

Javascript

Netscape developed Livescript

↳ jointly with Sunmicrosystems

Three layers

1. Core - includes operators, expressions, statements & subprograms
2. Client side - collection of objects that support control of browsers & interaction with ~~the~~ users
3. Server side - collection of objects that make the language useful on the web server.

Difference between Java & Javascript

Java	Javascript
i) Programming language	Scripting language
ii) Object-oriented language	Based on object model, but is not object-oriented.
iii) Static-typing (strongly)	Dynamically typed
iv) Objects are static (methods & data members are fixed at compile time).	Objects are dynamic (methods & data members can be changed).

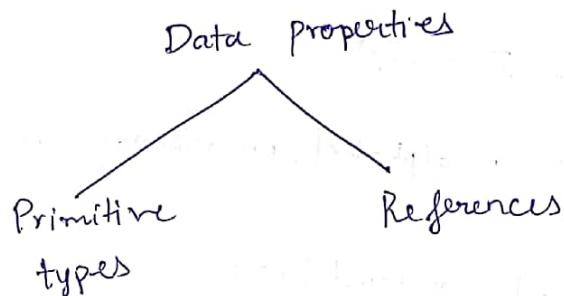
Advantages of Javascript

- i) Can be embedded in XHTML document
- ii) Using DOM model, Javascript can access & modify CSS & contents of XHTML document.
Because of this, the static web model becomes dynamic
- iii) Used to create cookies
- iv) Can be used to provide validation
- v) Supports event-driven programming

- vi) Mainly used to interact with users
- vii) Used to detect visitor's browser & can load the webpage accordingly.

JavaScript Object

- ↓
collection of properties
↳ numbers of class
- 1. Data property → "Properties"
 - 2. Method property



Objects can be accessed by using the variables & these variables act like references to the object.

Eg. Person.salary
↓ ↳
Variable property
(reference to
object)

Including JavaScript in XHTML

① <body>
<script type = "text/javascript">
=
</script>
</body>

Q. <body>
<script type = "text/javascript" src = "myPage.js">
</script>
</body>

Comments in JavaScript

- i) Single-line — //
- ii) Multiline —
 - a) /*
= */
 - b) <!-- . -->

Rules of Identifiers in JavaScript

- i) First character can be an alphabet, underscore or \$.
- ii) It is case-sensitive
- iii) No limit on the length of identifiers

Keywords

try, catch, finally, break, continue

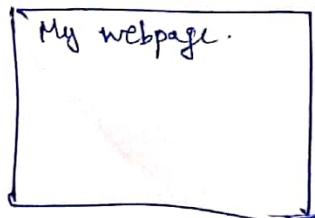
var → not in C or Java

function, delete

E.g. 1

```
<body>
<script type = "text/javascript" >
    document.write("My webpage");
</script>
</body>
```

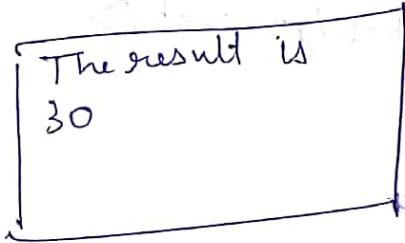
Output



E.g. 2

```
<body>
<script type = "text/javascript">
    var a,b,c;
    var string1;
    a = 10;
    b = 20;
    c = a+b;
    string1 = "Addition of 2 nos. " + c;
    document.write("The result is " + string1);
    document.write(c);
</script>
</body>
```

O.P.



Primitive types

Number

String

Boolean

} wrapper objects

undefined → If value is not assigned, then it shows undefined.

null → keyword

Literals (constants)

- i) Numeric - Integers, floating-point - 10, 2, 10, 10.5e-2, etc.
- ii) String - 'Hello', "Hello" (single or double quotes).

Operators

- Conditional
- Relational ($=$, $!$, $<$, $>$, $<=$, $>=$)
- Logical
- Arithmetic

Mathematical functions in Math object

Syntax: Math. fn (, , ,)
 - $\text{sqrt}(n)$, $\text{max}(x,y)$, $\text{min}(x,y)$, $\sin(x)$, $\cos(x)$, $\tan(x)$, $\text{ceil}(x)$, $\text{floor}(x)$

Q. Write Javascript code to find square root of a number & max & min of two numbers.

```

<body>
<script type = "text/javascript">
  var a, b;
  a = 25;
  b = 16;
  document.write(Math.sqrt(a));
  document.write(Math.min(a,b));
  document.write(Math.max(a,b));
</script>
</body>
  
```

Number object

1. MAX-VALUE
2. MIN-VALUE
3. POSITIVE-INFINITY
4. NEGATIVE-INFINITY

} document.write('Number.POSITIVE
INFINITY');

↓

O/P

Infinity

String concatenation operation

+

Methods

- toUpper Case()
- toLower Case()
- charAt(index value)
- substrng(begin, end)

Property

length

<body>

<script type = "text/javascript">

```

var
var st1 = "Hello";
var st2 = " webpage";
document.write(st1 + st2);
document.write(st1.toUpperCase());
document.write(st2.toLowerCase());
document.write(st1.charAt(3));
document.write(st1.substring(2,4));

```

</script>^{docu}

</body>

O/P

Hello webpage

HELLO

l

ll

5

typeof operator

Used to determine primitive type of the variable
typeof (varname)

<script type = "text/javascript">

Var no = 100;

Var st1 = "hai";

Var bool = true;

Var data1 = new Date();

Var obj1;

document.write(typeof(no));

document.write(typeof(st1));

document.write(typeof(bool));

document.write(typeof(data1));

document.write(typeof(obj1));

</script>

o.p

~~numbers~~ Number

~~string~~ String

Boolean

Object

Undefined

Date Object

UTC / GMT

Methods : `toString()`

`getDate()`

`getMonth()`

~~`getFullYear()`~~

`getHours()`

`getMinutes()`

`getSeconds()`

`getDay()`

```
var data1 = new Date();
```

```
document.write(data1.toString());
```

```
document.write(data1.getDate());
```

```
document.write(data1.getMonth());
```

```
document.write(data1.getFullYear());
```

```
document.write(data1.getMinutes());
```

Implicit type conversion

"XX" + 10 "XY10"

12 * "5" → 60

12 * "Hello" → NaN

Explicit type conversion

Q Give an example for converting the number to string

Var no = 6;
Var s1 = no.toString(); → 6
or

Var s2 = String(6); → 6

Var s3 = no.toString(2); → 110
↓ radix

Two methods used to separate numeric data from alphanumeric value

- parseInt()
- parseFloat()

Var nol = parseInt("12abc");

document.write(nol); → 12

parseInt("abc12"); NaN

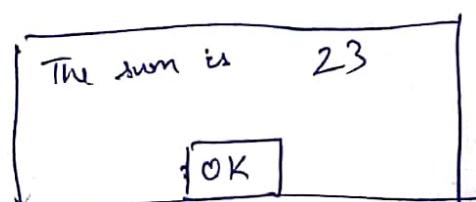
Var no2 = parseFloat("3.5 abcd"); → 3.5

Screen output & keyboard input

- Alert method
- Confirm method
- Prompt method.

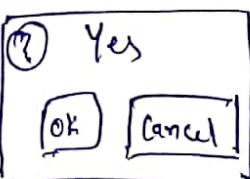
Alert method

1. Opens dialog with one button
2. alert("The sum is "+result + "ln");



Confirm

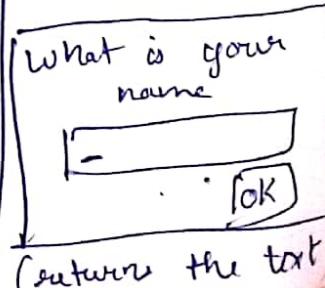
2 buttons dialog box
confirm("Yes");



(Store in variable
returns 1 or 0)

Prompt

Dialog with text box.
prompt("What is your name?")



Write a program to create a confirm box. If it is OK, display some message using alert, otherwise display prompt & read some text & display it using alert.

```
var con = Confirm ("Confirm ?");  
if (con == 1) {  
    if (Confirm ("Confirm ?"))  
        alert ("Confirmation complete");  
    else {  
        var name = prompt ("Enter your name");  
        alert (name);  
    }  
}
```

Write a program to take input of 3 numbers and print their maximum.

```
var a = parseInt (prompt ("number1"));  
var b = parseInt (prompt ("number2"));  
var c = parseInt (prompt ("number3"));  
var alert ("The max of 3 numbers" + Math.max (a, b, c));
```

Control Expression

- i) If the value of a control is a string, it is interpreted as true, else it is false.
- ii) If the value is a number (non-zero), it is true. If it is 0 it is false.
- iii) == } checks for data types
!= } does not check for data types
- iv) != } does not check for data types

$$"3" \neq 3 \rightarrow \text{false}$$

$$"3" \neq 3 \rightarrow \text{true}$$

List out the selection statements supported in Javascript

- if
- if else
- else if ladder
- switch

Loops

- for
- while
- do-while

Write a program to print a number & its square using while loop.

```
Var a = 1
while (a <= 10) {
    document.write(a)
    document.write(" " + a * a)
}
</table>
```

```
<table border = "3">
    <tr>
        <th>No </th>
        <th> Square </th>
    </tr>
    <script>
        var a = 1
        while (a <= 10) {
            document.write("<tr>");
            document.write("<td>" + a + "</td>");
            document.write("<td>" + (a*a) + "</td>");
            document.write("</tr>");
        }
    </script>
</table>
```

Objection Creation & Modification

- Q. Write a program to print the month names using switch statement. Use the method getMonth() of Date object

```
var date1 = new Date();
var month = date1.getMonth();
var monthName = month;
switch (month) {
    case 0:
        monthName = "Jan"; break;
    case 1:
        monthName = "Feb"; break;
    :
    case 11:
        monthName = "Dec"; break;
}
```

document.write("Month name: " + monthName);

Object creation and Modification

- i) In javascript, new operator creates a blank object with no properties
- ii) Javascript objects do not have properties & constructor creates & initialize the properties
- iii) Number of properties of javascript object are dynamic

Creation of object

```
var obj1 = new object();
```

- a. Create a ~~Person~~ Person object with two properties - name and age.

```
var person = new object();
```

```
person.name = "Varun";
```

```
person.age = 25;
```

```
document.write("Name " + person.name + " Age: "
+ person.age);
```

Another notation (JSON):

```
var person = {name: "XXX", age: 25}
```

- a. Print the properties of an object ~~using~~ using a for-loop.

```
for (var p in person)
```

```
document.write("Pname: " + p + " Pvalue: " +
person[p]);
```

O.P.	PName : Name	Pvalue : XXX
	PName : Age	Pvalue : 25

Array

It is an object

person [name] = "XXX" \cong person.name = "XXX"

\rightarrow var arr1 = new Array (1, 2, "30", "40");

\rightarrow $\underbrace{\text{1, 2, "30", "40"}_{\text{all strings}}}_{\text{does not have to be same datatype}}$

Array.length $\Rightarrow \rightarrow 4$

\rightarrow var arr2 = new Array (10) \rightarrow only one element \rightarrow size.

\rightarrow var arr1 = [10, 20, 30, 40]

i) Array elements can be primitive values or references to other objects

ii) JavaScript arrays have dynamic length.

E.g. If size is 100, but only 50 are initialized, then memory is only allocated for 50.

Array methods

i) Join() - Joins all elements in an array & returns a string

E.g. var arr1 = new Array ("XX", "YY", "ZZ");
arr1.join ":";

O/P: "XX:YY:ZZ"

ii) Concat()

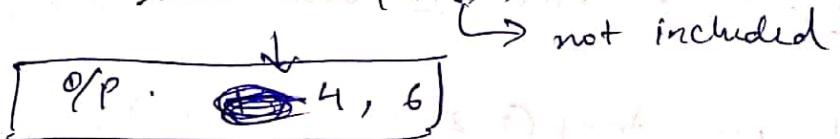
var str2 = arr1.concat ("T", "P");

O/P arr1 has XX, YY, ZZ, T, P

iii) Slice

var arr = [1, 4, 6, 8, 10];

arr.slice(1, 3); (indices)



iv) push()

v) pop()

vi) shift()

vii) unshift()

viii) sort()

var arr = new Array(11, 12, 13)

document.write(arr.pop()); // 13

arr.push(33);

arr.shift(); → remove beginning element

arr.unshift(11) → put back in the

last ~~last~~ shifted position

Q. Write a program to print the values of 1D dimensional array.

var arr = [1, 4, 6, 8, 10];

for (var i = 0; i < arr.length; i++)

document.write("Element" + (i + 1) + " : " + arr[i] + "\n")

2D Array

Var arr = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];

Print using 2 for-loops

Functions

i) Functions are objects in javascript

ii) JavaScript checks neither type nor number of parameters in a function call.

iii) Extra actual arguments are ignored

iv) Extra formal parameters remain undefined

v) All objects are passed by reference

```

<script type="text/javascript">
    function f1(x, y) {
        document.write("Inside function ");
        return (x * y);
    }
    var res = f1(10, 20);
    document.write("Result: " + res);
</script>

```

- Q. Write a program to create an Car object with the properties make, model and year. Create a function to change the value of the make property. Print the properties before & after the function call. Do not return anything from the function.

```

<script type="text/javascript">
    function f1(obj) {
        obj.make = "Honda";
    }
    var car = { make: "Hyundai", model: "Verna", year: 2016 };
    document.write("Before function ");
    document.write("Make: " + car[make] + " Model: " + car[model] + " Year: " + car[year]);
    f1(car);
    document.write("After function ");
    document.write("Make: " + car[make] + " Model: " + car[model] + " Year: " + car[year]);
</script>

```

Q. Pass an array to a function.

```
function Con(a) {  
    a.concat(12);  
}  
var arr = [10, 11];  
document.write("Before fn");  
for (for var i=0; i<2; i++)  
    document.write(arr[i] + " ");  
document.write("After fn");  
Con(arr);  
for (var i=0; i<2; i++)  
    document.write(arr[i] + " ");
```

Constructor

It is a special method used to create & initialize values for the properties of an objects.

When the constructor is called 'this' keyword is a reference to the newly created object.

Q. Define the object student , with the data properties name & age & initialize them using a constructor.

<script>

```
function student(sname, sage) {  
    this.name = sname;  
    this.age = sage;  
    this.display = stdInfo();  
}  
function stdInfo() {  
    document.write(this.name + " " +  
        this.age);  
}
```

```
var stud = new Student("XXX", 25);  
stud.display();
```

</script>

① P

XXX 25

Pattern Matching using Regular Expressions

There are two approaches for pattern matching in JavaScript

- Based on pre-defined RegExp object → not in syllabus
- Based on methods of String object



- Search() method

- i) takes the pattern as parameter
- ii) It will return the position in the string object where the pattern is matched; if there is no match it will return -1.

E.g.

```
var str = "Java Program";  
var i = str.search(/Program/);  
if (i >= 0)  
    document.write("Match found at " + (i + 1));  
else  
    document.write("Match not found");
```

Character & Character class Pattern

- Meta characters are characters that have some special meaning that can be used while specifying patterns

E.g. /3\..4/ → 3.4

/3..4/ → 34, 374, 384
↓
wildcard

- 2) $[abc] \rightarrow$ only abc
 $[a-h] \rightarrow$ a to h
 $[^aeiouv] \rightarrow$ only consonants

④ Predefined character classes.

Name	Matches
\d	A digit
\D	Non Digit
\w	Alphanumeric
\W	Non alphanumeric
\s	Whitespace
\S	Non whitespace

Q 1) $\backslash d \backslash . \backslash d \backslash d / \rightarrow 3.74$

2) $/ \backslash D \backslash d \backslash D / \rightarrow x5y$

3) $/ \backslash w \backslash w \backslash w / \rightarrow ab3$

4) $/ xy \{4\} z / \rightarrow xyyyyz$

3) Symbolic quantifiers

* - 0 or more

+ - 1 or more

? - 1 or 0

$/ *xy + z? /$

~~$/ \backslash d \backslash . \backslash d^* /$~~

ii) match () method

- Returns the matches as an array object. If there is no match returns null.

```
var st = "443.82";  
var i = st.match(/\d{1}.\d{1}\d{1}/);  
document.write(i);
```

O/P \Rightarrow 3.82

```
var st1 = "xyz";  
var i1 = st1.match(/xy\{4\}z/);  
document.write(i1);
```

O/P null

Anchors

A pattern must match at a particular position in a string.

```
var s1 = ". lang Prog";  
document.write(s1.match(/^\Prog/));
```

O/P Prog

/lang \$/

↳ end of the string

"Prog lang"

Modifiers
Specified as letters just after the right delimiter of
the pattern

E.g. /Apple/i → APPLE, APPle

/g → global

var s1 = "abc1), abcdy, abeghi";

s1.replace(/abc/g, "xyz");

O/P - xyz1), xyzdy, xyzghi

search()
match()
replace()
split()

Split method

var s1 = "xx:yy:zz";

s1.split(":");

O/P - xx, yy, zz

Errors in Script

Javascript interpreter is capable of detecting errors in
Javascript & these are the syntax errors.

Internet Explorer

Tools → Internet options → Advanced tab

unchecked {
 Disable script Debugging
 Display a notification about every script error
 box

Firefox

Javascript console or error console

Tools → Error console

Write a pattern validation for the following code

- First character - F, S, T, B
- 3-digit college code
- Branch code (2-digits)
- 4-digit candidate

~~var~~ s1 ; // code entered
s1.match ([FSTD]\d{3}\d{2}\d{4});

Javascript & HTML document

Javascript execution environment

i) Window object - Used for displaying the window

In javascript, all variables are data properties for some ~~window~~ object.

ii) Document object - Represents XHTML documents

Every window object has a property called 'document' which is a reference to Document object

Window object - property array called frames

Document object - property array called forms

Form object - property array called 'elements'

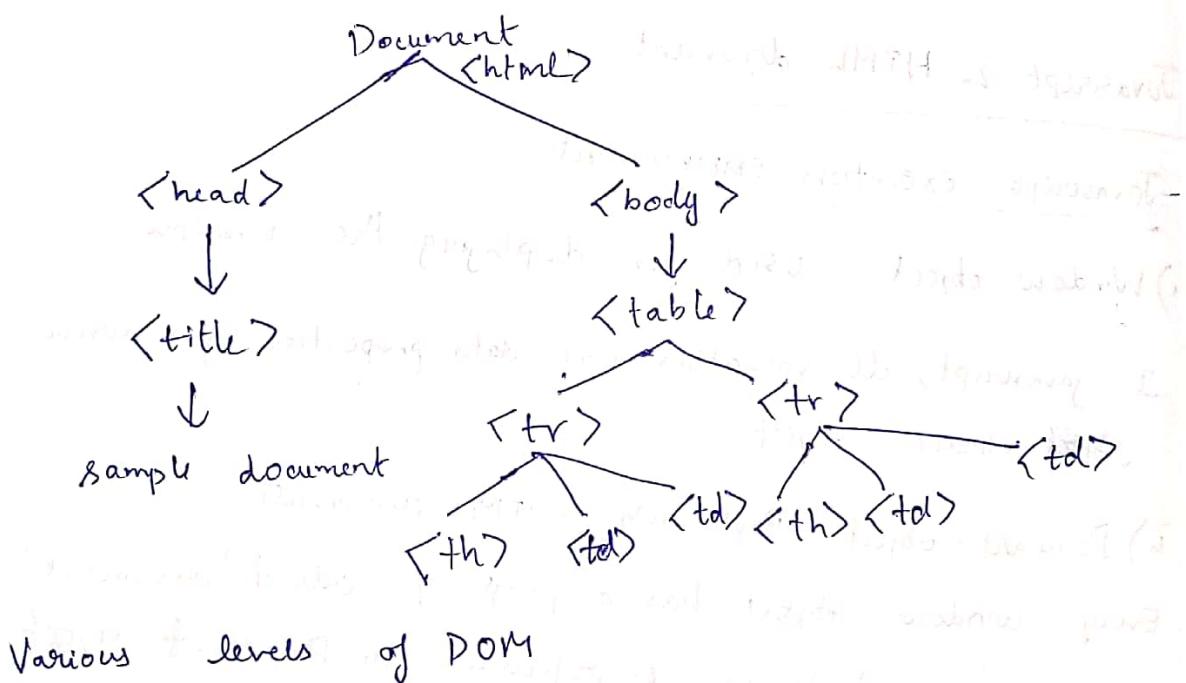
Document object - property arrays for anchors, links, images

DOM

- Document Object Model
- Provides the mapping for various components of the web document through objects, & these objects are accessible through javascript

It is an API that defines the interface between XHTML document & application program

Documents in DOM are represented using tree-like structure in which every element is represented as a node



DOM 0 - Implemented in Netscape 3.0 & IE 3.0 browser

DOM 1 - 1998, focused on XHTML & XML

DOM 2 - 2000, specifies the style sheet & also supports event model & traversal within documents

DOM 3 - 2004, deals with XML with DTD schemas, document validations etc.

DOM 4 - 2015

Accessing elements in Javascript

There are three methods

Consider

```
< form name = "form1" >  
< input type = "text" name = "t1" />  
</form>
```

Methods -

1. Every XHTML document is associated with some address called DOM address

```
var obj1 = document.form[0].elements[0];
```

Disadvantage - If elements are rearranged, the script needs to be changed.

Note: DOM 0 uses this

2. Use the name attribute in order to access the element

```
var obj = document.form1.t1;
```

Disadvantage - XHTML 1.1 does not support the 'name' attribute of the form tag & it will cause validation problem.

3. Access the desired element from web document using the javascript method getElementById()

```
var obj1 = document.getElementById("idvalue");
```

Note: This method is defined in DOM 1.

Events and Event handling

② 1st specification model - provided by HTML 4.0 standard
 ↓
~~DOM 0~~ event model

Event - Activity

Event handler - Script ~~file~~ that gets executed in response to ~~given~~ events

The process of connecting the event handler to an event is called event registration

It can be done using two methods

1. Assigning the tag attribute
2. Assigning the handler address to object properties

Event attributes & Tags

Event	Tag attribute
blur	onblur
click	onclick
change	onchange

Handling events from body elements

- Q. Write a program to display the alert message when the page is loaded.

```
HTML
<head>
  <script type="text/javascript" src="Ext.js">
    </script>
</head>
```

```
<body onload="f1()">>  
</body>
```

JS file (Ex1.js)

```
function f1() {  
    alert("hi");
```

}

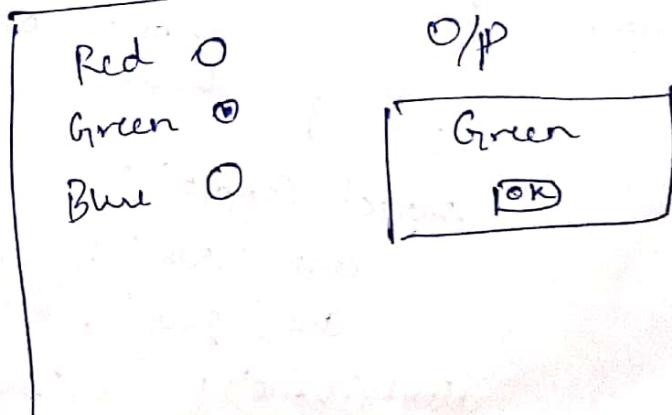
Handling events from button elements

Write a program using the following radio buttons. When ~~selected~~ a radio button is clicked, display its value on an alert. Write both XHTML & JS in same file.

XHTML



```
<html>  
    <head>  
        <title> select color </title>  
    </head>  
    <body>  
        <script type="text/javascript">  
            function f1(obj){  
                alert(obj.value);  
            }  
        </script>  
        <form name="form1">  
            Red <input type="radio" name="r1" value="red" onclick="f1(this)"/>  
            Green <input type="radio" name="r1" value="green" onclick="f1(this)"/>  
            Blue <input type="radio" name="r1" value="blue" onclick="f1(this)"/>  
        </form>  
    </body>  
</html>
```



Q. Write a program (for ~~the~~ previous example) for accessing the element using DOM object.
DOM object can be ~~obtained~~ obtained by using the method `getElementById()`.

```
<body>
  <form id="f1">
    Red <input type="radio" name="g1" value="red" />
    Green <input type="radio" name="g1" value="Green" />
    Blue <input type="radio" name="g1" value="Blue" />
  </form>
  <script type="text/javascript">
    var obj = document.getElementById("f1");
    obj.elements[0].onclick = f1; } Registering the events
    obj.elements[1].onclick = f2; }
    obj.elements[2].onclick = f3; }
  </script>
```

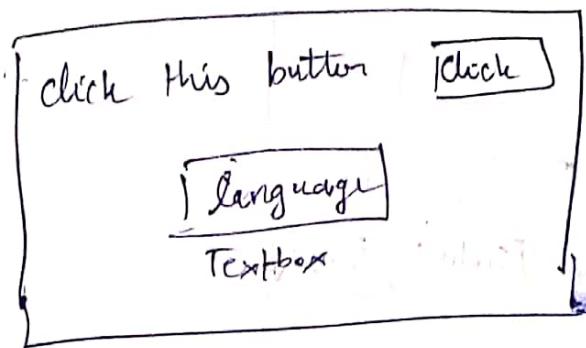
~~<body>~~

* `<head>`

```
<script>
  function f1() {
    var obj = document.getElementById("f1");
    for (var choice; choice = obj.g1[i].value; )
      if (obj.g1[i].checked) {
        choice = obj.g1[i].value;
        break;
      }
    switch (choice) {
      case "red": alert(choice);
      case "green": alert(choice)
    }
  }
</script> </head>
```

obj.g1[]
↓
Implicit arrow

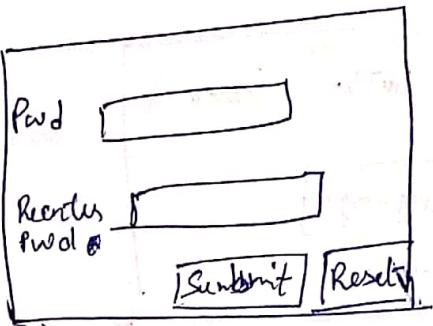
Handling events from Textbox & Password element



```
<body>
<form id = "form1">
    <label> Click this button <input type = "button" value = "click" onclick = "f1()" /> </label>
    <input type = "text" id = "t1" name = "textbox" onfocus = "blur()" />
</form>
</body>

<html>
    <head>
        <script>
            function f1() {
                document.getElementById("t1").value = "language"
            }
        </script>
    </head>
```

Q. Make a password form



Validating Form I/P

```
<html>
  <head>
    <script type="text/javascript" src="p1.js">
      </script>
    </head>
  <body>
    <form id="f1">
      Enter Pwd: <input type="password" value="" id="pwd"/>
      Re-enter Pwd: <input type="password" value="" id="repwd"/>
      <input type="submit" value="Submit" name="submit"/>
      <input type="reset" name="reset"/>
    </form> <script src="p2.js"> </script>
  </body>
</html>
```

p1.js → ~~Reg.~~ Event handling function.

```
function f1() {  
    var p1 = document.getElementById("pwd");  
    var p2 = document.getElementById("su-pwd");  
  
    if (p1.value == "") {  
        alert("Enter Pwd");  
        p1.focus();  
        p1.select();  
        return false;  
    }  
}
```

p2.js

```
document.getElementById("su-pwd").onblur=f1;  
document.getElementById("su-pwd").onsubmit=f1;
```

Test 2

DOM 2 Event Model

1. It is a complex event model
2. It does not include the features of DOM0 model

Module	Event Interface	Event types
HTML Events	Event	→ abort, blur, change focus, load, unload, select, etc
Mouse Events	MouseEvent	→ click, mouse up, mouse down, mouse out, mouse over

Ex2.js // Event handler for 1st text box

```
function chkName(event) {  
    var myname = event.currentTarget;  
    var pos = myname.value.search(/ /);  
}
```

Ex3.html

A hand-drawn diagram of an HTML form. It consists of a large rectangular box containing two smaller rectangular input fields labeled "Name:" and "Phone:", and two rectangular buttons labeled "Submit" and "Reset" at the bottom.

Test Questions

Q. Write a JavaScript code to ~~accept~~ validate VSN

- i) Digit from 1-4
- ii) 2 Uppercase chars
- iii) 2 digits
- iv) 2 uppercase chars
- v) 3 digits

start to
end

$\sim [1-4][A-Z]\{2\} [0-9]\{2\}[A-Z]\{2\}[0-9]\{3\}$

Q. Validate name

First name - 5-15

Middle name - 1-15

Last name - 5-15



```
var str = name.split(" ");  
var s1 = str[0].match(/[A-Z]{5-15});  
var s2 = str[1].match([A-Z]1);  
var s3 = str[2].match(/[A-Z]{5-15});
```

Q. Write a program to print the numbers in descending.

```
function fn(a, b) {  
    if (a < b)  
        return false  
    else  
    }  
    return true
```

```
var arr1 = new Array(12, 44, 23, 34, 09);  
arr1.sort();
```

O/p

12
23
34
44
9

Q. Write a program to print the name & version of the browser & name using the Navigation object.

navigator.appName
navigator.appVersion