JavaScript is a solution of client side dynamic pages.

-created by netscape 1990, initialy called as Livescript

-object-based scripting language

-is not compilation but translated.

The JavaScript Translator (embedded engine- javascript virtual machine in browser)

Different js engines:

chrome,opera -V8

Firefox-SpiderMonkey

IE-Trident,chakra

Edge-Chakracode

safari-Nitro,squirilfish

**How do engines work?**

1.The engine (embedded if it’s a browser) reads (“parses”) the script.

2.Then it converts (“compiles”) the script to the machine language.

3.And then the machine code runs, pretty fast.

**My first javascript program:**

<script type=”text/javascript”>

document.write("Hello JavaScript by JavaScript"); //print text in browser

</script>

**Comments:**

// single line comment

/\*multi lines comment\*/

**Escape sequence**

\’ \’’ \n \f \r \b \\ it need “pre” tag above script tag<pre><script> </script></pre>

**Interactions:**

alert()- alert(“hi hello”);

prompt()- prompt(“title”,”default text”); //2nd parameter must for IE

confirm()- confirm(“do want to quit”); //return boolean

open()- open(“google.com”);

close()- close(“google.com”);

**Three Places to put JavaScript code**

1.Between the body tag of html

<script type="text/javascript">

alert("Hello Javatpoint");

</script>

2.Between the head tag of html

3. By .js file (external javaScript)

|  |  |
| --- | --- |
| rajesh.html | hello.js |
| <html> <head>  <script type="text/javascript" src="hello.js"></script>  </head>  <body> </body> </html> | function msg(){  alert("Hello Javatpoint");  } |

**Variable:**Named storage for data

var variable\_name;

let variable\_name;

const CONST\_NAME; //cannot reassign

* The name must contain only letters, digits, symbols $ and \_.
* The first character must not be a digit.

Local variable: is declared inside block or function, accessible within the function or block only

Global variable: A JavaScript global variable is accessible from any function.

window.value=90; // To declare JavaScript global variables inside function

all global variable are window.variable\_name only. it is needed “window” when inside function

**Datatype:**

1.Primitive datatype- String,Number, Boolean,Undefined, Null

2.Non primitive datatype- Object, Array, RegExp

Note:

typeof varname; //return datatype of variable

typeof undefined; //undefined

typeof func; //function

typeof null; //object

null === undefined; //false

nul == undefined; //true

also typecasting can be done. var a=String(b); var a=Number(b); var a=Boolean(b);

var a=parseInt(“20hello”) //20 convert string to integer, parseFloat(); string to floating point

eval(“3+2\*5/7”) //evaluates string as js expression, operator precedence.

1**.**Number – both integer,floating point number. Special values like Infinity, -Infinity and NaN.

64Bit IEEE-754 format. Range: ±2.23×10^−308 to ±1.80×10^308 sixteen decimel digits

shorthand writing 7e5=700000; 7e-5=0.0000007

let a = 0b11111111; // binary form of 255 let b = 0o377; // octal form of 255 a==b//true

(NaN==NaN) is false

val=(0.1+0.2) is not equal to 0.3. so use val.toFixed(2) function //it gived 0.30

parseInt('100px') ; // 100

parseFloat('12.5em'); // 12.5

parseInt('12.3'); // 12, only the integer part is returned

parseFloat('12.3.4'); // 12.3, the second point stops the reading

parseInt('a123'); // NaN

2.String -”string”,’string’,`string`

Backticks allow a string as multiple lines string:

String wrapping – console.log(“total=${total}”);

UTF-8 format, not depend on html page language

Accessing string char: 1.str[0] 2.str.charAt(0) //we cant change char of string. str[0]=’h’//error

String search:

str=’idyamidn’; str.indexOf(‘id’);//0 str.indexOf(‘id’,2); //5.. the second parameter indicate where index to start search

str.lastIndexOf(“word”); //search in reverse order

str.includes(“word”); //return boolean

str.startsWith(“word”);str.endsWith(“word”); //return boolean

String substring: str=”stringify”

slice- str.slice(0,5);//strin str.slice(1);//from 1 print all other chars str.slice(-6,-4) ; //’in’ from reverse of string

substring- it allows str.substring(6,2); //slice returns empty but substring return ‘ring’

substr- return string with length. Str.substr(3,2); //return string from 3 position length of 2

3.Boolean

4.null – nothing,empty or value unknown

5.undefined – for unassigned values

6.object

7.symbol

**Array:**

1. var array\_name=[value1, value 2…. value N];

2. var array\_name=new Array(); array\_name[0]=11; array\_name[1]=22;

3. var array\_name=new Array(11,22);

delete arr[key]; //remove array index value;

JavaScript array also have string indexes. arr[‘name’]=’rajesh’; //this is assoc array

automatic assign array1[0],array1[1] value as null. if i directly give array1[3].

Note. we can use assoc array same as object. just using dot operator call it ok. make it fun da

Trailing comma: arr=[‘raj’,’vanakam’,’muthu’,]; //comma can come at end, same like object also allows trailing comma

pop-remove last position element push-insert in last position

shift-remove element at beginning unshift- insert element to beginning

Array copy- arr=[‘raj’]; arr2=arr1; arr=[‘raj’,’muthu’]; arr2; //also return two values. Thus array also reference. Not a value.

Implode: string(arr); return array value with comma seperator

**Operator:**

**Decision making:**

Note:"",0, null, undefined and NaN become false

1.if(expression){ //content to be evaluated }

2.if(expression){ //content } else{ //content }

3.if(expression1){ //content } else if(expression2){ //content } else if(exp3){ //content }

else{ //content }

4. switch(expression){

case value1: code to be executed; break;

case value2: code to be executed; break;

......

default: code to be executed if above values are not matched; }

5.Ternary operator:

(expression)?condition1:condition2;

(expression1)?condition1:(expression2)?condition2:condition3; //multiple ternary

**Looping:**

for loop- for (initialization; condition; increment) { code to be executed }

while loop - while (condition) { code to be executed }

do-while loop- do{ code to be executed }while (condition);

for-in loop- for(val in obj){ console.log(obj);} //return values

for-of loop- for(key of obj){ console.log(key); } //return key

**Functions:**

Traditional function : *function* function\_name([arg1, arg2, ...argN]){ //code to be executed }

Function Expression: *var* varname = *function*(){..........} //after declaration only we can call.

Callback function: we can pass a function itself pass as arguements for another function

*function* fun1(callbackfunction){ callbackfunction();}

*function* fun2(){....}

fun1(fun2); //calling the function passing another function as arguements

Immediately invoked function expressions (IIFE)

(function(){ ........... })(); //function calling

(function(a,b){ ............ }(22,33)); //with arguements

Function chaining: obj.func1().func(2); //calling two function seriesly using obj

**Class:**

There is no class in javascript, function name is class name

Ex.

function Book(){this.author=”raje”; this edition=8;}

var var\_name=new Book() //var\_name is object madr, Book is (class) or constructor

var\_name.author //raje

var\_name.edition //8

**Object:**

1.By object literal: var object\_name={name:”rajesh”, age:21} //can be called by obj.name

var object\_name={“name”:”rajesh”, “age”:21}

2.By creating instance of object: var objectname=new Object(); obj.name=”rajesh”;

3.By using literal var objname = {};

4.By using object **constructor**

<script>

function emp(id,name,salary){

this.id=id; this.name=name; this.salary=salary; }

e=new emp(103,"Vimal Jaiswal",30000);

document.write(e.id+" "+e.name+" "+e.salary);

</script>

Delete object property: delete objname.propertyname; delete objectname;

Note: last keyValue can be end with comma //obj={a:1,b:2,}

if object key is spaced string then use square brackets //obj={”my name”:”rajesh”}; console.log(obj[‘my name’]);

Computer properties: prop=”name”; obj={[prop]:”rajesh”}; //dynamic property key

Reserved words can use as object keys. //obj={let:”rajesh”}; but expect *\_\_proto\_\_ keyword*

key existence checks:

check undefined //if (obj.name === undefined) .but it fails when keyvalue is itself ‘undefined’ obj.name:’undefined’

using ‘in’ keyword //name in obj

‘for in’ loop //for (key in object) { object[key]}

Ascending order object: obj={22:2,44:4,11:11}; console.log(obj); //print 11,22,44

non integer property are listed on creation order

Object copies only reference not value. So less memory obj1=obj2;

Object comparison: === values can be same. But object of those values should be same;

Constant object: const usr={name:”raj”}; usr.name=”muthu”;//no error,but usr={name:”muthu”}; //cause error

Object cloning: method 1: for (let key in obj) { cloneObj[key] = obj[key]; }

method 2: Object.assign(cloneObj, sourceobj1, sourceobj2); //added to clone obj, if already key exist override it.

Garbage collection : if object override //obj=null;, then javascript destroy it by garbage collector

it uses mak-and-sweep algorithm;

objects are retain in memory while thy are reachable

This keyword:

obj={name:”raj”};

sayHi(console.log(this.name);) obj.sayHi(); //print raj

sayHi(console.log(obj.name);) obj2.sayHi(); //this also print raj

Note: calling if used function, without object. is a programming mistake

**JSON:**

Json.Stringify(obj); //sending data

Json.parse(obj); //receiving data

javasscript={name:”rajesh”} json={“name”:”rajesh”}

**Apply & call**

var person1 = {“name”=”rajesh”,fullname:function(){return this.name;}}

var person2 = {“name”:”muthu”}

person1.fullname.call(person2); //muthu

person1.fullname.apply(person2); //array(0->”muthu”)

**Access specifier:**

public- this keyword inside function (access in another function so it is public)

private- var keyword inside function (cannot access in another function outside)

**With keyword:**

it is used instead of using dot operator.

with(this){document.write(name); document.write(age);}

**Object prototype:**

use to assign values for many objects in case of same values, separate value giving to all objects is avoided

var var\_name=new Book();

Book.prototype.publisher=”tata”; // this value goes to all object at a time

**There are three main ways to pause a script:**

1. A breakpoint.

2. The debugger statements.

3. An error (if dev tools are open and the button is “on”)

**Time & Dates:**

Date() //"Sun Aug 26 2018 19:33:51 GMT+0530 (India Standard Time)"

new Date();

new Date(milliseconds); //starts from, 1st of 1970 UTC+0.

new Date("2017-01-26"); // Thu Jan 26 2017 05:30:00 GMT+0530 (India Standard Time)

new Date(year, month, day, hours, minutes, seconds, milliseconds)

//var var\_name=new Date();

document.write(var\_name); //Tue Jun 02 2015 07:03:50 GMT +530(Indian Standard Time)

Date objects:

getDate() -2 setDate(29) toLocaleString() -6/2/2015, 7:03:50AM

getMonth()-5 setMonth(7) toDateString()-Tue Jun 02 2015

getFullYear()-2015 setFullYear(2016)

getHours()-7 setHours(13)

getMinutes()-3 setMinutes(4)

getSeconds()-50 setSeconds(12)

Current Time: <span id="txt"></span>

<script>

var today=new Date();

var h=today.getHours();

var m=today.getMinutes();

var s=today.getSeconds();

document.getElementById('txt').innerHTML=h+":"+m+":"+s;

</script>

**Array functions**:

array1.length(); //does not work with assoc array

array1.concat(array2);

array1.join(seperator);

array1.reverse();

array1.sort();

array1.shift(); //delete first element

array1.unshift(new\_element); //add first element

array1.pop(); //delete last element

array1.push(new\_element); // insert new element

var var\_name= array1.slice(firstelement, lastlelement)

var var\_name=array1.splice(index\_position, replace elements);

**String functions:**

1.var string\_name=”rajesh”;

2.var string\_name=new String(“rajesh”);

string\_name.charAt(2)- “j”

string\_name.concat(string\_name2)

string\_name.indexOf(r) -0

string\_name.lastIndexOf()

string\_name.toLowerCase()

string\_name.toUpperCase()

string\_name.slice(2, 4) -“je”

string\_name.trim() - remove all unnecsssary spaces

**Math Object:**

Math.sqrt(n);

Math.random();

Math.pow(m,n);

Math.pi

Math.log(n);

Math.sin() Math.cos() Math.tan()

Math.min() Math.max()

Math.floor(n); gives nearby lowest integer

Math.ceil(n); gives nearby largest interger

Math.round(n); just remove value after dot, 4.6=4

Math.abs(n); modulus -4=4

**Number:**

Number object- var var\_name=new Number()value;

Number method- toExponential(x), to Fixed(x), toPrecision(x), toString(x), valueof()

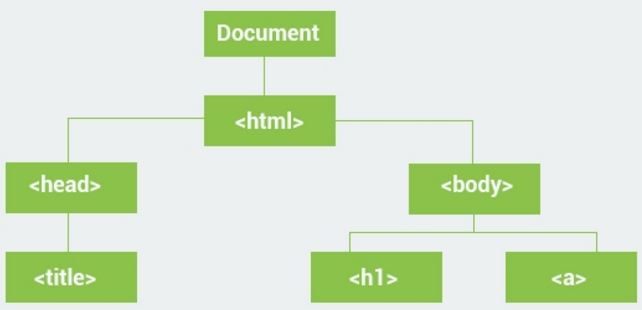
Number constants: MIN\_VALUE, MAX\_VALUE, POSITIVE\_INFINITY, NEGATIVE\_INFINITY, NaN

**Event handling**: 1.window, 2.mouse, 3.keyboard, 4. form

|  |  |  |
| --- | --- | --- |
| **Event** | **Events** | **Description** |
| Window event | onLoad  onUnLoad  onResize | when document has finished loading  when page is excited  when broswer window resized |
| Mouse event | onclick  ondblclick onmouseover onmousedown onmouseup onmouseout | when element is clicked.  when element is double-clicked.  when mouse is moved over an element.  when mouse button is pressed over an element.  when mouse is released (after mouse is pressed).  when mouse is moved out from an element |
| Keyboard event | onkeydown onkeyup onkeypress | occurs when key is being pressed.  occurs when key is released.  occurs when key up-down pair |
| Form event | onFocus onBlur onSelect onChange onInvalid onInput onReset onsubmit onload onunload onscroll onresized onreset | occurs when an element gets  occurs when form looses the focus from an element.  occurs when text get selected  occurs when text modified  occurs text not valid  occurs text written to input field  occurs text input is reset  occurs when form is submitted.  occurs when document, object or frameset is loaded.  occurs when body or frameset is unloaded.  occurs when document is scrolled.  occurs when document is resized.  occurs when form is reset. |

**Document Object Model: 1.Document 2.Element 3.Attributes 4.Events 5.style**

allows user to access parts of web page js code, broswer create DOM when page loaded



**document.getElementById(‘id’).innerHtml;**

getElementById=methodname

innerHtml =properyname

**Document-method:**

document.write(“string”)

document.writeln(“string”)

document.getElementById(“id”)

document.getElementByName(“name”)

document.getElements ByTagName(“name”)

document.getElementsByClassName()

document.querySelector(“ ”)

document.forms[‘formname’][‘inputtname’].value;

document.activeElement Returns the currently focused element in the document

document.addEventListener() Attaches an event handler to the document

document.adoptNode() Adopts a node from another document

document.anchors Returns a collection of all <a> elements in the document

document.applets Returns a collection of all <applet> elements in the document

document.baseURI Returns the absolute base URI of a document

document.body Sets or returns the document's body (the <body> element)

document.close() Closes the output stream previously opened with document.open() Open the outpur stream

document.cookie Returns all name/value pairs of cookies in the document

document.charset Deprecated. Use document.characterSet instead. document.characterSet Returns the character encoding for the document

document.createAttribute() Creates an attribute node

document.createComment() Creates a Comment node with the specified text

document.createDocumentFragment() Creates an empty DocumentFragment node

document.createElement() Creates an Element node

document.createTextNode() Creates a Text node

document.doctype Returns the Document Type Declaration associated

document.documentElement Returns the Document Element of the document (the <html> )

document.documentMode Returns the mode used by the browser to render the document

document.documentURI Sets or returns the location of the document

document.domain Returns the domain name of the server

document.domConfig Obsolete. Returns the DOM configuration of the document

document.embeds Returns a collection of all <embed> elements the document

document.forms Returns a collection of all <form> elements in the document

document.hasFocus() Returns Boolean value indicating whether document has focus

document.head Returns the <head> element of the document

document.images Returns a collection of all <img> elements in the document

document.implementation Returns the DOMImplementation object document.importNode() Imports a node from another document

document.inputEncoding Returns the encoding, character set, used for the document

document.lastModified Returns the date and time the document was last modified

document.links Returns a collection of all <a> and <area> have a href attribute

document.normalize() Removes empty Text nodes, and joins adjacent nodes

document.normalizeDocument() Removes empty Text nodes, and joins adjacent nodes

document.open() Opens an HTML output stream to collect output from document.querySelector() Returns the first element that matches a CSS selector(s)

document.querySelectorAll() Returns all elements that matches a specified CSS selector(s)

document.readyState Returns the (loading) status of the document

document.referrer Returns the URL of the document current page

document.removeEventListener() Removes an event handler from the document

document.renameNode() Renames the specified node

document.scripts Returns a collection of <script> elements in the document

document.strictErrorChecking Sets or returns whether error-checking is enforced or not

document.title Sets or returns the title of the document

document.URL Returns the full URL of the HTML document

document.write() Writes HTML expressions or JavaScript code to a document

document.writeln() Same as write(), but adds a newline character after each statem

document.attributes Documents don't have attributes

document.hasAttributes() Documents don't have attributes

document.nextSibling Documents don't have siblings

document.nodeName This is always #document

document.nodeType This is always 9 (DOCUMENT\_NODE)

document.nodeValue Documents don't have an node value

document.ownerDocument Documents don't have an owner document

document.ownerElement Documents don't have an owner element

document.parentNode Documents don't have a parent node

document.previousSibling Documents don't have siblings

document.textContent Documents don't have a text content

**Element-propery:**

innerHtml

attribute

style.property

chileElementCount()

createElement()

appendChild()

element.accessKey Sets or returns the accesskey attribute of an element

element.addEventListener() Attaches an event handler to the specified element

element.appendChild() Adds a new child node, to an element, as the last child node

element.attributes Returns a NamedNodeMap of an element's attributes

element.blur() Removes focus from an element

element.childElementCount Returns the number of child elements an element has

element.childNodes Returns a collection of an element's child nodes

element.children Returns a collection of an element's child element

element.classList Returns the class name(s) of an element

element.className Sets or returns the value of the class attribute of an element

element.click() Simulates a mouse-click on an element

element.clientHeight Returns the height of an element, including padding

element.clientLeft Returns the width of the left border of an element

element.clientTop Returns the width of the top border of an element

element.clientWidth Returns the width of an element, including padding

element.cloneNode() Clones an element

element.compareDocumentPosition() Compares the document position of two elements

element.contains() Returns true if a node is a descendant of a node, otherwise false

element.contentEditable Sets or returns whether content of an element is editable or not

element.dir Sets or returns the value of the dir attribute of an element

element.firstChild Returns the first child node of an element

element.firstElementChild Returns the first child element of an element

element.focus() Gives focus to an element

element.getAttribute() Returns the specified attribute value of an element node

element.getAttributeNode() Returns the specified attribute node

element.getElementsByClassName() Returns a collection of all child elements

element.getElementsByTagName() Returns a collection of all child elements element.getFeature() Returns an object which implements APIs of a specified feature

element.hasAttribute() Returns true if element has specified attribute, otherwise false

element.hasAttributes() Returns true if an element has any attributes, otherwise false

element.hasChildNodes() Returns true if an element has any child nodes, otherwise false

element.id Sets or returns the value of the id attribute of an element

element.innerHTML Sets or returns the content of an element

element.insertBefore() Inserts a new child node before a specified, existing, child node

element.isContentEditable Returns true if content of an element is editable, otherwise false

element.isDefaultNamespace() Returns true if a specified namespaceURI is the default

element.isEqualNode() Checks if two elements are equal

element.isSameNode() Checks if two elements are the same node

element.isSupported() Returns true if a specified feature is supported on the element

element.lang Sets or returns the value of the lang attribute of an element

element.lastChild Returns the last child node of an element

element.lastElementChild Returns the last child element of an element

element.namespaceURI Returns the namespace URI of an element

element.nextSibling Returns the next node at the same node tree level

element.nextElementSibling Returns the next element at the same node tree level

element.nodeName Returns the name of a node

element.nodeType Returns the node type of a node

element.nodeValue Sets or returns the value of a node

element.normalize() Joins adjacent text nodes, removes empty text nodes in element

element.offsetHeight Returns the height of an element

element.offsetWidth Returns the width of an element, including padding

element.offsetLeft Returns the horizontal offset position of an element

element.offsetParent Returns the offset container of an element

element.offsetTop Returns the vertical offset position of an element

element.ownerDocument Returns the root element (document object) for an element

element.parentNode Returns the parent node of an element

element.parentElement Returns the parent element node of an element

element.previousSibling Returns the previous node at the same node tree level

element.previousElementSibling Returns the previous element at the same node tree level

element.querySelector() Returns first child element that matches a CSS selector

element.querySelectorAll() Returns all child elements that matches a CSS selector(s)

element.removeAttribute() Removes a specified attribute from an element

element.removeAttributeNode() Removes a specified attribute node, and returns removed node

element.removeChild() Removes a child node from an element

element.removeEventListener() Removes an event handler with the addEventListener() method

element.replaceChild() Replaces a child node in an element

element.scrollHeight Returns the entire height of an element, including padding

element.scrollIntoView() Scrolls the specified element into the visible area of the browser

element.scrollLeft Sets or returns number of pixels element scrolled horizontally

element.scrollTop Sets or returns number of pixels element's scrolled vertically

element.scrollWidth Returns the entire width of an element, including padding

element.setAttribute() Sets or changes the specified attribute, to the specified value

element.setAttributeNode() Sets or changes the specified attribute node

element.style Sets or returns the value of the style attribute of an element

element.tabIndex Sets or returns the value of the tabindex attribute of an element

element.tagName Returns the tag name of an element

element.textContent Sets or returns the textual content of a node and its descendants

element.title Sets or returns the value of the title attribute of an element

element.toString() Converts an element to a string

nodelist.item() Returns the node at the specified index in a NodeList

nodelist.length Returns the number of nodes in a NodeList

**Attributes property:**

attr.isId Returns true if the attribute is of type Id

attr.name Returns the name of an attribute

attr.value Sets or returns the value of the attribute

attr.specified Returns true if the attribute has been specified,

nodemap.getNamedItem() Returns a specified attribute node from a NamedNodeMap

nodemap.item() Returns the attribute node at a index in a NamedNodeMap

nodemap.length Returns the number of attribute nodes in a NamedNodeMap

nodemap.removeNamedItem() Removes a specified attribute node

nodemap.setNamedItem() Sets the specified attribute node (by name)

attr.appendChild() Attributes don't have child nodes

attr.attributes Attributes don't have attributes

attr.baseURI use document.baseURI instead

attr.childNodes Attributes don't have child nodes

attr.cloneNode() Get or set the attr.value instead

attr.firstChild Attributes don't have child nodes

attr.hasAttributes() Attributes don't have attributes

attr.hasChildNodes Attributes don't have child nodes

attr.insertBefore() Attributes don't have child nodes

attr.isEqualNode() Makes no sense

attr.isSameNode() Makes no sense

attr.isSupported() Is always true

attr.lastChild Attributes don't have child nodes

attr.nextSibling Attributes don't have siblings

attr.nodeName Use attr.name instead

attr.nodeType This is always 2 (ATTRIBUTE\_NODE)

attr.nodeValue Use attr.value instead

attr.normalize() Attributes cannot be normalized

attr.ownerDocument This is always your HTML document

attr.ownerElement This is the HTML element you used to access the attribute

attr.parentNode This is the HTML element you used to access the attribute

attr.previousSibling Attributes don't have siblings

attr.removeChild Attributes don't have child nodes

attr.replaceChild Attributes don't have child nodes

attr.textContent Use attr.value instead

**Style property:**

alignContent Sets or returns the alignment between the lines inside container

alignItems Sets or returns the alignment for items inside a flexible container

alignSelf Sets or returns the alignment for selected items

animation A shorthand property for all the animation properties below

animationDelay Sets or returns when the animation will start

animationDirection Sets or returns whether animation should play in reverse on alternate

animationDuration Sets or returns how much time ,animation takes to complete one cycle

animationFillMode Sets or returns what values are applied by the animation outside

animationIterationCount Sets or returns the number of times an animation should be played

animationName Sets or returns a name for the @keyframes animation

animationTimingFunction Sets or returns the speed curve of the animation

animationPlayState Sets or returns whether the animation is running or paused

backgroundAttachment Sets or returns whether a background-image is fixed or scrolls

backgroundColor Sets or returns the background-color of an element

backgroundImage Sets or returns the background-image for an element

backgroundPosition Sets or returns the starting position of a background-image

backgroundRepeat Sets or returns how to repeat (tile) a background-image

backgroundClip Sets or returns the painting area of the background

backgroundOrigin Sets or returns the positioning area of the background images

backgroundSize Sets or returns the size of the background image

backfaceVisibility Sets or returns whether or not an element should be visible

border Sets or returns borderWidth, borderStyle, and borderColor

borderBottom Sets or returns all the borderBottom\* properties in one declaration

borderBottomColor Sets or returns the color of the bottom border

borderBottomLeftRadius Sets or returns the shape of the border of the bottom-left corner

borderBottomRightRadius Sets or returns the shape of border of the bottom-right corner

borderBottomStyle Sets or returns the style of the bottom border

borderBottomWidth Sets or returns the width of the bottom border

borderCollapse Sets or returns whether the table border collapsed into a single

borderColor Sets or returns the color of an element's border

borderImage A shorthand property for setting or returning all borderImage\* properties

borderImageOutset Sets or returns the amount by which the border image area extends

borderImageRepeat Sets or returns whether the image-border should be repeated

borderImageSlice Sets or returns the inward offsets of the image-border

borderImageSource Sets or returns the image to be used as a border

borderImageWidth Sets or returns the widths of the image-border

borderLeft Sets or returns all the borderLeft\* properties in one declaration

borderLeftColor Sets or returns the color of the left border

borderLeftStyle Sets or returns the style of the left border

borderLeftWidth Sets or returns the width of the left border

borderRadius A shorthand property for setting or returning all the four border

borderRight Sets or returns all the borderRight\* properties in one declaration

borderRightColor Sets or returns the color of the right border

borderRightStyle Sets or returns the style of the right border

borderRightWidth Sets or returns the width of the right border

borderSpacing Sets or returns the space between cells in a table

borderStyle Sets or returns the style of an element's border

borderTop Sets or returns all the borderTop\* properties in one declaration

borderTopColor Sets or returns the color of the top border

borderTopLeftRadius Sets or returns the shape of the border of the top-left corner

borderTopRightRadius Sets or returns the shape of the border of the top-right corner

borderTopStyle Sets or returns the style of the top border

borderTopWidth Sets or returns the width of the top border

borderWidth Sets or returns the width of an element's border

bottom Sets or returns the bottom position of a positioned element

boxDecorationBreak Sets or returns the behaviour of the background and border

boxShadow Attaches one or more drop-shadows to the box

boxSizing Allows you to define certain elements to fit an area in a certain way

captionSide Sets or returns the position of the table caption

clear Sets or returns the position of the element relative to floating objects

clip Sets or returns which part of a positioned element is visible

color Sets or returns the color of the text

columnCount Sets or returns the number of columns an element should be divided

columnFill Sets or returns how to fill columns

columnGap Sets or returns the gap between the columns

columnRule A shorthand property for setting or returning all the columnRule

columnRuleColor Sets or returns the color of the rule between columns

columnRuleStyle Sets or returns the style of the rule between columns

columnRuleWidth Sets or returns the width of the rule between columns

columns A shorthand property for setting or returning columnWidth

columnSpan Sets or returns how many columns an element should span across

columnWidth Sets or returns the width of the columns

content Used with the :before and :after pseudo-elements, to insert content

counterIncrement Increments one or more counters

counterReset Creates or resets one or more counters

cursor Sets or returns the type of cursor to display for the mouse pointer

direction Sets or returns the text direction

display Sets or returns an element's display type

emptyCells Sets or returns whether to show the border and background of empty

filter Sets or returns image filters (visual effects, like blur and saturation)

flex Sets or returns the length of the item, relative to the rest

flexBasis Sets or returns the initial length of a flexible item

flexDirection Sets or returns the direction of the flexible items

flexFlow A shorthand property for the flexDirection and flexWrap properties

flexGrow Sets or returns how much the item will grow relative to the rest

flexShrink Sets or returns how the item will shrink relative to the rest

flexWrap Sets or returns whether the flexible items should wrap or not

cssFloat Sets or returns the horizontal alignment of an element

font Sets or returns fontStyle

fontFamily Sets or returns the font family for text

fontSize Sets or returns the font size of the text

fontStyle Sets or returns whether the style of the font is normal, italic or oblique

fontVariant Sets or returns whether font should be displayed in small capital letter

fontWeight Sets or returns the boldness of the font 1

fontSizeAdjust Preserves the readability of text when font fallback occurs

fontStretch Selects a normal, condensed, or expanded face from a font family

hangingPunctuation Specifies whether a punctuation character may placed outside line box

height Sets or returns the height of an element

hyphens Sets how to split words to improve the layout of paragraphs

icon Provides author ability to style an element with an iconic equivalent

imageOrientation Specifies a rotation in right or clockwise direction applies to an image

justifyContent Sets or returns alignment between the items inside a flexible container

left Sets or returns the left position of a positioned element

letterSpacing Sets or returns the space between characters in a text

lineHeight Sets or returns the distance between lines in a text

listStyle Sets or returns listStyleImage, listStylePosition, and listStyleType

listStyleImage Sets or returns an image as the list-item marker

listStylePosition Sets or returns the position of the list-item marker

listStyleType Sets or returns the list-item marker type

margin Sets or returns the margins of an element (can have up to four values)

marginBottom Sets or returns the bottom margin of an element

marginLeft Sets or returns the left margin of an element

marginRight Sets or returns the right margin of an element

marginTop Sets or returns the top margin of an element

maxHeight Sets or returns the maximum height of an element

maxWidth Sets or returns the maximum width of an element

minHeight Sets or returns the minimum height of an element

minWidth Sets or returns the minimum width of an element

navDown Sets or returns where to navigate when using the arrow-down

navIndex Sets or returns the tabbing order for an element

navLeft Sets or returns where to navigate when using the arrow-left

navRight Sets or returns where to navigate when using the arrow-right

navUp Sets or returns where to navigate when using the arrow-up

opacity Sets or returns the opacity level for an element

order Sets or returns the order of the flexible item, relative to the rest

orphans Sets or returns the minimum number of lines for an element

outline Sets or returns all the outline properties in one declaration

outlineColor Sets or returns the color of the outline around a element

outlineOffset Offsets an outline, and draws it beyond the border edge

outlineStyle Sets or returns the style of the outline around an element

outlineWidth Sets or returns the width of the outline around an element

overflow Sets or returns what to do with content that renders outside element

overflowX Specifies what to do with the left/right edges of the content

overflowY Sets or returns the padding of an element (can have up to four values)

paddingBottom Sets or returns the bottom padding of an element

paddingLeft Sets or returns the left padding of an element

paddingRight Sets or returns the right padding of an element

paddingTop Sets or returns the top padding of an element

pageBreakAfter Sets or returns the page-break behavior after an element

pageBreakBefore Sets or returns the page-break behavior before an element

pageBreakInside Sets or returns the page-break behavior inside an element

perspective Sets or returns the perspective on how 3D elements are viewed

perspectiveOrigin Sets or returns the bottom position of 3D elements

position Sets or returns the type of positioning method used for an element

quotes Sets or returns the type of quotation marks for embedded quotations

resize Sets or returns whether or not an element is resizable by the user

right Sets or returns the right position of a positioned element

tableLayout Sets or returns the way to lay out table cells, rows, and columns

tabSize Sets or returns the length of the tab-character

textAlign Sets or returns the horizontal alignment of text

textAlignLast Sets or returns how last line or a line before a line break is aligned

textDecoration Sets or returns the decoration of a text 1

textDecorationColor Sets or returns the color of the text-decoration

textDecorationLine Sets or returns the type of line in a text-decoration

textDecorationStyle Sets or returns the style of the line in a text decoration

textIndent Sets or returns the indentation of the first line of text

textJustify Sets or returns justification method used when text-align is "justify"

textOverflow Sets or returns what should happen when text overflows

textShadow Sets or returns the shadow effect of a text

textTransform Sets or returns the capitalization of a text

top Sets or returns the top position of a positioned element

transform Applies a 2D or 3D transformation to an element

transformOrigin Sets or returns the position of transformed elements

transformStyle Sets or returns how nested elements are rendered in 3D space

transition A shorthand property for setting or returning four transition properties

transitionProperty Sets or returns the CSS property that the transition effect is for

transitionDuration Sets or returns how many seconds or milliseconds a transition effect

transitionTimingFunction Sets or returns the speed curve of the transition effect

transitionDelay Sets or returns when the transition effect will start

unicodeBidi Sets or returns whether the text support multiple languages i

userSelect Sets or returns whether the text of an element can be selected or not

verticalAlign Sets or returns the vertical alignment of the content in an element

visibility Sets or returns whether an element should be visible

whiteSpace Sets or returns how to handle tabs, line breaks and whitespace in a text

width Sets or returns the width of an element

wordBreak Sets or returns line breaking rules for non-CJK scripts

wordSpacing Sets or returns the spacing between words in a text

wordWrap Allows long, unbreakable words to be broken and wrap to next line

widows Sets or returns minimum number of lines for element visible at the top

zIndex Sets or returns the stack order of a positioned element

Ex.

<script>

function callme(){

document.getElementById(“D1”).style.backgroundColor=”blue”;

}

fucntion callme1(){

var b=document.getElementByClassName(“C1”);

b[0].style.color=”green”;

b[0].style.fontSize=”20px”;

b[1].style.weight=”Bold”; }

function callme2(){

var b1=document.getElementByTagName(“H1”);

b1[0].style.backgroundColor=”red”;

b1[1].style.backgroundColor=”green”; }

</script>

<body>

<div id=”D1” style=”background-color”red; heigh:100px”> hii</div>

<div class=”C1” style=”background-color:orange”>hello</div>

<div class=”C1” style=”background-color: yellow”>how are u</div>

<input type=”button” value=”click me first” onclick=”callem1()”/>

<input type=”button” value=”click me second” onclick=”callem1()”/>

<h1 if=”r1” onmouseover=”callme2()”> muthu</h1>

<h2 if=”r2” onmouseover=”callme2()”> rajesh</h1>

</body>

Other properties and method of DOM:

**document.innerHtml** use to change content of HTML element

document.getElementById(“D1”).innerHTML=”<h1>hello</h1>”;

**document.childElementCount()** used to count number of child elements

alert(document.getElementById(“B1”).childElementCount) //gives number of tags used

**document.createElement(“**tagname**”)** used to create new element

var a=document.createElement(“H2”);

**document.createTextNode(“**text**”)** used to create new text

var b=document.createTextNode(“hello”);

**appendChild()** used to join child element to parent

a.appendChild(b);

**removeChild()** used to remove a from parent

a.removechild(b);

**document.createAttribut()** used to create an attribute for element

**document.setAttributeNode()** used to set attribute to element

var b=documents.createElement(“H1”);

var a=document.createAttribute(“id”);

a.value=”p1”;

b.setAttributeNode(a);

**document.nextElementSibling()** used to return next element after current element

**document.previousElementSibling()** used to return previous element before current element

**EventListener:** DOM+event handling



element.getElementById(“idname”).event=handlers;

element.getElementById(“idname”).event=function fun\_name(){}; //anonymous event handler

Ex.

<body> <div id=”D1” style=”height:200px;width:200px”></div>

<div id=”D2” style=”height:200px;width:200px”></div>

<script>

document.getElementById(D1).onclick=callme;

function callme(){ alert(“hello mouse clicked”); }

document.getElementById(D2).onmouseover=function callover()

{ alert(“mouse over”);}; </script>

**Types of Event listeners:**

1.addEventListener()

element.addEventListener(event, function, event\_propagation);

2.removeEventListener()

element.removeEventListener(event, function, event\_propagation);

<script> window.onload=function()

{var a=document.getElementById(“H1”);

a.addEventListener(“click”,callme);

a.addEventListener(“mouseover”,overme);

a.removeEventListener(“mouseover”,overme);

function callme()

{this style.color=”green”; alert(“iam clicked”);}

}</script>

3.attachEvent(event,function) //only for internet explorer

4.detachEvent(event,function)

**Event Propagation:** defining order which event to occurs first

Bubbling- inner most element handled first, it is default, boolean value false

Capturing- outer most element handled first, boolean value true

**Regular Expression:**

var var\_name=/pattern/option; i=case ignore, g=used to match all pattern, m=multiple lines

var var\_name= new RegExp(“string”, “option”); // var a=new RegExp(“in”,”g”);

var\_name.test(string); //return true or false

var\_name.exec(string) //return values that matched

Ex:

var a=”rajes”;

var b=/jes/ig;

var\_name.match(var2\_name)

var\_name.search(var2\_name)

var\_name.replace(var2\_name)

Ex.

<script type=”text/javascript”>

var ip=”raj/nESh is good /neshwaran”;

var r=/esh/gm;

var op=ip.match(r); alert(op); //esh //eshwaran will not match

var ip=”rajesh is good eshwaran”;

var r=RegExp(“esh”,”gi”);

var op=ip.search(r); alert(op); //3 is the position of search “esh”

var ip=”rajesh is good eshwaran”;

var r=/good/;+

var op=ip.replace(r,”bad”); alert(op); //rajesh is bad eshwaran

**Metacharacter:**

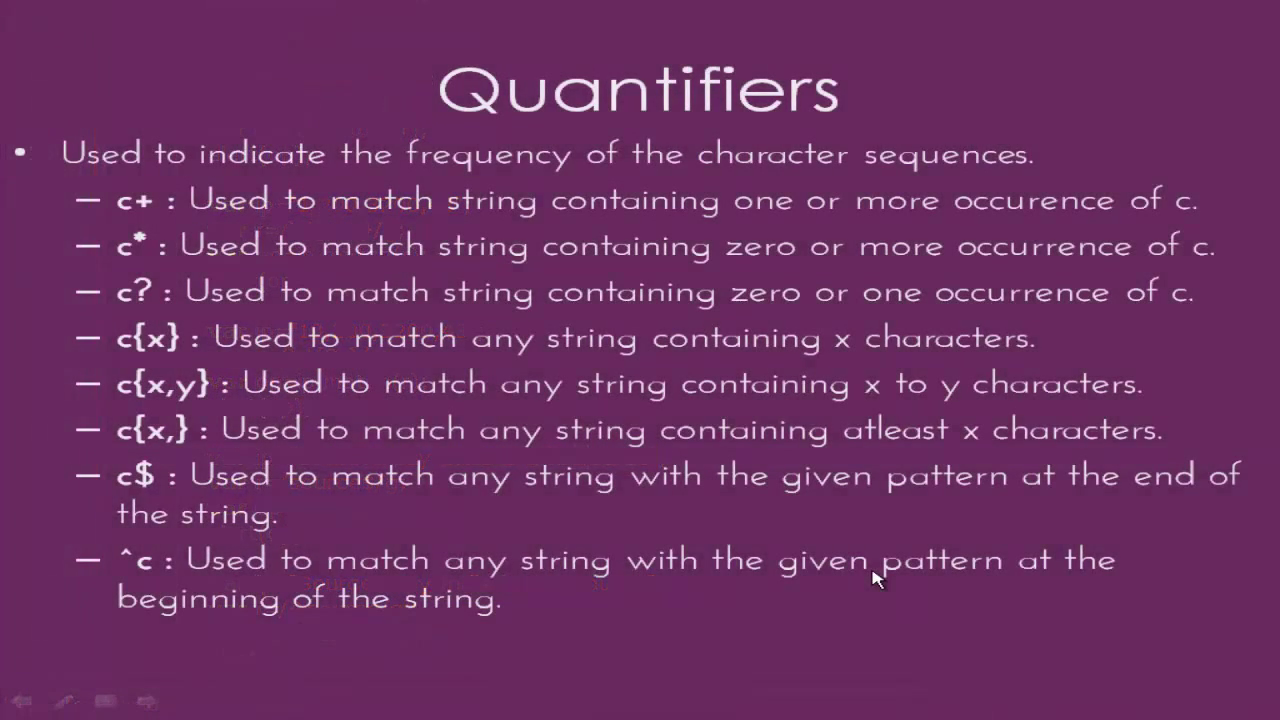
var ip=”RamSource\_100%”;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .(dot) | –find single char | var r=/o..c/g; | var op=ip.match(r); | //ourc |
| /w | -a-z A-Z 0-9 | var r=/\w+/g; | var op=ip.match(r); | //RamSource\_100 |
| /d | -non word char | var r=/\W/g; | var op=ip.match(r); | //% |
| /D | -0-9 | var r=/\d/g; | var op=ip.match(r); | //1,0,0 |
| /s | -white space count | var r=/\D/g; | var op=ip.match(r); | // , //no whitespace/ |
| /S | -non white char | var r=/\s/g; | var op=ip.match(r); | //R,a,m,S,o,u,r,c,e,\_,% |
| /b | -find match front back | var r=/\B/g; | var op=ip.match(r); | //Ram |
| /B | -find match middle | var r=/\B/g; | var op=ip.match(r); | //null |

Brackets:

var ip=”RamSource\_100%”;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [a-z] | match between a-z | var r=/[a-e]/g; | var op=ip.match(r); | // a,c,e |
| [0-9] | match between 0-9 | var r=/[0-2]/g; | var op=ip.match(r); | //1,0,0 |
| [~a-z] | match not between a-z | var r=/[^a-e]/g; | var op=ip.match(r); | //R,m,S,o,u,r,\_,1,0,0 |
| [~0-9] | match not between 0-1 | var r=/[^0-2-e]/g; | var op=ip.match(r); | //R,m,S,o,u,r,\_,1,0,0, |
| [a|b] | check if a or b present | var r=/[R|X]/g; | var op=ip.match(r); | //R |

**Quantifiers:**

**Form validation:**

<html> <head> <script type=”text/javascript”>

function callme()

{ var nme=document.getElementById(“fname”);

//var r1=/^[a-zA-z]+$/;

var r1=/^[a-zA-Z]\*\s{1}[a-zA-Z]\*$; //*accepting 1 space by using “\*”*

if(r1.test(nme.value)==false)

{

document.getElementByteId(“L1”).style.color=”red”;

document.getElementByteId(“L1”).style.fontSize=20; document.getElementByteId(“L1”).innerHTML=”Invalid name”;

return false; }

else{

document.getElementByteId(“L1”).style.color=”green”; document.getElementByteId(“L1”).style.fontSize=20; document.getElementByteId(“L1”).innerHTML=”valid name”;

return true; }}

function callage()

{ var nme=document.getElementById(“age”);

//var r1=/^[0-9]{1,2}$/; *//accepting only 2digit number as age*

f(r1.test(nme.value)==false)

{

document.getElementByteId(“L1”).style.color=”red”;

document.getElementByteId(“L1”).style.fontSize=20; document.getElementByteId(“L1”).innerHTML=”Invalid age”;

return false; }

else{

document.getElementByteId(“L1”).style.color=”green”; document.getElementByteId(“L1”).style.fontSize=20; document.getElementByteId(“L1”).innerHTML=”valid age”;

return true; }}

</script>

</head><body>

<form>

enter the name:<input type=”text id=”fname” size=20 onkeyup=”callmeup”/>

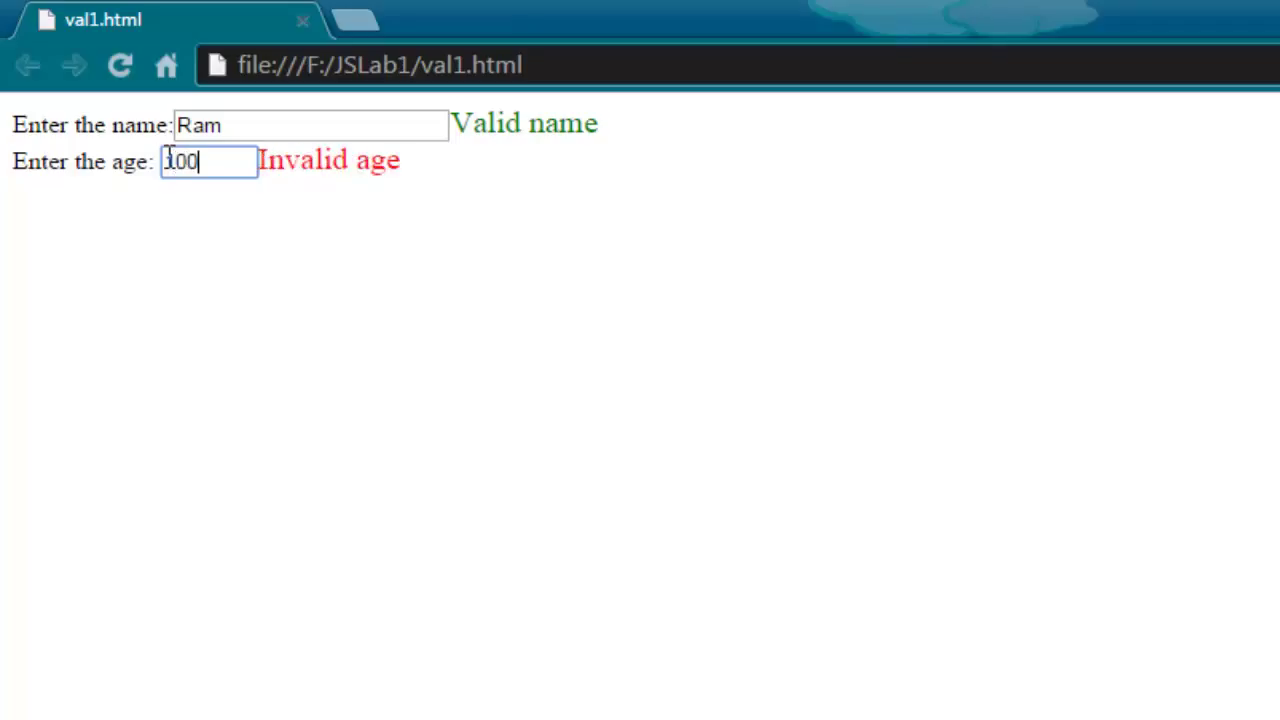
<label id=”L1”></label>

enter the age:<input type=”text id=”fname” size=20 onkeyup=”callageup”/>

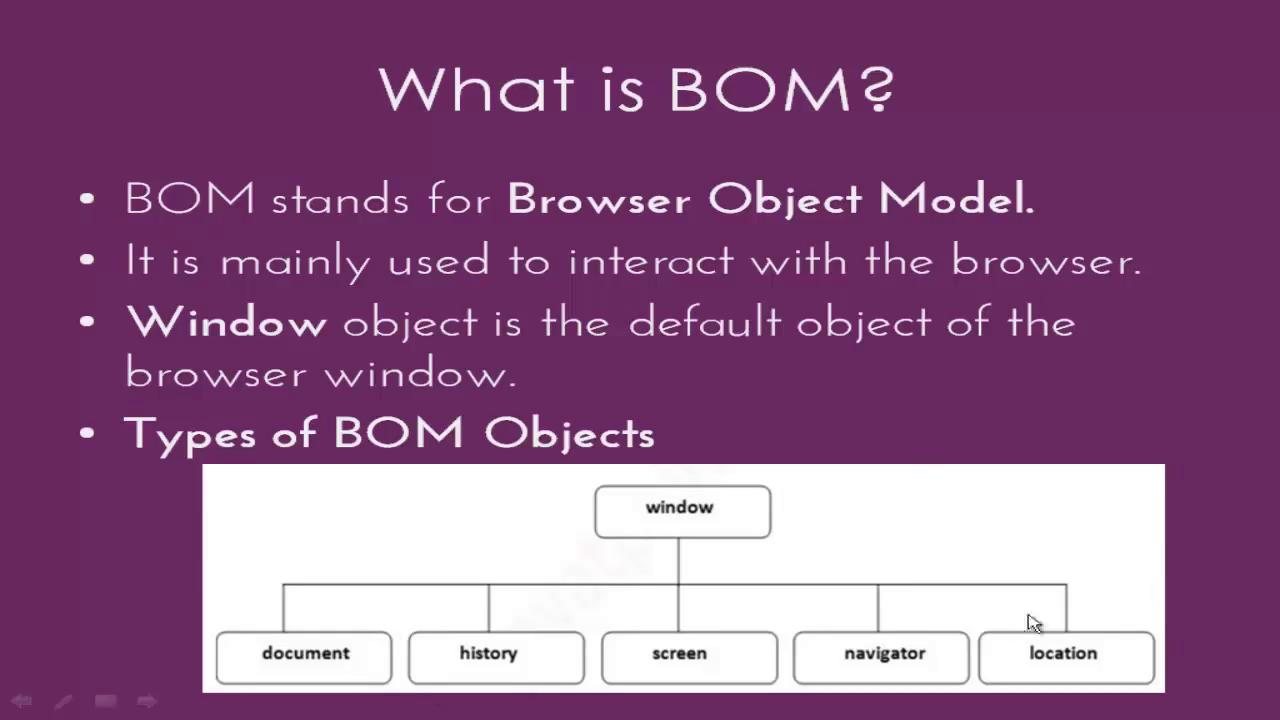
<label id=”L2”></label>

</form>

</body></html>



**BOM:** Browser Object Model



**Window Object:** it is default so we not use window.operator in front of this objects

alert()- alert(“hi hello”);

prompt()- prompt(“title”,”default text”);

confirm()- if(confirm(“do want to quit”)==true){....}else{....};

open()- open(“google.com”);

close()- close(“google.com”);

setTimeout(function, 2000);

**Window object:**

closed Returns a Boolean indicating whether a window has been closed or not

defaultStatus Sets or returns the default text in the statusbar of a window

document Returns the Document object for the window (See Document object)

frameElement Returns the <iframe> element in which the current window is inserted

frames Returns all <iframe> elements in the current window

history Returns the History object for the window (See History object)

innerHeight Returns the inner height of a window's content area

innerWidth Returns the inner width of a window's content area

length Returns the number of <iframe> elements in the current window

localStorage Allows to save key/value pairs in a web browserlocation

name Sets or returns the name of a window

navigator Returns the Navigator object for the window (See Navigator object)

opener Returns a reference to the window that created the window

outerHeight Returns the outer height of a window, including toolbars/scrollbars

outerWidth Returns the outer width of a window, including toolbars/scrollbars

pageXOffset Returns the pixels the current document has been scrolled (horizontally)

pageYOffset Returns the pixels the current document has been scrolled (vertically) parent Returns the parent window of the current window

screen Returns the Screen object for the window (See Screen object)

screenLeft Returns the horizontal coordinate of the window relative to the screen

screenTop Returns the vertical coordinate of the window relative to the screen

screenX Returns the horizontal coordinate of the window relative to the screen

screenY Returns the vertical coordinate of the window relative to the screen

sessionStorage Allows to save key/value pairs inbrowser. Stores data for one session

scrollX An alias of pageXOffset

scrollY An alias of pageYOffset

self Returns the current window

status Sets or returns the text in the statusbar of a window

top Returns the topmost browser window

**Windows method:**

alert() Displays an alert box with a message and an OK button

atob() Decodes a base-64 encoded string

blur() Removes focus from the current window

btoa() Encodes a string in base-64

clearInterval() Clears a timer set with setInterval()

clearTimeout() Clears a timer set with setTimeout()

close() Closes the current window

confirm() Displays a dialog box with a message and an OK and a Cancel button

focus() Sets focus to the current window

getComputedStyle() Gets the current computed CSS styles applied to an element

getSelection() Returns a Selection object representing the range of text selected by user

matchMedia() Returns a MediaQueryList object representing CSS media query string

moveBy() Moves a window relative to its current position

moveTo() Moves a window to the specified position

open() Opens a new browser window

print() Prints the content of the current window

prompt() Displays a dialog box that prompts the visitor for input

resizeBy() Resizes the window by the specified pixels

resizeTo() Resizes the window to the specified width and height

scroll() Deprecated. This method has been replaced by the scrollTo() method.

scrollBy() Scrolls the document by the specified number of pixels

scrollTo() Scrolls the document to the specified coordinates

setInterval() Calls a function or evaluates an expression at specified intervals

setTimeout() Calls a function or evaluates an expression after a specified number

stop() Stops the window from loading

**Location object property & methods**

location.hash Sets or returns the anchor part (#) of a URL

location.host Sets or returns the hostname and port number of a URL

location.hostname Sets or returns the hostname of a URL

location.href Sets or returns the entire URL

location.origin Returns the protocol, hostname and port number of a URL

location.pathname Sets or returns the path name of a URL

location.port Sets or returns the port number of a URL

location.protocol Sets or returns the protocol of a URL

location.search Sets or returns the querystring part of a URL

location.assign() Loads a new document

location.reload() Reloads the current document

location.replace() Replaces the current document with a new one

**Screen Object property and methods:**

screen.width used to return total width of screen

screen.height used to return total height of screen

screen.availWidth return width excluding toolbar, taskbar etc

screen.availHeight return height excluding toolbar, taskbar etc

screen.colorDepth

screen.pixelDepth

**History object property and methods:**

histroy.back(); // previous page

history.forward(); //next page

histor.length //number of history

history.go(2); //next 2nd page

history.go(-2); //previous 2nd page

**Navigator:**

navigator.appName //return browser name

navigator.appVersion //return browser version

navigator.product //return javascript translater name

navigator.appCodeName //code name

navigator.cookieEnabled //return true if cookie enabled

navigator.userAgent //return header http

navigator.language //return browser language

navigator.platform //return os name

navigator.userLanguage

navigator.plugins

navigator.systemLanguage

navigator.mimeTypes[]

navigator.online

navigator.javaEnabled()

navigator.taintEnabled()

**Timing object**

var a= window.setTimout(funtionname().milliseconds); //will call after given time

window.clearTimeout(a);

var a=window.setInterval(functionname(), milliseconds); //will call each given time intervel

**Boolean:**

Boolean b=new Boolean(value);

Boolean method: toSource(), toString(), valueOf()

**Location object:**

Properties: location.hostName used to return name of the host

location.href used to return url of window

location.protocol used to return protocol used in current window

Methods: location.assign() used to load new document

location.reload() used to reload an existing document

**Exeception Handling:**

try

catch

throw

finally

**Ajax**

var xhttp = new xmlHttpRequest();

xhhtp/onreadystatechange = function(){

if(this.readystate ==4 && this.status ==200){

document.getElementById(“demo”).innerHtml=this.responseText; }

}

};

xhttp.open(“GET”,”page.php”,true);

xhttp.send();

**Readystate:**

|  |  |  |
| --- | --- | --- |
| **Value** | **State** | **Description** |
| 0 | UNSENT | readystate not intialized |
| 1 | OPENED | server connection established |
| 2 | HEADERS\_RECEIVED | request received |
| 3 | LOADING | processing request |
| 4 | DONE | request finished and response ready |

**Http Response:**

|  |  |  |
| --- | --- | --- |
| **Information Response** | | |
| 100 | Continue | everything ok client should continue or ignore |
| 101 | Switching protocol | response for upgrade request header |
| 102 | Processing | server received and process request but no response |
| **Successful Response** | | |
| 200 | Ok | request is success |
| 201 | Created | request success , new resource has created. ex PUT |
| 202 | Accepted | request received but not yet acted upon |
| 203 | Non-Authoritative information |  |
| 204 | No content | There no content for the request |
| 205 | Reset Content | tell user agent reset document view |
| 207 | Multi status | conveys info about multiple resources |
| 208 | Multi status |  |
| 226 | Im Used |  |
| **Redirect Messages** | | |
| 300 | Multiple choice | request has more than one possible response |
| 301 | Moved permanently | requested resource has been changed |
| 302 | Found | requested resource has changed temporarily |
| 303 | See Other | requested resource at another uri |
| 304 | Not Modified | used for caching |
| 305 | Use Proxy | requested reponse must accessed by proxy |
| 306 | Unused | not in use |
| 307 | Temporary Redirect | redirect client to another uri which is temporary |
| 308 | Permanent Redirect | redirect client to another uri which is permanently |
| **Client Error Reponses** | | |
| 404 | Bad Request | server could not understand request |
| 401 | Unauthorized | un authenticated request |
| 402 | Payment Required | not used currently |
| 403 | Forbidden | un authenticated, but server know id of client |
| 404 | Not Found | cannot find requested resource |
| 405 | Method Not Allowed | requested method disabled in server |
| 406 | Not Acceptable | server content negotiation, doesnt find conten |
| 407 | Proxy Authentication required | 401 but authentication needed |
| 408 | Request Timeout | for idele browser connection |
| 409 | Conflict | request conflicts with current state of server |
| 410 | Gone | requested content permanently delted |
| 411 | Lenght Required | content-lenght header field not defined |
| 412 | Precondition Failed | indicated preconditions in header, server not meet |
| 413 | Payload Too Large | request entity is large than limits by server |
| 414 | URI Too Long | URi is too long than server is willing to interpret |
| 415 | Unsupported Media Type | media format not supported |
| 416 | Requested Range Not Satifiable | range header field in request cannot fulfilled |
| 417 | Exeception Failed | Exception indicated in header cant meet by server |
| 418 | I’m a teapot | server refuses attempt to brew coffe with teapot |
| 421 | Misdirected Request | server not abled to produce response |
| 422 | Unprocessable Entity | unable to followed due to semantic errors |
| 423 | Locked | resource accessed is locked |
| 424 | Failed Dependency | request failed due to failed of previous request |
| 426 | Upgrade Required | server refuses to perform the request |
| 428 | Procondition Required | server require request to conditional |
| 429 | Too Many Requests | user has sent too many requests |
| 431 | Request Header Fields Too Large | server is unwilling to process the request |
| 451 | Unavailable For Legal Response | user request am illegal resource |
| **Server Error Responses** | | |
| 500 | Internal Server Error | server doesnt know how to handle |
| 501 | Not Implemented | request method is not suppoerted by server |
| 502 | Bad Gateway | invalid response |
| 503 | Service Unavailable | server not ready to handle request |
| 504 | Gateway Timeout | server acting as gateway and cannot get response |
| 505 | HTTP Version Not Supported | Http version not supported |
| 506 | Variant Also Negotiates | internal configuration error |
| 507 | Insufficient Storage |  |
| 508 | Loop Detected | server detected infinite loop |
| 510 | Not Extended | further extension of request are require to fulfill |
| 511 | Network Authentication Required | need authenticate to gain network access |