementById("demo1").innerHTML = "<h1>" + s  + "</h1>"; // "hello world" 'hey'
    </script>
```

multiline string

template literals allows multiline strings.

```
<body>
    <p id="demo"></p>
    <script>
var str1 = `hello world
we are cybrom student
from bhopal `;
document.getElementById("demo").innerHTML = "<h1>" + str1 + "</h1>";
  //hello world we are cybrom student from bhopal

    </script>
```

interpolation

template literals provide an easy way to interpolation variable and expressions into strings
the method is called string interpolation
syntax:
${...}

```
    <script>
var name = "sachin"
var age = "sachin"
var city = "sachin"
var str=` i am ${name} , age ${age} ,city ${city}; `
document.write("<h1>",str,"</h1>" )           //i am sachin , age sachin ,city sachin;
    </script>
```

expression substitution

template literals allow expression in sting -->

```
    <script>
let radius=12.45;
var str=` area : ${(3.14*radius*radius).toFixed(2)} `;
document.write("<h1>",str,"</h1>" )         //area : 486.71;
    </script>
```

Javascript date objects:

#Example
const d = new Date():

Creating date objects.
* Date objects are created with the new Date() constructor.

there are 4 ways to create a new date object:
* new Date()
* new Date(milliseconds)
* new Date(date string)
* new Date(year, month, day, hours, minutes, seconds, milliseconds)

new Date(year, month, day, hours, minutes, seconds, milliseconds)=
new Date(year,month...) creates a new date object with a specified date and time.
7 numbers specify year,month,day,hour,minute,second,and milliseconds(in that order.)

note** Javascript counts months from 0 to 11.
january=0
december = 11
specifying a month higher than 11, will not result in an error but add the overflow to the next year.

new Date(milliseconds)

new Date(milliseconds) creates a new date object as zero time plus milliseconds.

Example :
const d = new Date(1565656565656);

Example :
const d = new Date(86400000);
one day (24 hours) is 86 400 000 milliseconds.

new Date(date string) :

Javascript date input:
new Date(date string) creates a new date object with a date string.
there are generally 3 types of Javascript date input formats.

types	Example
ISO date	"2015-03-25"(the international standard)
Short Date	"03/25/2015"
Long Date	"March 25, 2015" or "25 mar 2015"

the iso format follows a atrict standard in js.
the other formats are not so well defined.

Date Input- parsing dates

Javascript has a built-in function called Date.parse() that parses a date string and returns the number of milliseconds since January 1, 1970.

Example :
const d = Date.parse("March 25, 2015");

```
<script>
  let mydate = Date.parse("august 15, 1999");
  document.write("<h1>", mydate);  //934655400000
</script>
```

```
<body>
    first way
<script>
let mydate= new Date();
document.write("<h1>",mydate); //Thu Jun 08 2023 13:33:16 GMT+0530 (India Standard Time)
</script> -->



    second way
<script>
let mydate= new Date(1999,7,15,4,30,40,70) //max 7 min 2 arguments // 1 sec=1000 milliseconds

    document.write("<h1>",mydate); //Sun Aug 15 1999 04:30:40 GMT+0530 (India Standard Time)
</script>


    third way
     <script>
        let mydate= new Date(86400000*365);      //86400000 one day milisecond
        document.write("<h1>",mydate);
    </script>

    <!-- fourth way -->
    <script>
        let date= new Date("1999-08-15");    // ISO
        let date1= new Date("08/15/1999");
        let date2= new Date("aug 15 1999");
        let date3= new Date("15 aug 1999");
        document.write("<h1>",date);
        document.write("<h1>",date1);
        document.write("<h1>",date2);
        document.write("<h1>",date3);
            //all output:
          //  Sun Aug 15 1999 05:30:00 GMT+0530 (India Standard Time)
    </script>
```

```
<body>
  <script type="text/javascript">
    let date=new Date();
    let mytime=date.getTime();
    const week=["sun","mon","tue","wef","thurs","fri","satu"]

 document.write("<h1>",mytime,"</h1>"); //1687335483598
 document.write("<h1>",week,"</h1>"); //sun,mon,tue,wef,thurs,fri,satu
    document.write("<h1>",date,"</h1>");//Wed Jun 21 2023 13:48:39 GMT+0530 (India Standard
Time)
    let myday=date.getDay();
    document.write("<h1>",myday,"</h1>");//3
  </script>
</body>
```

```
  <script type="text/javascript">
    let date=new Date();
document.write("<h1>",date,"</h1>"); //Wed Jun 21 2023 13:49:59 GMT+0530 (India Standard
Time)
    let mymilisecond=date.getMilliseconds();
    document.write("<h1>",mymilisecond,"</h1>");//654
 </script>
```

```
  <script type="text/javascript">
    let date=new Date();
    document.write("<h1>",date,"</h1>"); //Wed Jun 21 2023 13:51:01 GMT+0530 (India Standard
Time)
    let mymonth=date.getMonth();
    const
month=["jan","feb","march","april","may","june","jul","aug","sep","oct","nov","dec"]
    document.write("<h1>",month[mymonth],"</h1>");//june
  </script>
```

spread operator (imp for react)

spread operator(...) allows us to quickly copy all or part of existing array or object into another array or object .
we can use spread operator with object to.
spread operator is used to expand or spread an iterable or an array

```
<script>
    const name1=["ram","amarth","patel","pankaj"]
    const name2=["vicky","subhau","patel","pankaj"]
    const name3=[...name1,...name2]
    console.log(name3) //["ram","amarth","patel","pankaj","vicky","subhau","patel","pankaj" ]
</script>
```

```
    <script>
    const a=[1,2,3,4,5,6]
     const [one,two,three,...b]=a
      console.log(a) // [1, 2, 3, 4, 5, 6]
      console.log(one ,typeof(one)) // 1 'number'
      console.log(two) //2
      console.log(three)//3
      console.log(b , typeof(b)) //(3) [4, 5, 6] object
    </script>
```

```
    <script>
    // distractive method
     const [one,two,three]=[1,2,3]
      console.log(one ,typeof(one)) // 1 'number'
      console.log(two)   //2
      console.log(three) //3
    </script>
```

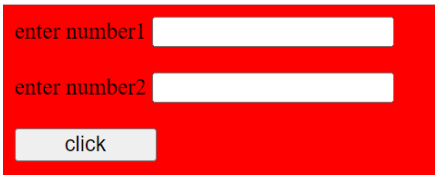
```
<script>
    const student={"rolno":120,"name":"sachin","age":20,"city":"Mumbai"}
    const fess={"student_fees":400,"fees_paid":300,"balance":100,}
    const a={...student,...fess}
    console.log(a)
// a={ "rolno":120,"name":"sachin","age":20,"city":"Mumbai",    "student_fees":400,
"fees_paid":300,"balance":100,}
</script>
```

```
<script>
    const name1=["ram","amarth","patel","pankaj"]
    console.log(...name1 , typeof(name3)) //ram amarth patel pankaj undefined
</script>
```

ADD event listerner

```
<body>
    <button id="demo" > hello world......</button>
    <script>
        document.getElementById("demo").addEventListener("click",display1)
        function display1(){
            document.getElementById("demo").innerHTML="not hello world......"
        }
    </script>
```

```
enter number1 <input type="text" id="no1"><br><br>
enter number2 <input type="text" id="no2"><br>  <br>
<button id="btn" style="width: 100px; font-size: 15px; ">click</button> <br>

  <h1 id="ans" ></h1>
<script >
document.getElementById("btn").addEventListener("click",mycal)
function mycal(){
    let no1=parseInt( document.getElementById("no1").value);
    let no2=parseInt(document.getElementById("no2").value);
    let ans= no1 + no2
document.getElementById("ans").innerHTML="ADDITION" +" "+ ans
}
    </script>
```



filterr():

syntax :
iterable.filter(function(){})
filter , is used to filter .
when we pass function as argument to another function and return value if true called callback.

```
       <script>
      const n=[22,51,55,62,40,80,51]
      const even= n.filter(function (val){ if (val%2==0){return true}})
      console.log(n)
      console.log(even)
   </script>
```

```
▼ Array(7) ℹ
    0: 22
    1: 51
    2: 55
    3: 62
    4: 40
    5: 80
    6: 51
    length: 7
  ▶ [[Prototype]]: Array(0)
▼ Array(4) ℹ
    0: 22
    1: 62
    2: 40
    3: 80
    length: 4
  ▶ [[Prototype]]: Array(0)
```

Another way

```
    <script>
      const n=[22,51,55,62,40,80,51]
      const even= n.filter(myfun)
      function myfun (val){ if (val%2==0) {return true} }
      console.log(n)
      console.log(even)
   </script>
```

Another way

```
    <script>
      const n=[22,51,55,62,40,80,51]
      const even= n.filter( (val) => val%2==0 )
    //   const even= n.filter( fun = (val) =>  {if (val%2==0) {return true} } )
      console.log(n)
      console.log(even)
   </script>
```

Reduce()

```
<script>
  const n=[1,1,1,1,1]
const r= n.reduce(function( val,init){return  val+init } , 0 )
  console.log(r) //5
</script>
```

Calculator

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #box{
            width: 220px;
            height: 320px;
            background-color: grey;
            border: 5px solid black;
            align-items: center;
        }
        #mytext{
            width: 211px;
            height: 35px;
            border: 3px solid black;
            background-color: aqua;
            margin-bottom: 10px;
        }
        input{
            width: 50px;
            height: 50px;
            font-size: larger;
        }
        #beql{
            width: 160px;
            font-size: xx-large;
        }
        </style>
    <script>
        function f1(){
            a=document.getElementById('mytext').value;
            b=a.slice(0,-1);
            document.getElementById('mytext').value=b;
        }
    </script>
```

```
</head>
<body>
    <div id="box">
        <input type="text" id="mytext">
        <br>
        <input type="button" value="1" id="b1"
onclick="document.getElementById('mytext').value+=document.getElementById('b1').value">
        <input type="button" value="2" id="b2"
onclick="document.getElementById('mytext').value+=document.getElementById('b2').value">
        <input type="button" value="3" id="b3"
onclick="document.getElementById('mytext').value+=document.getElementById('b3').value">
        <input type="button" value="+" id="b+"
onclick="document.getElementById('mytext').value+=document.getElementById('b+').value">
        <input type="button" value="4" id="b4"
onclick="document.getElementById('mytext').value+=document.getElementById('b4').value">
        <input type="button" value="5" id="b5"
onclick="document.getElementById('mytext').value+=document.getElementById('b5').value">
        <input type="button" value="6" id="b6"
onclick="document.getElementById('mytext').value+=document.getElementById('b6').value">
        <input type="button" value="-" id="b-"
onclick="document.getElementById('mytext').value+=document.getElementById('b-').value">
        <input type="button" value="7" id="b7"
onclick="document.getElementById('mytext').value+=document.getElementById('b7').value">
        <input type="button" value="8" id="b8"
onclick="document.getElementById('mytext').value+=document.getElementById('b8').value">
        <input type="button" value="9" id="b9"
onclick="document.getElementById('mytext').value+=document.getElementById('b9').value">
        <input type="button" value="" id="b"
onclick="document.getElementById('mytext').value+=document.getElementById('b*').value">
        <input type="button" value="0" id="b0"
onclick="document.getElementById('mytext').value+=document.getElementById('b0').value">
        <input type="button" value="." id="b."
onclick="document.getElementById('mytext').value+=document.getElementById('b.').value">
        <input type="button" value="C" id="bc"
onclick="document.getElementById('mytext').value='">
        <input type="button" value="/" id="b/"
onclick="document.getElementById('mytext').value+=document.getElementById('b/').value">
        <input type="button" value="BCK" id="bck" onclick="f1();">
        <input type="button" value="=" id="beql"
onclick="document.getElementById('mytext').value=eval(document.getElementById('mytext').value
)">
    </div>
</body>
</html>
```

document object model

when a web page is loaded the browser create a document object model of page .
the html dom model is constructed as a tree of object
with the object model js gets all the power it need to create dynamic HTML .

changing html element

property	description
element.innerHTML = NEW Html content	chnge the inner html of an element
element.attribute = new value	chnge the attribute value html of an element
element.style.property=new style	chnge the style of an html of an element

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<style>
    #box{
        width: 300px;
        padding: 30px;
        border-radius: 30px;
        background-color: aquamarine;
        border: 2px solid black;
        font-size: 40px;}
</style>

<body>
<div id="box" onmouseover="mydata()"  onmouseout="mydata2()">
cybrom
</div>

<script>
    function mydata(){ document.getElementById("box").innerHTML="bansal"}
    function mydata2(){ document.getElementById("box").innerHTML="cybrom"}
</script>
</body>
</html>
```

```
    enter city : <input type="text"  id="a">
    <button onclick="mycity()">click</button>
<script>
    function mycity(){
        document.getElementById("a").value="bhopal"
    }
</script>
```

```
<body>
<center>
           <img src="download.jpeg"  alt="" srcset="" id="a" style="border-radius: 250px; width: 150px; height: 150px;"> <br><br>

    <button onclick="mydata()"> click here </button>
    <button onclick="mydata2()">click here</button>
    <button onclick="mydata3()">click here </button>
</center>

    <script>
        function mydata(){ document.getElementById("a").src="download (1).jpeg"}
        function mydata2(){ document.getElementById("a").src="download.jpeg" }
        function mydata3(){ document.getElementById("a").src="download (2).jpeg" }
</script>
```

click here click here click here

```
 <div id="box">
        Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fugiat, totam voluptate eaque eius nisi nihil corporis sapiente dolorem quas necessitatibus similique tempora vel, sit porro quam error odit pariatur perspiciatis?m
    </div>
    <button onclick="display();">click here</button>
    <script>
        function display()
        {    document.getElementById("box").style.color="red";
            document.getElementById("box").style.fontFamily="arial";
            document.getElementById("box").style.backgroundColor="yellow";
            document.getElementById("box").style.border="5px solid black";
            document.getElementById("box").style.width="200px";
            document.getElementById("box").style.padding="20px";
        }
    </script>
```

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fugiat, totam voluptate eaque eius nisi nihil corporis sapiente dolorem quas necessitatibus similique tempora vel, sit porro quam error odit pariatur perspiciatis?m

click here

after click:

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fugiat, totam voluptate eaque eius nisi nihil corporis sapiente dolorem quas necessitatibus similique tempora vel, sit porro quam error odit pariatur perspiciatis?m

click here

CREATE ELEMENT HTML

```
<button onclick="display();">Click here</button>
    <script>
        function display(){
            let mybtn=document.createElement("button");
            mybtn.innerHTML="Click here";
            document.body.appendChild(mybtn);
        }
    </script>
```

Append child

```
<body>
    <button id="btn">Click Here !</button>
    <h1>My subject</h1>
    <ol id="mylist">
        <li>Python</li>
        <li>PHP</li>
        <li>JAVASCRIPT</li>
    </ol>
    <script>
    document.getElementById('btn').addEventListener("click",display);
        function display(){
            let myli=document.createElement("li");
            myli.innerHTML="JAVA";
            document.getElementById("mylist").insertBefore(myli,mylist.children[1]);
        }
        </script>
```

Click Here !

My subject

1. Python
2. PHP
3. JAVASCRIPT

Click Here !

My subject

1. Python
2. JAVA
3. JAVA
4. PHP
5. JAVASCRIPT

```
    <button id="btn">Click Here !</button>
    <div id="box"> </div>
    <script >
document.getElementById('btn').addEventListener("click",display)
function display(){
    let mypara=document.createElement("p");
    mypara.innerHTML="Enter city : <input type='text'>";
    mypara.style.width="300px";
    document.getElementById("box").appendChild(mypara);
}    </script>
```

Click Here !

Enter city :

Enter city :

INSERT BEFORE ELEMENT

```
    <button id="btn1">Click here</button>
    <div id="box1">
        <h1 id="head1">ThIS IS JAVASCRIPT CLASS</h1>
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia, libero! Ducimus
animi laboriosam vel perferendis odio laudantium pariatur! Aliquid labore quae beatae soluta
perferendis, enim nostrum quas nobis laudantium quisquam.
        <h1 id="head2">SECOND PARA</h1>
        Lorem ipsum dolor sit, amet consectetur adipisicing elit. Accusamus, qui. Iusto
dignissimos illo eos blanditiis accusantium minima nam, voluptatem quis.
    </div>
    <script >

document.getElementById('btn1').addEventListener("click",display1)
function display1(){
    let myobj=document.createElement("h1");
    myobj.innerHTML="hi this is sunil's here !!";
    let myele=document.getElementById('head2');
    document.getElementById("box1").insertBefore(myobj,myele);
}
    </script>
```

---

window setInterval() , clearinterval():

is a method calls a function at specified interval (in miniseconds)

AND continue calling a function untill clearINterval() is called or the window is closed
setinterval(function , milisecond)

window clearinterval():
method clear a timer set with setinterval() method.

```
<body>
    <script>
     function mydisplay(){ document.write("<h6>Welcome To Cybrom</h6> ") }
        setInterval(mydisplay,3000) //3000 = 3 SEC
    </script>
</body>
</html>
```

```
<button onclick="mydisplay()">click me </button>
<p id="demo">   </p>
    <script>
     function mydisplay(){
       function data(){document.getElementById("demo").innerHTML+="amarth"+" " }
       setInterval(data,1000)
    }
     </script>
```

```
<h4 id="demo"></h4>
<script>
function mydate() {
const d= new Date();
document.getElementById("demo").innerHTML = d; }
setInterval(mydate,1)
</script>
```

```
<button onclick="stop()"> stop... </button>
    <p id="a"></p>
    <script>
        function display(){
        document.getElementById("a").innerHTML+="amarth" +" " }

        var x=setInterval(display,1000)

        function stop(){ clearInterval(x)}
    </script>
```

```
<center>
    <h1 id="cnt">0</h1>
    <button onclick="start()">Start.</button>
    <button onclick="stop()">Stop.</button>
</center>

 <script>
var mycnt=0;
var mystp;
function mystartnow(){
    var data=parseInt(document.getElementById("cnt").innerHTML)
    mycnt=data+1;
    document.getElementById("cnt").innerHTML=mycnt; }

function start(){
 mystp=setInterval(mystartnow,1000)}

function stop(){
    clearInterval(mystp);
}  </script>
```

3

Start. Stop.

window setTimeOut()

method calls a function after a number of milisecond
1 secound = 1000 milisecond
notes
the setTimeOut is executed only once.
if you need repeated execution ,use setInterval() instead.
syntax
setTimeOut(function , milisecond , param1 , param2 , ...)
clearTimeOut(mytimeout) to prevent function from running.

```
<script>
    function display(){
document.write("<h1> cybrom </h1>")
    }
    setTimeout(display,5000)
</script>
```

```
<button onclick="mystop()"> stop </button>
<script>
var myvar;

    function display(){
document.write("<h1> cybrom </h1>")
    }
  myvar=  setTimeout(display,5000)

  function mystop(){
    clearTimeout(myvar)
  }
</script>
```

Modules

Javascript modules allow you to break your code into separate files.
this makes it easier to maintain the code-base
js modules rely on the import and export statements.

There are two types of exports :
*default exports
*named exports

#named exports :-
with named export we can use export keyword with {}.
using named export we can export multiple items

Import :--

There are two types of imports :
1. Named import
2. Default import

1. with named import we can use destructive{} method with items

like :
import{rollno, name} from "/.filename.js";

2. with default import we dont use {} destructive method we direct import the item with or without name because only one item will be export.

like:
import item from "/.filename.js";

File name amarth.js

```
var name="amarth";
var age=23;
export{name,age};
const myname=()=>{
    return " THIS IS amarth !!"
}
export default myname;
```

Name export

```
    <script type="module">
        import {name, age} from "./amarth.js";
        document.getElementById("demo").innerHTML="my NAme is  "+name+"   my age is  "+age;
    </script>
```

defalut

```
    <script type="module">
        import mmm from "./sunil.js";
        document.getElementById("demo").innerHTML="my NAme is  "+mmm();
output: THIS IS amarth !!"
    </script>
```

Both name and default

```
    <script type="module">
        import mm from "./sunil.js";
        import {name,age} from "./sunil.js";
        document.getElementById("demo").innerHTML=mm();  // THIS IS amarth !!"
        document.getElementById("demo1").innerHTML="my NAme is  "+name+"  age :"+age;
    </script>
```

BOM

Window location

The window.location object can be used to get the current page address(url) and to redirect the browser to a new page.

Window location
The window.location object can be written without the window prefix.

Some examples:

*window.location.href returns the href(url) of the current page.
*window.location.hostname returns the domain name of the web host
window.location.pathname returns the path amd filename of the current page.
window.location.protocal returns the web protocal used (http : or https:)
window.location.assign() loads a new document

```
<!DOCTYPE html>
<html>
<head></head>
<body bgcolor="yellow">

    <a href="history_object.html">Home</a>|
    <a href="about1.html">About</a>|
    <a href="service.html">Service</a>|
    <a href="join.html">Join</a>|
    <a href="contact.html">Contact</a>|

    <script>
        var add=window.location.href;
        document.write("<h1>",add);    // http://127.0.0.1:5500/BOM/BOM/window-lo-1.html

       var host=window.location.hostname;
    document.write("<h1>",host);                    // 127.0.0.1
```

```
        var path=window.location.pathname;
        document.write("<h1>",path);                    // /BOM/BOM/window-lo-1.html


        var protocol=window.location.protocol;
        document.write("<h1>",protocol);                //HTTP

        function newload(){
            window.location.assign("history-obj-1.html");  //
        }
    </script>
    <button onclick="newload();">Click here</button>
</body>
</html>
```

History.back() and history.forward()

```
<button onclick="history.forward()"> Forward now</button>
<button onclick="history.back()"> Forward now</button>
```

---

cookies and local storage :--

jAvscript cookies :

A cookie is an amount of information that persists bwn a server-side and a client-side. a browser stores this information at the time of browsing.

a cookie contains the info as a string generally in the form of a name-value pair seperated by semi-colons. it maintains the state of users info among all the web pages.

how to create a cookie in js :-

in js we can create,read,update and delete a cookie by using **document.cookie property.**

The following syntax is used to create a cookie :

document.cookie="name=value";

```
<body>
    <script>
        function setcookie(){
            document.cookie="name=sunil";
        }
        function getcookie(){
            if (document.cookie!=""){
                alert("my cookie value : "+document.cookie);
            }
            else{
                alert("cookie does not set");
            }
        }
    </script>
    <button onclick="setcookie();">Create my cookie</button>
    <button onclick="getcookie();">see cookie value</button>
</body>
```

```
<body>
    <script>
        function setcookie(){
            document.cookie="name=sunil";
            alert("cookie created");
        }
        function getcookie(){
            if (document.cookie!=""){
                var myval=document.cookie.split("=");
                console.log(myval);
                alert("my name is : "+myval[1]);
            }
            else{
                alert("cookie does not set");
            }
        }
    </script>
    <button onclick="setcookie();">Create my cookie</button>
    <button onclick="getcookie();">see cookie value</button>
</body>
```

Window Localstorage :--
example
set and retrieve a localstorage name/value pair:

localStorage.setitem("lastname","thakur");
localStorage.getitem("lastname");

definition and usage :
the local storage object allows you to save key/value pairs in the browser.
Note : the localStorage object stores data with no expiration date.
the data is not deleted when the browser is closed and are available for future sessions

window.localStorage
or just
localStorage

Save data to local storage :-- localStorage.setitem(key,value);

read data from local storage:- let lastname = localStorage.getItem(key);

Remove data from localstorage :- localStorage.removeItem(key);

Remove all data from localstorage :- localStorage.clear();

File name - local-storage-1.html

```
<head> <script>
        function userset(){
            var myval=document.getElementById('usr').value;
            localStorage.setItem("uname", myval);
            var username=localStorage.getItem("uname");
  document.getElementById("demo").innerHTML="welcome "+username+" <a href='logout.html'>
Logout</a> "; }
    </script> </head>
<body >
     <a href="local-storage-1.html">Home</a>|
    <h1 id="demo"></h1>
    Enter Id : <input type="text" id="usr">
    <button onclick="userset();">click</button> </body>
```

File name logout.hmtl

```
<body>
    <script>
        localStorage.clear();
        window.location.assign("local-storage-1.html")
    </script></body>
```