JS POINT READING TUTORIAL ==>

==
=

Main Js Docs

JavaScript : javascript is a very powerful client side scripting language. It is used for animation, validation, event handling etc. It is mainly used for enhancing the interaction of users with the web page. We can make a more interactive webpage with the help of javascript.
case sensitive

Type of JavaScript
Internal

```
 <script>
  Code Here
 </script>
```

External : File Create (filename.js) and write code in this filename

```
   attach : <script src="filename.js"></script>
```

Output :

```
 document.write("Hello")  web page
 alert("Hello");    alert box
 console.log("Hello")  console panel
```

Comments

```
 // single line comments


 /*
  multiple line comments
 */
```

New Line

```
 <br> webpage
 \n  alert
```

concatinate +

Variables

```
 var variable name = value;
  varibale name = value;
```

Rules for declare varibale name

```
   * start from a-zA-Z_$
   * can combination a-zA-Z0-9
```

e.g.

```
  var a = 5;
   a = 5;

   document.write(a);
```

Data Types : allocate automatically at run time

number,string,array,object,boolean,null (by default undefined)

Check Data Type

```
  typeof()
```

e.g

```
  a = 56;
  document.write(typeof(a));
```

```
Operator
Conditional Statements
 if
 if else
 else if
 switch

 same as php


Loops
 for
 while
 do while

 same as php




Array
 array name = Array(value,value,value);
  or
 array name = [value,value,value];


 e.g.

 arr = Array(22,"blue",88.7);

 arr = [22,"blue",88.7];



count array elements
 document.write(arrname.length);
 e.g.
 document.write(arr.length);
```

User Input (Popup Box)

```
 Prompt("title","placeholder")
 confirm("message");
```

Function : It is set of reusable code

Type of Function

1 Predefine function

 library function

 inbuild function

2 user define function

```
 function abc()
 {
  alert('Hello');
 }

 function xyz()
 {
  alert('Welcome');
 }



 abc();
 xyz();
 abc();
```

Events

onmouseover

onmouseout

```
onclick
ondblclick
onkeyup
onkeydown
onblur
onfoucs
onload
```

```
setInterval() // run continuosly
setTimeout() // run one time
clearInterval() // stop
```

Dom (Document Object Model)

Element Attribute Style

```
document.getElementById()._____________
document.getElementsByName().__________
document.getElementsByTagName().________
document.getElementsByclassName().________
document.querySelector().__________
document.querySelectorAll().__________
```

==

How to write value :

```
document.write("hello");   // print hello in the browser page
alert("hello");          // print hello in alert box open page
console.log('hello');    // display hello in console
// how to open console :
```

open browser + right click on mouse + goto to inspect + after goto console link open (and show and statement excute using console.log();)

```
console.log();      // print and statement using console command
console.error("this is error");   //  print you create error
```

then use it (devloper most uses this when created code)

```
console.table();   // show value on console table formate
console.clear()     // clear your console using this one
================================================================
====================================
1) conditions :
if
else
else if
2) Ternany opreator   ( like work if else condition statement
single line condition .)
3) switch case / default case :
4) variables :
var
Let
Count
5) Continue And Break Statement
6) Mathmetical oparator
* , + , - , % , / , =
7) Logical oparator
&& (and both condition true then work)
|| (only one condition true then work)
! (if condition true then return false , else condition false
then return true)
8) Conditional oparator
=== (using to value and data type check then use it)
== (using to value check then use it)
!= (using to value is not then check it)

9) alert (open popup using js)
10) confirm (using true or false condition using this one ==>
ok == true ; and cancel == false)
uisng only condition based confirm
11) prompt box (using the user input then use it .)
12) function        // create function and call many time (its
use only code reusevisiblity is low then use)
13) function with parameter   // function with parameter (use
some change inside code)
14) function override         // function override varriable is
inside function parameter
```

15) function overload // function value pass 2 var , and calculate with three then use it.

16) function with return value // function return value after print then store is first any var then print.

17) Global and Local variable in function.

Global varriable (access any where inside function or outside function)

Local varriable (access only inside function)

// Both is define the (fisrt come to local varriable after come to global) .

18) Javascript Events :

Events :

click (onclick) // when button click then call any function user create onclcik button.

Double click (ondblclick) // when user click button double time then call function

Right click (oncontextmenu) // when right on text menu

On change (onchange) // only work with dropdown value changes.

Mouse Hover (onmouseenter) // on mouse

Mouse Out (onmouseout) // out mouse

Mouse Down (onmousedown) // down mouse

Mouse Up (onmouseup) // up mouse

Key Press (onkeypress) // when key press then js call function ;

keypress only (working with body and form only not working with any html attributes.)

Key Up (onkeyup) // onkeyup (after keypress then left key then working this open function .

Load (onload) // when window load then working function

Unload (onunload) // when window close then working function

Resize (onresize) // when resize window (zoom in Zoom out then working) / function

Scroll (onscroll) // when page scroll the window open js file data and excute.

Loop in Javascript :

Javascript loops :
1) While Loop
2) do / While Loop
3) for Loop
4) for/in Loop (with use objects)
5) forEach Loop (with use Array)

While Loop :
synatx :
Ex :

```
var i = 0;
while (i<=10)
{
 document.write(i);
 i = i +1;
}
```

Do While Loop :
synatx :
Ex :

```
var i = 1;
do{
 document.write(i);
 //i = i+ 1;
 i++;
}while(i<=10)
```

For Loop :

```
for(var a = 1; a <=10 ; a++)
{
 document.write(a);
}
```

Continue and Break statement :
Continue Statement :

```
for(var a = 1; a <=10 ; a++)
{
 if(a == 3){
  continue;    // skip a = 3 another print all no (o/t = >
1,2,4,5,6,7,8,9,10)
 }
 document.write(a);
}
```

Break Statement :

```
for(var a = 1; a <=10 ; a++)
{
 if(a == 3){
  break;       // break  a = 3 another print before 3 no after
all skip no (o/t = >   1,2)
 }
 document.write(a);
}
```

Nested Loop : (loop inside loop is called nested loop)
Even and odd no using Js

===

DataType of Javascript :
String (combination of charcters is called Strings)
Number (Float and Integer both are called Number in JS)
Boolean (Its return Only True or False 1 or 0)
Array   (Combination of all datatype in arary)
Object (Any things is object its using in Advance Js)
Null (No one value is here is called Null)
Undefined (create varriable but not define any value is there is called Undefined datatype).

==

Array In Javascript :
Array : Array in Javascript the combination of all datatype is called list (array) .

```
     0 1 2 <---     index
  Ex :  var arr = [10,20,30];
       |
       values of array
```

```
Ex :  var arr = [10,20,30];
   document.write(arr);
   document.write(arr[2]);
   o/t =>   10,20,30
   o/t => 30
   // note : if user input index is not array lenght and fill
more than then ( value return is Undefined)
Ex :  var arr = new array(10,20,30);
```

Multidemsional array :

json vlaue called also :

```
var a = [
 [10,20,30],
 [20,30,40],
];
document.write(a[0][0]);
document.write(a[1][2]);
o/t =>  10
o/t =>  40
```

Modify And Delete list value of array in Js :

Modify :

```
var arr = [10,20,30];
arr[0] = 100;
document.write(arr);
o/t =>  100 ,20,30
```

Delete :

```
var arr = [10,20,30];
delete arr[0];
document.write(arr);
o/t =>  20 ,30
```

Array method in javascript :

```
 sort()     slice()      find()
 reverse()    splice()     findIndex()
```

pop()
push()
shift()
unshift()
concat()
join()
isArray()
indexOf()
lastIndexOf()
entries()
every()
filter()
inculdes()
some()
forEach()
toString()
valueOf()
fill()

---

sort() => sorting array with assending to descending order .
reverse() => array reverse (last index to first index) .
pop() => remove at the end value.
push() => add the end index value in given array.
shift() => shift delete value on first index of array.
unshift() => unshift append value on First index of array.
concat() => merge two array using this.

```
Ex ->  var arr1 = [10,20];
  var b = arr1.concat(30,40);
  document.write(b);
  o/t ->  10,20,30,40
Ex ->  var arr1 = [10,20];
  var arr2 = [30,40];
  var c = arr1.concat(arr2);
  document.write(c);
  o/t ->  10,20,30,40
```

Join() : join is use the array value with which oprator (like |,& , space)

```
Ex : var arr1 = [10,20];
   var arr2 = [30,40];
   var c = arr1.concat(arr2);
   var d = c.join("&");
   document.write(d);
   o/t :  10&20&30&40
```

slice() : slice is used array value index to end index user want then use it .

```
  var arr1 = [10,20,30,40];
  b = a.slice(1,3);
  document.write(b);
  o/t : 20,30
```

splice() : value post after which index then use it:

syntax : arr.splice(indexvalue , how many delete no , "added values ")..

```
 Ex : var a = [10,20,30];
      a.splice(2,0,90,100)
      document.write(a);
      o/t =>  10,20,90,100,30
```

point : splice using data insert which index and after how many value you want delete ,after pass value you insert indexing value then use it.

how many no delete using value then (2,2,"hello").

then after 2 index value delete in array 2 value and insert hello values in array.

```
//  var a = [10,20,30,40];
 a.splice(2,2)
  document.write(a);
  // o/t ->  10,20
```

splice using start with 2 index after remove 2 values in array then use it.

----

isArray() : It is Check array is or not then use it .

```
Ex : var a = [10,20,30];
  var b = array.isArray(a);
  document.write(a);
  o/t =>  true;
```

indexOf() : You can check this index is an array or not then use it .

syntax : indexOf('search item' , 'Starting index value you want to search value after this index.');

```
Ex :  var a = [10,20,30,10];
   var b = a.indexOf(10);
   var b = a.indexOf(10,1);
   document.write(b);
   // o/t : 0
   // o/t : 3
```

lastIndexOf() : searching with the last indexing value if you want to serach it .

```
Ex :  var a = [10,20,30,10];
   var b = a.lastIndexOf(10);
   document.write(b);
   // o/t =>  3
```

By default lastIndexOf("value") :   its return last index means always (-1) if you want to search but the no is not an array then return -1.

includes() : includes seraching value is array is not then use includes function in js. (ites return true or false.)

syntax : includes ('serach item value');

```
Ex :  var a = [10,20,30,10];
   var b = a.includes(10);
   document.write(b);
   // o/t =>  true.
```

some() : Using this you can test condition of using elements any one is satisfied valie of array using some () function its return true .

```
Ex :  var a = [10,20,30,10];
   var b = a.some(checkage);
   document.write(b);
   function checkage(a){
    return a >=20;
   }

   0/t : true .
```

every() : when check all condition is satisfied then return true otherwise return false.

```
Ex :  var a = [10,20,30,10];
   var b = a.every(checkage);
   document.write(b);
   function checkage(a){
    return a >5;
   }
```

0/t : true.

find() : find is used only check condition to in array element and return first true value.

Ex : var a = [10,20,30,10];
  var age = 20;
  a >= age =====> 30
  find(function name)

findIndex() : findIndex is used only check condition to in array element and return first true value but print only index .

Ex : var a = [10,20,30,10];
  var age = 20;
  a >= age =====> 1 (index value return value is 20 , index is 1.)
  findIndex(function name)

filter() : filter is use with the condition check and getting true which element pass condition then store with another array .

Ex : var a = [10,20,30,10];
  var age = 20;
  condition pass in function name : condition (a >= age)
====> [20,30,40] its return all value pass condition store in array.
  filter(function name)

toString() : Convert Array with String .
syntax : arrayname.toString();

valueOf() : print all value of array then use this function .
syntax : array.valueOf(); // output : complete return value of array.

fill() : fill value of same in all indexes value then use it .
syntax : array.fill('ram');

----

forEach loop () :

```
var a = [10.20,30,40];
a.forEach(function(value , index)){
 document.write(index + " : " + value + "<br>");
});
```

o/t ==>

```
0 10
1 20
2 30
3 40
```


What is Object In Javasrcipt : It is like store any value like as datatype anyone store with in array . its denoted all value in {} curly bracesses . its like dict . Ex :

```
var a = {
 firstName:"Ram",        // index of associated array key and
value store of any object.
 lastName:"Shukla",
 age:18,
 favMovie:['shole','hum','takdeervala'],
 living : {
  city:"Kanpur",
  country:"India",
 };
 salary : function(){
 return 25000;
 },
 fullname : function(){
 return this.firstName + " " + this.lastName;
 },
};
console.log(a);   // print all object property in Console.
document.write(a)   // object Object
document.write(a.firstName)   // Ram
document.write(a.favMovie[1])   // takdeervala
document.write(a.salary())    // 25000 (call function inside
object.)
document.write(a.fullname())  // Ram Shukla  (call to function
inside property call through the object.)
document.write(a.living.city)      // Kanpur . (call to object
inside object property to access that.)
```

If you want to print any object property access then use it . Ex
: document.write(a.firstname)
return the property value using any value through the obejct
data .

sample : Object is called store property to any value like
array , string , dict anyone here.
     object created Multiple Function also be created .

****

Second Method : Object (Then I create a object and add some
property with object any time and any where Use it.)

```
var a = new Object();    // new is keyword use to create any
object then use it .
a.firstName = "Ram";     // add to assign property of obejct
create and Add also be there .
a.lastName = "Kumar";
document.write(a.firstName);  //  Ram
```

****

Array Of Object :
Any array inside call to object .

```
var student = [
  {Name:"Ram",Age:15},
  {Name:"shyam",Age:45},
  {Name:"sumit",Age:25},
 ];
 for(var a=0;a<student.lenght;a++){
  document.write(student[a].name + " " + student[a].age +
"<br>");
```

```
 }

 0/t ==>   Ram 15
     shyam 45
     sumit 25

********************************************************
Const a = ['ram','shyam','mohan'];
a = ['shyam','raj','vijay'];
document.write(a)        // cannot define const a so cannot
change it fast.
// I can change only const value then using index value not
change any value same value with same varriable ...
a[1] = "Hello";
document.write(a)        //  ram,shyam,mohan

const a = {
 name:"Ram",
 age:25,
};
a.name ="Yahoo";
document.write(a.name);
o/t =>  Yahoo.

*************************************************************
```

For/In loop in Javascript :

```
var obj = {

 firstName:"Ram",
 lastName:"kumar",
 age:25,
};
for(var key in obj){
 document.write(key +" : "+obj[key] + "<br>");
}
o/t ==>  firstName : Ram
   lastName : kumar
   age : 25
```

****

Array - Map() function : Map is the function relation relate with the calculate value in second array .

syntax :

```
a.map(function(){
 Statement;
});
```

Ex :

```
var a = [1,3,5,7,10];
var b = a.map(test);
document.write(b);
function test(x){
 return x*10;
}
```

```
0/t ==>  10,30,50,70,100 .
```

Second Condition // how to map array of object in Js :

```
var arr = [
 {fname:"yahoo",lname:"baba"},
 {fname:"shyam",lname:"kumar"},
 {fname:"saurabh",lname:"shukla"},
];
var b = arr.map(test);
document.write(b);
function test(x){
 //return x.fname;
 return x.fname +" "+x.lname;
}
o/t =>   yahoo,shyam,saurabh
0/t =>   yahoo baba , shyam kumar , saurabh shukla .
```

****

Javascript string Method :

length ==> property . indexOf() Concat()

toLowerCase() lastIndex() split()

toUpperCase() replace() repeat()

includes() trim() slice()

startWith() charAt() substr()

endsWith()

search()

match()

charCodeAt()

fromCharCode()

valueOf()

substring()

toString()

How impleiments Methods and property of string in Javascript.

var str = "Hello It is javascript";

var a = str.length // find the length of string how many charcters Ans : (22)

document.write(a);

var a = str.toLowerCase(); // it is convert string to Lowercase in js.

var a = str.toUpperCase(); // it is convert string to Uppercase in js.

var a = str.includes('is'); // it search substring it is have the string then return True/false.

Note : include is case senstive string its return true or false.

var a = str.startsWith('Hello'); // Its search with the first string word or charcter then return true/false.

var a = str.endsWith('javascript'); // Its search with the end string word or charcter then return true/false.

var a = str.search('javascript'); // Its return index or postion of given chacter if its have then retun index.

// result is : 13 (index of charcter return value.)

match return the string or chacter using the collection in given array .

var a = "Hello It is javascript is ?";

var a = str.match(/is/g); // its using only regular expression value in given string.

documents.write(a);

o/t ==> is,is

indexOf() => its return the index value of searching charcters. is starting index value .

lastIndexOf() => its return the index value of searching charcters is last postion after return position.

replace() ==> its replace charcter to charcter wise you can change the value of replace position.

trim() ==> its use to remove all spaces left or right both side you can use it.

charAt() ==> when you are use the charAt with postion it return the postion of charcter. if you find the postion which one is not is string then not return any value .

$ ex : var a = str.charAt(3);   // output : l

charCodeAt() ==> charCodeAt() using user use postion of charcter then return Ascci Value of that .

Ex : var a = str.charCodeAt(1);   // if return a then return value is (97) .

fromCharCode() ==> using fromCharCode is using the return ascii code no return any charcter .

Ex : var a = String.fromCharCode(65);   // output : A

concat() ==>   merge two or more than string using it .

Ex :

```
str1 = "hello";
str2 = "text";
str3 = "its good";
var a = str.concat(str2);
var b = str.concat(str2,str3);
document.write(a);
document.write(b);
o/t ==>   hello text
o/t ==>   hello text its good
```

split() ==> split using string is break which one charcters then use it .

Ex : var a = str.split(" ");   // break string with space .

repeat() ==> repeat string with the which times .

Ex : var a = str.repeat(5);   // it is print 5 times the same string value .

slice() ==> slice() function is used with the which with startwith the endswith print the string value inside its.

Ex : var a = str.slice(10,17);   // its start with the value chacter 10 to 17 between print all chacters return.

substr() ==> its print string with inside substr .

Ex : var a = str.substr(2,5);   // print substring between (2 to 5 charcter inside it.)

toString() ==> Its is Convert int to string value.
valueOf() ==> Print string value using this function.

****
**

Number methods In javascript :
Number()
parseInt()
parseFloat()
isFinite()
isInteger()
toFixed(x)
toPrecision(x)

Number() : Number return any string convert with number this function .

```
var a = "99";
var num = Number(a);
document.write(num+100);      // convert string to int using
value and revert value in integer.
```

parseInt() : Convert String with Numeric Value then use it. (its return first value of number.) // also retrun any value is return integer value .

```
Ex : var a = "10.33";
  var num = parseInt(a);
  document.write(num);   //  10.
```

parseFloat() : Convert Number with decimal to decimal format value then use it.

```
Ex :  var a = "10.33";
   var num = parseFloat(a);
   document.write(num);       //  10.33
```

isFinite() : isFinite() is using then return True or false (its check number is finite or Infinite.)

isInteger() : isInteger() is using then retun True or false (its check Integer is or not.)

toFixed() : toFixed() is function then return how much decimal is fixed after decimal how much digit fixed.

```
Ex : var a = 1000.087;
```

```
  var b = a.toFixed(2);
  document.write(b);     //  1000.08
```

toPrecision(2) : this function is used then return two digit before decimal and after both count digit no is return.

```
Ex : var a = 1000.007;
  var b = a.toPrecision(2);
  document.write(b);     //  1000.0
```

****

JavaScript Math Methods :

```
ceil()       sqrt(x)
floor()       cbrt(x)
round()       pow(x,y)
trunc(x)      random()
max(x,y,z,....,n)    abs(x)
min(x,y,z,....,n)    PI
```

Its is using to create website slider , animation etc then use it , basically is not use in browser site.

ceil() / floor() ==>   ceil and floor use only float value but print integer value then use it.

ceil() : ceil float value is return with nearest bigger value .

```
Ex : var a = math.ceil(5.2);
  document.write(a);     //   o/t ==>   6
  var a = math.ceil(-1.7);
  document.write(a);     //   o/t ==>   -1
```

floor () : floor float value is return with nearest smallest value .

```
Ex : var a = math.floor(5.2);
  document.write(a);     //   o/t ==>   5
  var a = math.floor(-1.7);
  document.write(a);     //   o/t ==>   -2
```

round() : round return the round value of given float value .

```
Ex : var a = math.round(5.2);
  document.write(a);     //   o/t ==>   5
  var a = math.round(-1.7);
  document.write(a);     //   o/t ==>   -2
```

trunc() : trunc return float value is remove any value after decimal digits .

```
Ex : var a = math.trunc(5.233);
  document.write(a);     //   o/t ==>   5
  var a = math.trunc(-1.711);
  document.write(a);     //   o/t ==>   -1
```

max() : it return max value of given numbers.

```
Ex : var a = math.max(5,23,3);
  document.write(a);     //   o/t ==>   23
```

min() : it return min value of given numbers.

```
Ex : var a = math.min(5,23,3);
  document.write(a);     //   o/t ==>   3
```

sqrt() : its given no return with the square root of given number .

```
Ex : var a = math.sqrt(4);
  document.write(a);     //   o/t ==>   2
```

cbrt() : its given no return with the cube root of given number .

```
Ex : var a = math.cbrt(27);
  document.write(a);     //   o/t ==>   3
```

pow() : its return to pass value after calculate values of power .

```
Ex : var a = math.pow(2,3);
  document.write(a);     //   o/t ==>   8
```

random() : Its retrun Random Number of given user values.

```
Ex : var a = math.random();
  document.write(a);     //   o/t ==>   0.100929938
Ex : var a = math.floor(math.random()*10) + 1;
```

```
  document.write(a);     //   o/t ==>   8  ( it return between 1 to 10).
```

abs() : return Abslute value given any number than.

```
Ex : var a = math.abs(5.26);
  document.write(a);     //   o/t ==>   5.26
Ex : var a = math.abs(-5.26);
  document.write(a);     //   o/t ==>   5.26
```

its return any modulus value not any effect with (- or + sign.)

PI() : Its return PI value of using this.

```
Ex : var a = math.PI();
  document.write(a);     //   o/t ==>   3.142576490142576490
```

****

Date Methods Of Given Number in Javascript :

Date method call then create a object of date then use it :

```
var now = new Date();
```

Date Methods :

```
toDateString()           getMiliseconds()
getDate()    setDate()
getFullYear()   setFullYear()
getMonth()    setHours()
getDay()    setMiliseconds()
getHours()    setMinutes()
getMinutes()   setMonth()
getSeconds()   setSeconds()
```

```
Ex :
 var now = new Date();
 document.write(now);         // its return Date and time with current location.
 document.write(now.toDateString());   // its return 6/21/2020 : 1:20:00  ( only return date with time)
 document.write(now.getDate);      // Its return Only date (like 21)
 document.write(now.getFullYear());     // Its return Year ( like 2020)
 document.write(now.getMonth);     // Its return month (like 06)
 document.write(now.getDay());     // Return Day name ( like sunday)
 document.write(now.getHours());   // return Hours (like 13)
 document.write(now.getMinutes()); // return Minutes
 document.write(now.getSeconds()); // return seconds
 document.write(now.getMiliseconds);    // return Miliseconds

 var now = new Date();
 now.setDate(21/06/2020)      // set date than use it.
 document.write(now.setDate());         // set date than use it.
 document.write(now.setFullYear());     // set Year than use it.
 document.write(now.setHours());        // set Hours than use it.
 document.write(now.setMiliseconds());   // set Miliseconds than use it.
 document.write(now.setMinutes());      // set Minutes than use it.
 document.write(now.setMonth());        // set Month than use it.
 document.write(now.setSeconds());      // set Seconds than use it.

*****************************************************************
DOM (Document Object Module) ....
DOM =>  It is a Html document (head and body tag) . It is Html Tree tag.
How To Target DOM Object :
1) id          ==>   document.getElementById(id)
2) class  Name ==>   document.getElementsByClassName(name)
3) Tag name    ==>   document.getElementsByTagName(name)
**************************************
Other DOM Targeting  methods :
document         document.links
document.all                  document.forms
document.documentElement      document.doctype
```

document.head
document.title
document.body
document.images
document.anchors

document.URL
document.baseURI
document.domain

****

We can get with DOM :
HTML
Text
Attribute
DOM Get Methods :
innerText
innerHTML
getAttribute
getAttributeNode
Attribute

innerText ==> Its return text in html docs file which created by users.
syntax : var a = document.getElementById(id).innerText;
its return only text of html file inside id tag which one user can featch it.

innerHTML ==> Its return Html docs file inside with html tag return values.
syntax : var a = document.getElementById(id).innerHTML;
Its return Html code with tag inside user featch data using Id.

getAttribute ==> Its return Any attributes value using this one.
syntax :
html code : <div id="header" class="hello-class" style="border:1px solid red;"><h2> Hello Text </h2></div>
var a = document.getElementById("header").getAttribute("id");
O/t => header
var a = document.getElementById("header").getAttribute("class");
o/t => hello-class
var a = document.getElementById("header").getAttribute(style);

0/t => border:1px solid red;

getAttributeNode ==> Its retrun attributes with value of inside attributes.

syntax :

```
html code : <div id="header" class="hello-class" style="border:1px solid red;" onClick="abc();"><h2> Hello Text </h2></div>
var a = document.getElementById("header").getAttributeNode("onClick");
0/t => onClick="abc();"       // its return value with attributes name.
```

Attributes ==>   Its return Object all attributes Values :

syntax :

```
html code : <div id="header" class="hello-class" style="border:1px solid red;" onClick="abc();"><h2> Hello Text </h2></div>
var a = document.getElementById("header").Attributes;
o/t ==>  retrun all attributes in given array of objects .
NameNodeMap
{0:id,1:class,2:style,3:onClick,id:id,class:class,style:style,onClick:onclick,length:4};
```

==> return index array and Associative array of object (o,1,2 index and keys also retrun with value using attributes.)

```
var a = document.getElementById("header").Attributes[1];    // which one You are print any attributes value in js.
o/t =>  class:hello-class;
```

.value() ==> return attributes value.

.name() ==> return atributes name.

Ex :

syntax :

```
html code : <div id="header" class="hello-class" style="border:1px solid red;" onClick="abc();"><h2> Hello Text </h2></div>
var a = document.getElementById("header").Attributes[1];
var a = document.getElementById("header").Attributes[1];
document.write(a);
document.write(a);
```

```
o/t ==>
hello-class
class
****************************************************************

DOM Set Methods :
innerText
innerHTML
setAttribute
Attribute
removeAttribute
SetAttributes :
var a = document.getElementById("header").innerText = "hello";
var b = document.getElementById("header").innerHTML =
"<h1>Wow</h1>";
var c = document.getElementById("header").innerHTML = "Wow";
var d =
document.getElementById("header").setAttributes("style","10px
dotted yellow;");
var e =
document.getElementById("header").Attributes[1].value="abc";
var f =
document.getElementById("header").removeAttribute("style");
document.write(a);
document.write(b);
document.write(c);
document.write(d);  // Its set Attributes value xyz in style.
document.write(e);  // set any attributes value using this.
document.write(f);  // remove any attributtues than use
removeattribute();
0/t ==>   hello.
     Wow
     Wow
     style="10px dotted yellow;"
     class="abc";
     remove style attribute.

**********************************************
```

querySelector ==> document.querySelector(css selector) ==> its return collect all value on page using css selector but return first value of given collection using css selector.

querySelectorAll ==> document.querySelectorAll(css selector) ==> its return collect all value on page using css selector but return All values of given collection using css selector user print using indexes values .

syntax :

```
document.querySelector("#header").innerHTML="<h1>Header
Text</h1>";
var a = document.querySelector("#header").getAttribute("class");
var b = document.querySelector(".list");
document.write(a);
document.write(b);
```

o/t => abc;

  return list of in array index of object. (but retrun output only first value of collection array.)

```
var b = document.querySelectorAll(".list");
var c = document.querySelectorAll(".list")[1].innerHTML;
var d = document.querySelectorAll("ul")[1].innerHTML;       //
get attributes value using in page all ul tags.
var e = document.querySelectorAll("#header h1");
var f = document.querySelectorAll("#header h1")[1].innerHTML;
document.write(b);
document.write(c);
document.write(d);
document.write(e);
document.write(f);
```

o/t ==> return list of in array index of object. (and retrun all values of output value of collection array and print using index values.)

  return list value first index inside all Html retrun here ??

  return all ul tag of html in your page return all values.

  return all header Id inside all print h1 tag value values.

  retrun all header Id inside h1 tag this on use can use so return all value of Ist index value of h1 tag.

****

Javascript DOM CSS Styling Methods :

style
className
classList

```
# var f = document.querySelectorAll("#header").style.color;
```

its return any get value with style in css and getting to all attributes find it that.

```
# var f = document.querySelectorAll("#header").style.backgoundColor="blue";
```

its return any attributes style background color than return style.

```
# var f = document.querySelectorAll("#header").className="abc";
```

its return set after class value.

```
# var f = document.querySelectorAll("#header").classList;
```

its return all class in list show value under list.

```
# var f = document.querySelectorAll("#header").classList.add("xyz","efg");
```

its return three class (abc xyz efg)

```
# var f = document.querySelectorAll("#header").classList.remove("efg");
```

its return class values (abc,xyz).

****

JavaScript Basic Events :

```
Click(onclick)                // button click
Double Click(ondblclick)      // on double click
Right Click (oncontextmenu)   // right menu text
Mouse Hover (onmouseenter)    // hover on text
Mouse Out (onmouseout)        // remove mouse
Mouse Down (onmousedown)      // remove mouse
Mouse Up (onmouseup)          // mouse up
key press (onkeypress)        // any keypress
key up (onkeyup)              // any key press after left key
Load (onload)                 // when page load
Unload (onunload)             // when window tab is close
Resize (onresize)             // window size resize
Scroll (onscroll)             // on scrolling window up to down
```

How to use another method :

# how to use It :

```
document.getElementById("header").onclick=abc;
function abc(){

 document.getElementById("header").style.background="green";
}
```

then onclick he is change color to green.

# DOM add EventListener() Method :

```
document.getElementById(id).addEventListener("click",functionName);
    OR
document.getElementById(id).addEventListener("click",function(){

 this.style.border = "10px solid red";
});
```

UseCapture :

```
addEventListener(event,function,useCapture);
```

Ex 1 :

```
document.querySelector('#inner').addEventListener('click',function(){
 alert('inner body');
},true);
document.querySelector('#inner').addEventListener('click',function(){
 alert('inner body');
},true);
// output ==>  first working outer div after inner div event call.
```

Ex 2:

```
document.querySelector('#inner').addEventListener('click',function(){
 alert('inner body');
},false);
document.querySelector('#inner').addEventListener('click',function(){
 alert('inner body');
},false);
// output ==>  first working inner div after outer div event call (its default set).

// by default flase is working in div addEventListener in Javascript (useCapture == true) if you set then otherwise useCapture == false ;
```

****

# DOM removeEventListener() Method :

```
Ex :  element.removeEventListener('dblonclick',functionName);
example :
document.getElementById("header").addEventListener("mouseleave", "abc");
document.getElementById("header").addEventListener("click","xyz");
function abc(){
 document.getElementById("header").style.background="red";
}
function xyz(){

document.getElementById("header").removeEventListener("mouseleave",abc);
}
```

0/t ===>   remove event method use then not call any event mathod .

 // its working on without click , But if you are click then remove event so it is working ..

****

JavaScript ClassList methods :

add(class1,class2,....)
remove(class1,class2,....)
toggle(class)
contains(class)
item(index)
Length

----

add method() : Any Event to call any class to addclass method use in Js.

Ex :

```
document.getElementById("header").addEventListener("click",abc);
function abc(){
 document.getElementById("header").classList.add("xyz","efg");
}
```

0/t ==>   Onclick add two class in abc classes .

```
class ="abc xyz efg";
document.getElementById("header").classList;
```

o/t ==> [0: abc , 1:xyz , 2:efg]

----

remove ClassList() : Its Uses class is remove .

```
document.getElementById("header").classList.remove("xyz");
// output ==>   remove class xyz from your classes.
```

Length() : Its uses find the length of classList() :

Ex :

```
document.getElementById("header").classList.length;
```

o/t ==> class count ..   (like return no of class = 2);

----

toggle(class) : toggle is used first time click add class after second time click when class is remove.

Ex :

```
document.getElementById("header").addEventListener("click",abc);
function abc(){
 document.getElementById("header").classList.toggle("xyz");
}
```

o/t ==>   onclick class abc and xyz both class add ,

after next onclick then remove xyz class from abc + xyz class.

so toggle is used class add onclick after remove on next click then use toggle.

----

item() : classList method Item is used to print classList value using index value then use it .

```
html code :  <div id="header" class="abc xyz efg"></div>
js code :
document.getElementById("header").addEventListener("click",abc);
function abc(){
 document.getElementById("header").classList.item(1);
}
o/t ===>   xyz (its Ist index class value return.)
```

Contains() : Its check class is or not. (if class is then return true ; otherwise retrun false.)

Ex :

```
html code :  <div id="header" class="abc xyz efg"></div>
js code :
document.getElementById("header").addEventListener("click",abc);
function abc(){
 document.getElementById("header").classList.contains("xyz");
}
o/t ===>   true
```

Ex :

```
html code :  <div id="header" class="abc xyz efg"></div>
js code :
document.getElementById("header").addEventListener("click",abc);
function abc(){
 document.getElementById("header").classList.contains("gfy");
}
o/t ===>   false
```

DOM Traversal Method :

parentNode
parentElement
Children
childNodes
firstChild
firstElementChild
lastChild
lastElementChild
nextElementSibling
nextSibling

previousElementSibling

previousSibling

****

parentElement() : Its return the targeting id with her parents id call it.

Ex :

```
var a = document.getElementById("inner").parentElement.style.background="red";
var a = document.getElementById("inner").parentElement;
var b = document.body.parentElement;
document.write(a);
document.write(b);
var a = document.getElementById("class-c").parentElement.style.background="red";
var a = document.getElementById("class-c").parentElement;
o/t ==>  <div id="outer">...</div>
   <html id="main"></html>
   <div id="inner">...</div>
```

ParentNode() :

```
var a = document.getElementById("main").parentElement.style.background="red";
var a = document.getElementById("main").parentElement;
var a = document.getElementById("inner").parentElement.style.background="red";
var a = document.getElementById("inner").parentElement;
output ==>  #document
   <div id="outer">...</div>
```

Parent Node and parent element only one difference if parent element have no parent then return Null .

But parent Node return #document its return somethings .

----

children() : its help to parent class to target her child class then using this one.

```
var a = document.getElementById("outer").children;
var a = document.getElementById("main").children;
var a = document.getElementById("main").children[1].style.background="red";
var a = document.getElementById("main").children[1].innerHTML;
```

o/t ==> inner , h2 all children tag of html inside print in array. (its return array of childrens)

outer

inside outer id print first index children return.

children() & childNodes() : Only one diffrence is childNodes return to all tags and text also but children return only tags of html.

----

childNodes() : childNodes return all tag of html & all text .

Ex :

```
var a = document.getElementById("main").childNodes;
var a = document.getElementById("main").childNodes[1];
var a = document.getElementById("main").childNodes[1].innerHTML;
```

o/t ==> text,h2,p,all paragraph tag and html tag also.

return text of First index values.

retrun text innerHTML of text of childNodes.

----

firstElementChild() : Its return the first element of child.

```
var a = document.getElementById("main").firstElementChild;
var a = document.getElementById("main").firstElementChild.innerHTML;
```

----

lastElementChild() : Its return the last element of child.

```
var a = document.getElementById("main").lastElementChild;
var a = document.getElementById("main").lastElementChild.innerHTML;
```

----

firstChild() : Its return the first text of child (like text) does not change any property of with the help.

```
var a = document.getElementById("main").firstChild;
var a = document.getElementById("main").firstChild.innerHTML;
```

----

lastChild() : Its return the last text of child (like text) does not change any property of with the help.

```
var a = document.getElementById("main").lastChild;
var a = document.getElementById("main").lastChild.innerHTML;
```

----

nextElementSibling() : Its return next element of class A after class B is next sibling.

nextSibling() : its retrun text after one class to next class.

previousElementSibling : Its return first element of class B before class a is first sibling.

previousSibling : its retrun text before one class to first class.
