Web Application Development

JavaScript

Outline

- Concepts of JavaScript
- Basic Data Types
- Variables
- Flow Control Statements
- Objects
- Functions
- Events
- Forms

What is JavaScript?

- A programming language
- A client-side script language, executed by Web browsers
- An interpreted language

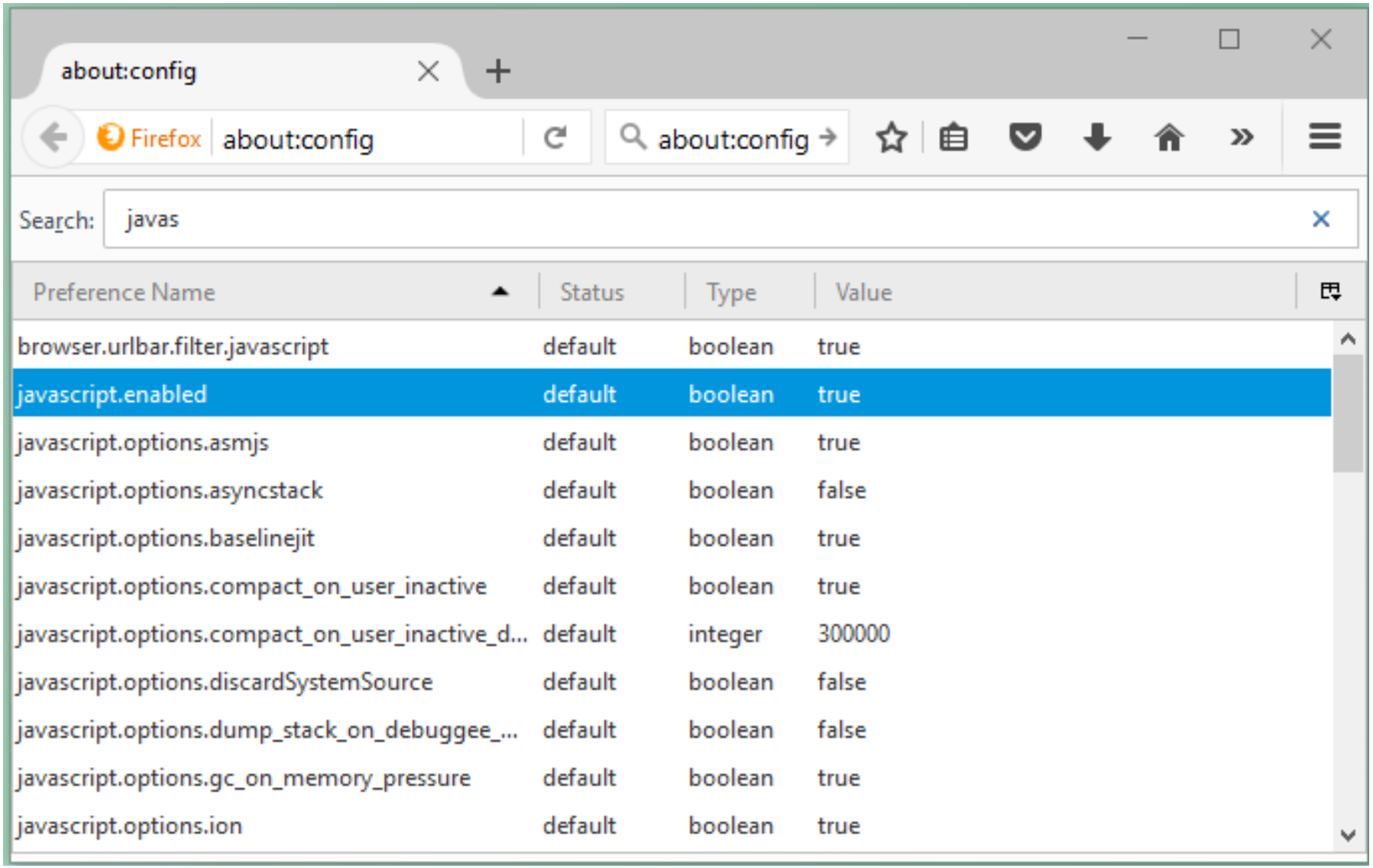
What is it used for?

- HTML: define content of the web page
- CSS: specify layout of the web page
- JavaScript: define behavior of the web page

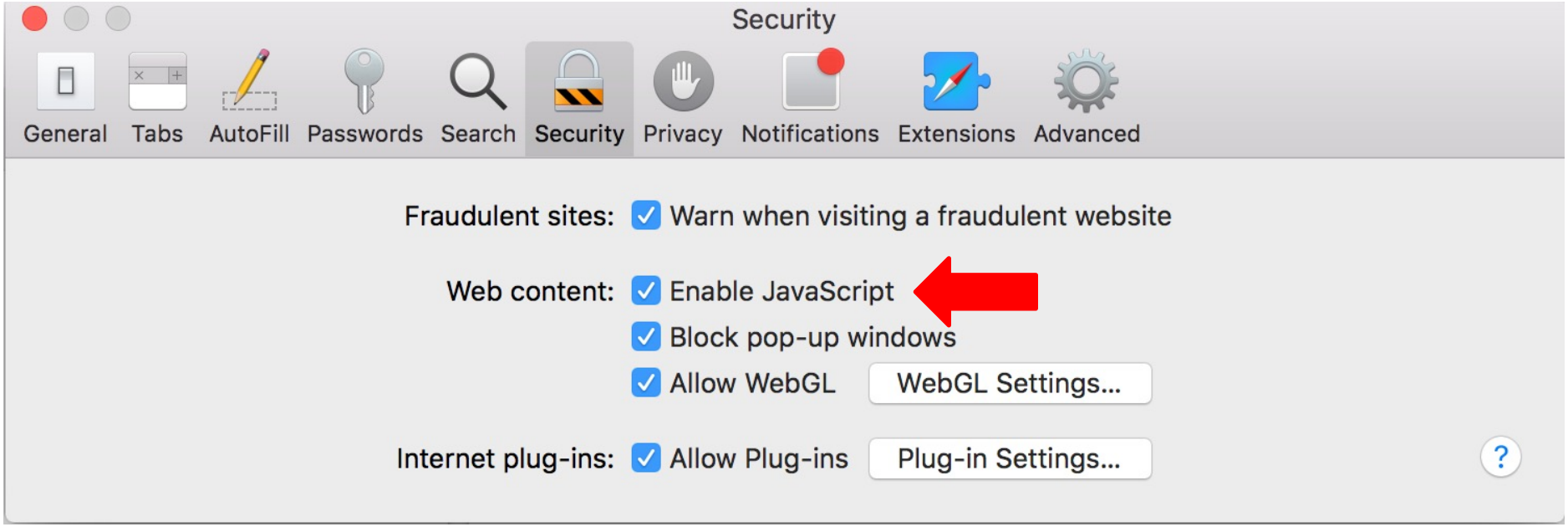
What is it used for?

- Handling user interaction
- Doing calculations/manipulations of forms input data
- Search a small database embedded in the downloaded page
- Manipulate cookies
- Generating dynamic HTML documents

Enable JavaScript-Firefox

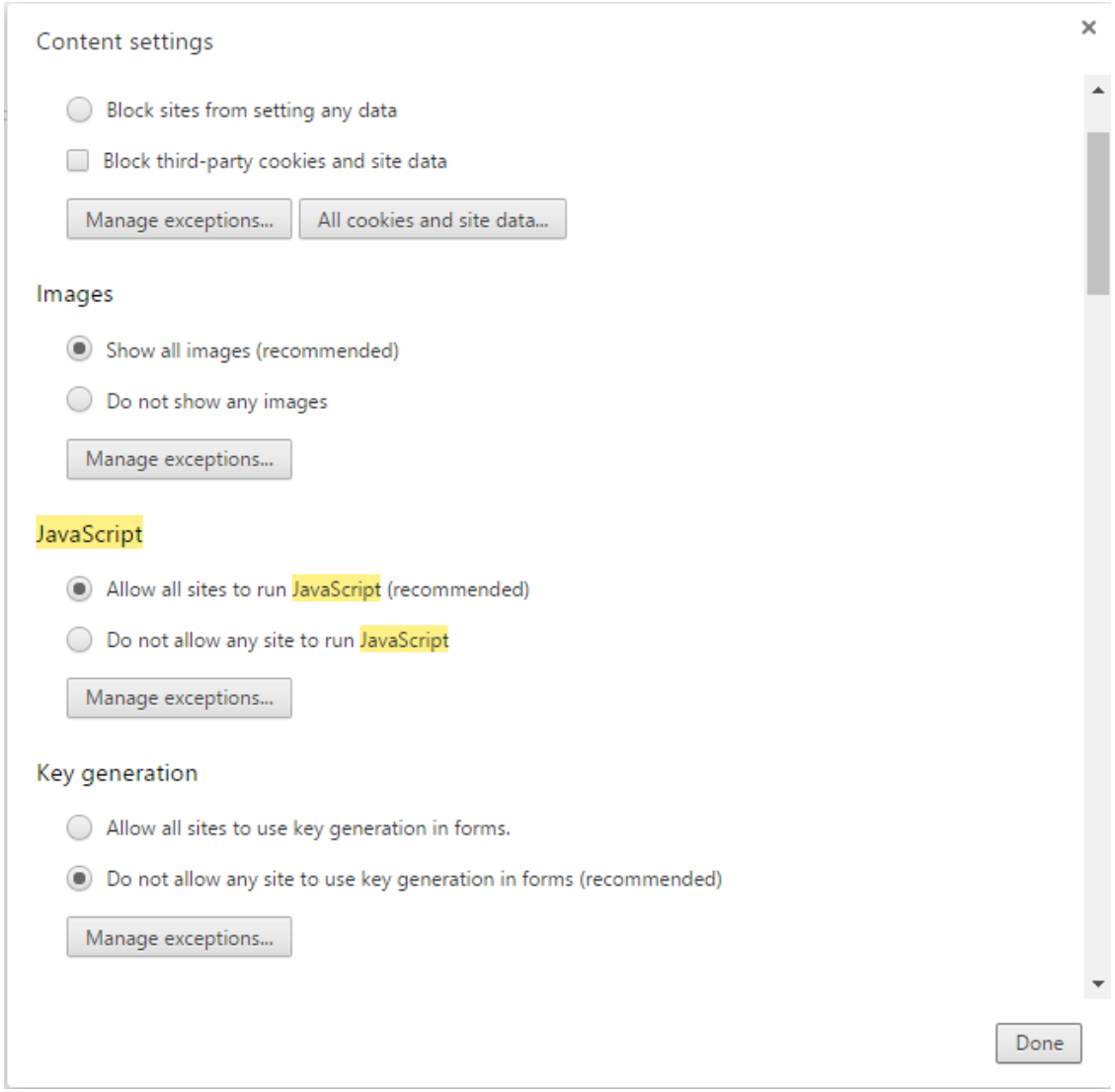


Enable JavaScript-Safari



Safari -> Preferences -> Security -> Web Content

Enable JavaScript-Chrome



Adding JavaScript to HTML

- JavaScript codes can be placed in the <body> section, in the <head> section, or in an external file
- JavaScript code must be put in the <script> tag when put inside HTML page

```
<script>
 document.getElementById("demo").innerHTML =
"My First JavaScript Code";
</script>
```

Placing JavaScript to Head Section

```
<html>
   <head>
         <title>First JavaScript </title>
         <script>
               function myFunction() {
document.getElementById("demo").innerHTML=          "Welcome to
the World of JavaScript";
               }
         </script>
   </head>
   <body>
         <p id=“demo">Javascript</p>
         <button type="button" onclick=“myFunction()">
         Click here</button>
   </body>
</html>
```

Placing JavaScript to Body Section

```
<html>
    <body>
        <p id=“demo”>Javascript</p>

        <button type="button" onclick=“myFunction()">
                Click here</button>

        <script>
            function myFunction() {
document.getElementById("demo").innerHTML=          "Welcome to
the World of JavaScript";
            }
        </script>

   </body>
</html>
```

Placing JavaScript to External Files

- Here is my external file: myScript.js

```
function myFunction() {
        document.getElementById(“demo”).innerHTML =
        "Welcome to the World of JavaScript";
}
```

```
<html>
<body>
     <p id="demo">Javascript</p>
     <button type="button" onclick=“myFunction()">
     Click here</button>
     <script src="myScript.js"></script>
</body>
</html>
```

JavaScript Comments

- Single line: //comments
- Multiple lines: /* comments */

Basic Types of Data

- Numerical data
  - Integer numbers: 74, -10
  - Floating-point numbers: 1.234
- Text data: “hanoi”, ‘USTH’, “USTH’s website”
- Boolean data: true, false

Example: Displaying text data

```
<html>
   <head>
         <title>JavaScript Objects</title>
   </head>
   <body>
         <h2 align=“center”>
         <script type="text/javascript">
               var name=prompt("What is your name?");
               document.write("Hello " + name);
         </script>
         </h2>
   </body>
</html>
```

Example: Displaying text data

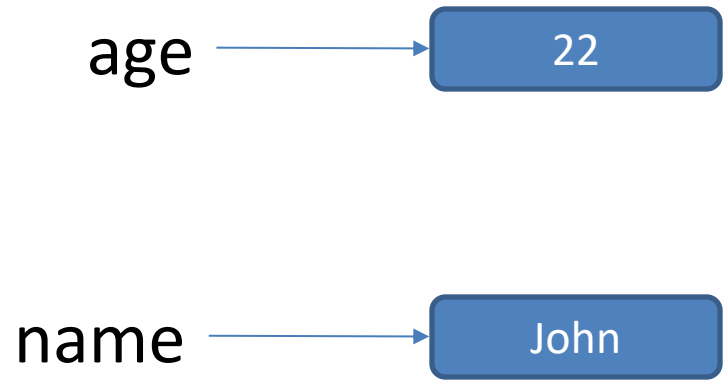
- **document** is the *object* representing the HTML document
- **write** is a *method* which writes text in HTML document

```
<script type="text/javascript">
   var name=prompt("What is your name?");
   document.write("Hello "+name);
</script>
```

Example: Adding two numbers

```
<script type="text/javascript">
        var str1=window.prompt("First number:");
        var str2=window.prompt("Second number:");
        var n1=parseInt(str1);
        var n2=parseInt(str2);
        var sum=n1+n2;
        document.write("The first number: "+str1+"<br\>");
        document.write("The second number: "+str2+"<br\>");
        document.write("Sum: "+sum);
</script>
```

Variables



- Declaration of variables
  - **var** variable_name
  - **var** varable_name=initial_value
- Dynamic vs. static typing

Identifiers

- Combination of letters, digits, underscores, and dollar signs
- Must begin with a character, dollar sign, or underscore: name, _name, $name
- Case-sensitive: name vs. Name vs. naMe
- Must not be a *keyword*

JavaScript keywords

break
case
catch
continue
debugger
default
delete
do
else
finally
for
function
if
in
instanceof
new
return
switch
this
throw
try
typeof
var
void
while
with

String

- var strString = "This is a string";
- var anotherString= 'But this is also a string';
- var stringNumbers=“123456”;
- var strQuote1='This is a "string" with a quote‘
- var strQuote2= “This is a ‘string’ with a quote”
- var emptyStr=“”;
- var specialStr=“This is a special \\string\\”;

Special Characters

Escape Sequences	Character Represented
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Tab
\’	Single quote
\”	Double quote
\\	Backslash
\xNN	NN is a hexadecimal number that identifies a character in the Latin-1 character set.

Convert to Strings

```
var num_value = 35.00;
var string_value = "This is a number:" + num_value;

var strValue = "4" + 3 + 1;
var strValueTwo = 4 + 3 + "1";

var firstResult = "35" - 3;

 var str=String(value);
```

Boolean Type

```
var isMarried = true;
var hasChildren = false;

var someValue = 0;
var someBool = Boolean(someValue) // evaluates to false

var strValue = "1";
var numValue = 0;
var boolValue = !!strValue; // converts "1" to a true
boolValue = !!numValue; // converts 0 to false
```

Number Data Type

- var negativeNumber = -1000;
- var zero = 0;
- var fourDigits = 2534;
- var someFloat = 0.3555
- var anotherNumber = 144.006;
- var negDecimal = -2.3;
- var lastNum = 19.5e-2
- var firstHex = -0xCCFF;

Null and *Undefined* Variables

- A null variable is one that you have defined and to which you have assigned *null as its value*
  - var nullString = null;
- If you have declared but not initialized the variable, it is considered undefined
  - var nullString;

```
var s = ""; //s is undefined?
```

NaN

- NaN: Not a Number

```
var nValue = 1.0;
if (nValue == 'one' ) // false, the second operand is NaN
```

- a null value is NaN.
- Test if a variable is a NaN

```
if (isNaN(sValue))  // if string cannot be implicitly converted
//into number, return true
```

Assignment Statement

- nValue = 35.00;
- nValue = nValue + 35.00;
- nValue = someFunction();
- var firstName = lastName = middleName = "";
- var nValue1,nValue2 = 3;
- var nValue1=3, nValue2=4, nValue3=5;
- nValue += 3.0;

Arithmetic Operators

- Binary arithmetic operators: +, -, *, /, %
- Unary arithmetic operators: -, ++, --

```
var v = 3.0;
var v2 = ++v;
var v3 = v++;
```

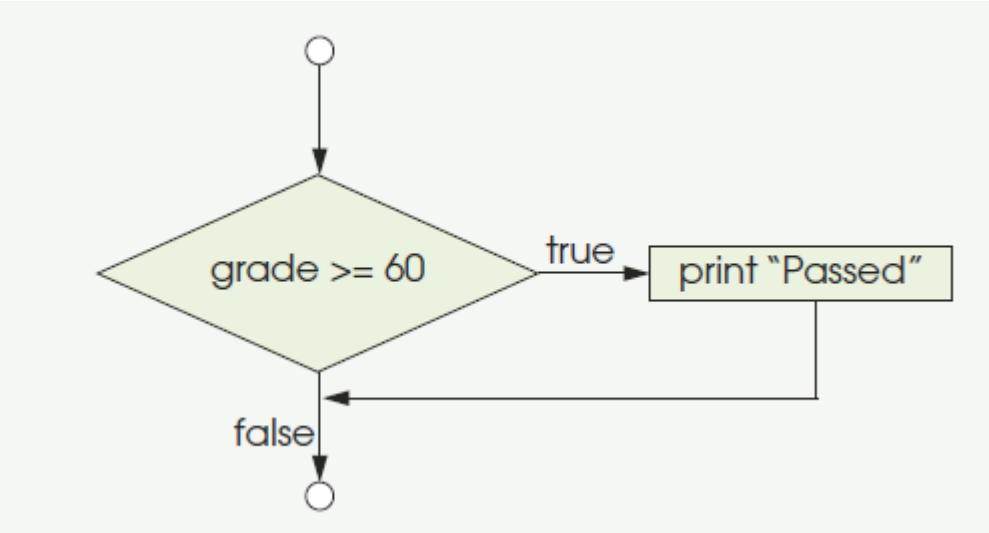
Conditional Statements

```
if (condition) {
        statements;
}
```

```
if (condition) {
        statements;
} else {
        statements;
}
```

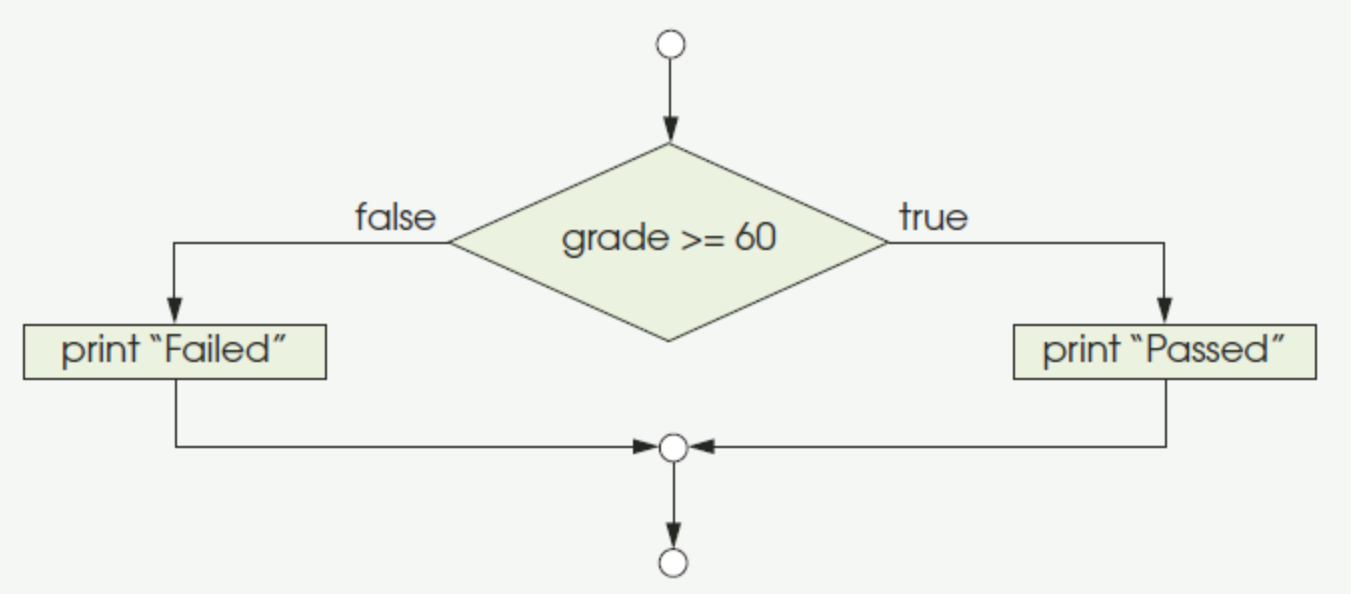
Conditional Statements

If student's grade is greater than or equal to 60
Print "Passed"




```
if ( studentGrade >= 60 )
        document.writeln( "Passed" );
```

Conditional Statements




```
if ( studentGrade >= 60 )
        document.writeln( "Passed" );
else
        document.writeln( "Failed" );
```

```
document.writeln( studentGrade >= 60 ? "Passed" : "Failed" );
```

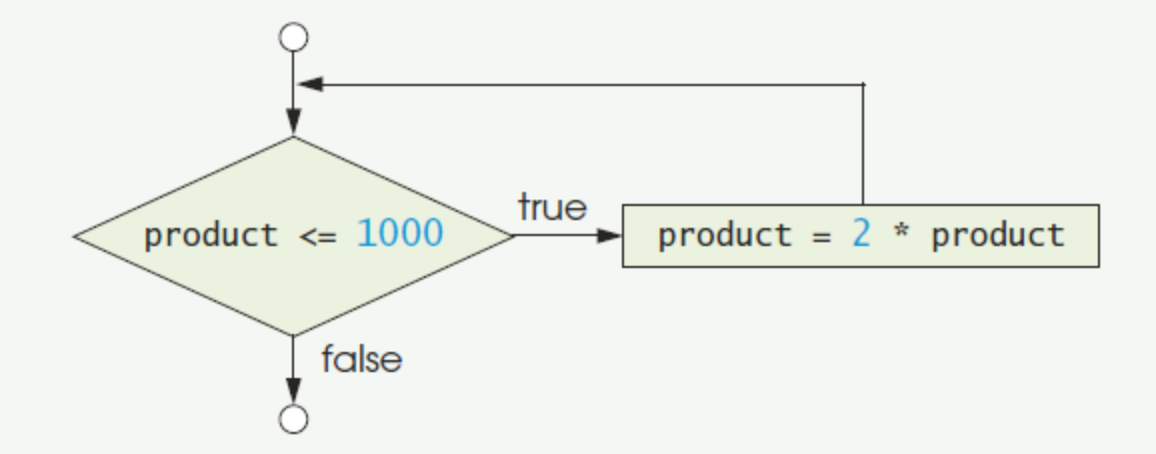
switch Statement

```
switch (expression) {
case firstlabel:
        statements;
        break;
case secondlabel:
        statements;
        break;
        ...
case lastlabel:
        statements;
        break;
default:
        statements;
}
```

```
var country="";
switch (code) {
case 84:
        country="Vietnam";
        break;
case 1:
        country="US";
        break;
case 44:
        country="UK";
        break;
default:
        country="Somewhere";
}
```

The while Loop

```
var product = 2;
while ( product <= 1000 )
        product = 2 * product;
```



The while Loop

```
var strValue = "";
var nValue = 1;
while (nValue <= 10) {
        strValue+=nValue;
        nValue++;
}
```

do...while Loop

```
do {
        strValue+=nValue;
        nValue++;
} while (nValue <= 10)
```

for Loops

```
for (initial value; condition; update)
{
        statements;
}
```

```
var s=0;
for (var i=10; i<100;i++){
        if(i%2==0)
                s=s+i;
}
```

JavaScript Objects

- Boolean
- Number
- String
- Date
- Math
- document
- window

Number Object

- Number.MAX_VALUE
  - The maximum number representation in JavaScript
- Number.MIN_VALUE
  - The smallest positive number representation in JavaScript
- Number.NaN
  - Represents Not-a-Number
- Number.NEGATIVE_INFINITY
  - Represents negative infinity
- Number.POSITIVE_INFINITY
  - Represents infinity

String Object

- Instantiate: var sObject = new String("Sample string");
- Some representative methods/properties
  - length
  - charAt, charCodeAt
  - indexOf
  - concat
  - substr
  - toLowerCase, toUpperCase

Methods of Strings

- ***charAt(index)****:* Returns a string containing the character at the specified *index.*
- ***concat(string)****:* Concatenates its argument to the end of the string that invokes the method.
- ***indexOf(substring, index)****:* Searches for the first occurrence of *substring* starting from position *index* in the string that invokes the method.
- ***substr(start, length)****:* Returns a string containing *length* characters starting from index *start* in the source string.
- ***substring(start, end)****:* Returns a string containing the characters from index *start up to* but not including index *end* in the source string.
- ***toLowerCase()***: Returns a string in which all uppercase letters are converted to lowercase letters.

Date Object

- Instantiate
  - var dtNow = new Date();
  - var dtMilliseconds = new Date(5999000920);
  - var newDate = new Date("March 12, 1980 12:20:25");
- Methods
  - getFullYear: The four-digit year
  - getHours: The hours component of the date/time
  - getMilliseconds: The milliseconds component of the date/time
  - getMinutes: The minutes component of the date/time
  - toString: Outputs the string in local time

Math Object

- Properties
  - PI: value of Pi
  - SQRT1_2: square root of 1/2
  - SQRT2: The square root of 2
- Methods
  - Math.sin(x)
  - Math.cos(x)
  - Math.tan(x)

document Object

- getElementById(*id)*
  - Returns the DOM node representing the HTML element whose id attribute matches id.
- write(*string)*
  - Writes the string to the HTML document as HTML code.
- writeln(*string)*
  - Writes the string to the HTML document as HTML code and adds a newline character at the end.
- cookie
  - A string containing the values of all the cookies stored on the user's computer for the current document

window Object

- Presents the window of Web browser
- Properties
  - window.document: This property contains the document object representing the document currently inside the window.
  - window.closed: set to true if the window is closed, and false if it is not.
- Methods
  - ***open(url, name, options)*** Creates a new window with the URL of the window set to url, the name set to name to refer to it in the script, and the visible features set by the string passed in as option.
  - ***prompt(prompt, default)*** Displays a dialog box asking the user for input.
  - ***close()*** Closes the current window and deletes its object from memory.
  - ***focus()*** This method gives focus to the window (i.e., puts the window in the foreground, on top of any other open browser windows).
  - ***blur()*** This method takes focus away from the window

Arrays

- var newArray = new Array('one','two');
- var newArray = ['one','two'];
- Multi-dimentional array
  - var threedPoints = new Array();
  - threedPoints[0] = new Array(1.2,3.33,2.0);
  - threedPoints[1] = new Array(5.3,5.5,5.5);
  - threedPoints[2] = new Array(6.4,2.2,1.9);

Functions

function *functionname (param1, param2, ..., paramn) {*
    function statements
}

function sayHi(toWhom) {
    alert("Hi " + toWhom);
}
…
sayHi(“Tom”);

Function Returns and Arguments

- Variables based on primitives, such as a string, boolean, or number, are passed to a function by value.
- Objects passed to a function, on the other hand, are passed by reference.
- A function may or may not return a value.
- Application encounters a **return** statement, it stops processing the function code

Anonymous Functions

```
var variable = new Function ("param1", "param2", ... , "paramn", "function body");
```

```
var sayHi = new Function("toWhom","alert('Hi ' + toWhom);");
sayHi("World!");
```

Write an anonymous function that finds the greatest number in three input numbers

```
var fcn=new Function("a", "b", "c","var g=a; if (g<b) g=b; if(g<c) g=c; return g");
s=fcn(7, 4, 6);
```

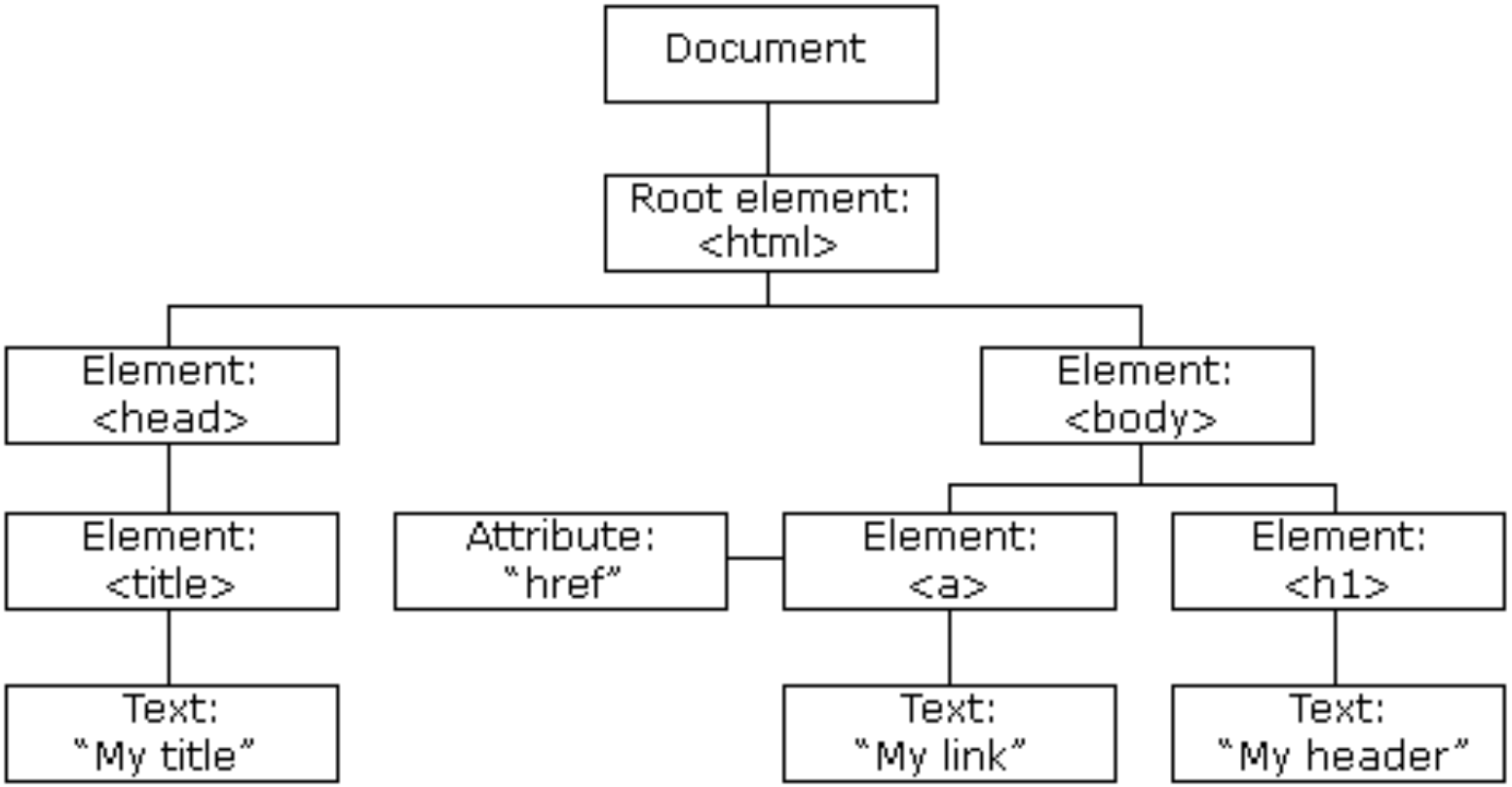
Function Literals

```
var func = function (params) {
        statements;
}

var func = function (x, y) {
        return x * y;
}
alert(func(3,3));
```

HTML DOM

```
<html>
   <head>
         <title>My title</title>
   </head>
   <body>
         <a href=“http://www.usth.edu.vn”>My link</a>
         <h1>My header</h1>
   </body>
</html>
```

HTML DOM


Document
Root element:
<html>
Element:
<head>
Element:
<body>
Element:
<title>
Attribute:
"href"
Element:
<a>
Element:
<h1>
Text:
"My title"
Text:
"My link"
Text:
"My header"

Manipulating DOM

- JavaScript creates dynamic HTML by manipulating DOM. Examples are:
  - Changing the content of HTML elements
  - Changing the style of HTML elements
  - Adding/deleting HTML elements
  - Reacting to HTML DOM events

Events

- Events allow scripts to respond to a user who is moving the mouse, entering form data or pressing keys, etc.
- Events and event handling help make web applications more responsive, dynamic and interactive
- Events must be registered for handling

Inline Registration

```
<html>
   <head>
    <title>My title</title>
    <script type="text/javascript">
          function changeHeader(){
document.getElementById("heading1").innerHTML =
               "Welcome to the Web App Class";
          }
   </script>
  </head>
   <body>
          <a href="http://www.usth.edu.vn">My link</a>
          <h1 id="heading1" onclick="changeHeader()">My
header</h1>
   </body>
</html>
```

Traditional Model Registration

```
<html>
   <head>
    <title>My title</title>
    <script type="text/javascript">
          function changeHeader(){
document.getElementById("heading1").innerHTML =
               "Welcome to the Web App Class";
          }
   </script>
  </head>
   <body onload="changeHeader()">
          <a href="http://www.usth.edu.vn">My link</a>
          <h1 id="heading1">My header</h1>
   </body>
</html>
```

Popular Events

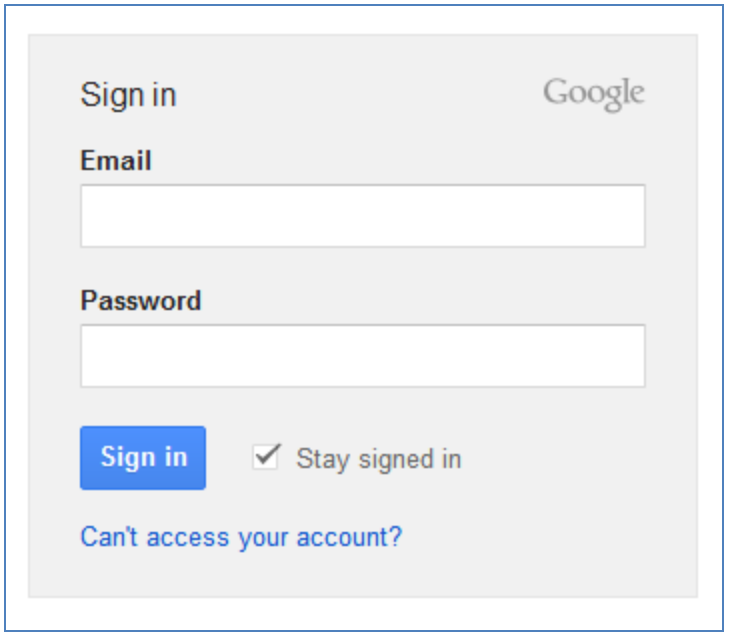
- **onclick**: fires when the user clicks an HTML element
- **onchange**: fires when an HTML element has been changed
- **onkeydown**: fires when the user pushes a keyboard key
- **onload**: fires when the browser has finished loading the page

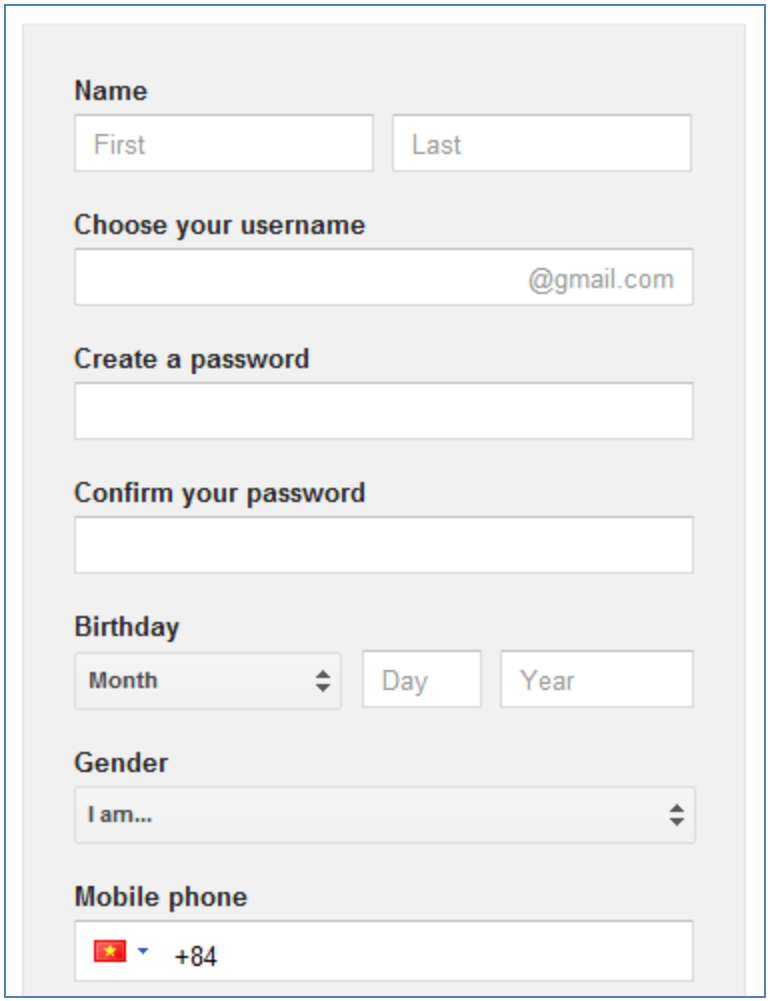
Popular Events

- **onmousemove**: fires repeatedly whenever the user moves the mouse over the web page
- **onmouseover**: fires when the user moves the mouse over an HTML element
- **onmouseout**: fires when the user moves the mouse away from an HTML element

Forms

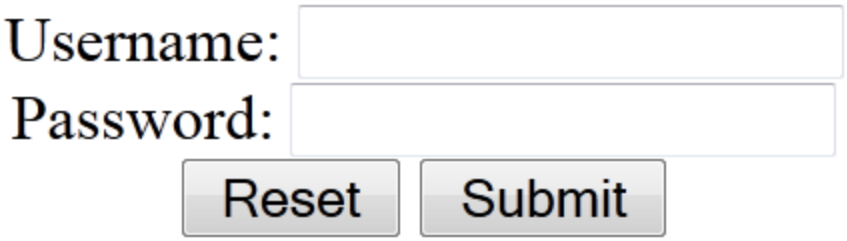

Google
UK
Google Search
I'm Feeling Lucky


Sign in
Google
Email
Password
Sign in
Stay signed in
Can't access your account?


Name
First
Last
Choose your username
@gmail.com
Create a password
Confirm your password
Birthday
Month
Day
Year
Gender
I am...
Mobile phone
+84

Forms

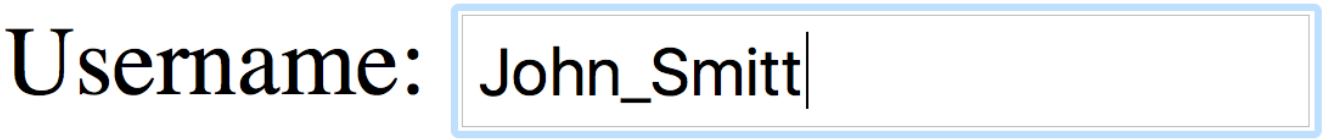
```
<form name="signin">
    Username: <input type="text" name="email"/> <br/>
    Password: <input type="password" name="passwd"/> <br/>

   <input type="reset" name="reset" value="Reset"/>
   <input type="submit" name="submit" value="Submit"/>
</form>
```



Form Elements

- Text field: a box which users can use to provide one-line text data

Username: <input type="text" name="username" />



Form Elements

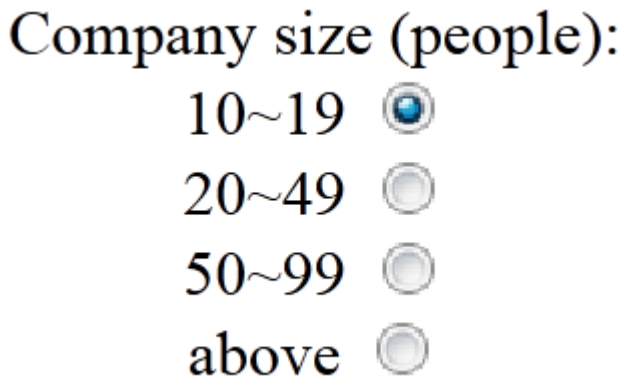
- Password field: defines a password field

```
Password: <input type="password" name="passwd"/>
```

Form Elements

- Radio button: provides users a means to select one from previously listed items

```
Company size (people): <br/>
  10~19 <input type="radio" name="csize" value="10_19" checked/> <br/>
  20~49 <input type="radio" name="csize" value="20_49"/> <br/>
  50~99 <input type="radio" name="csize" value="50_99"/> <br/>
  above <input type="radio" name="csize" value="above"/>
```



Form Elements

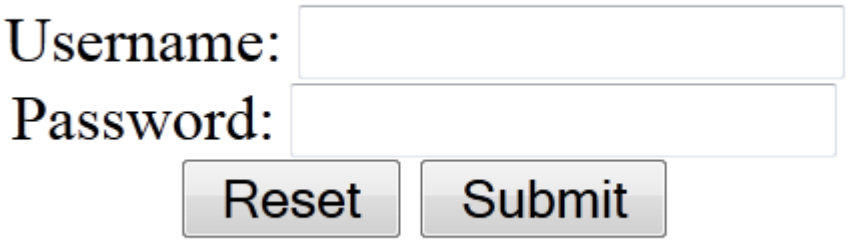
- Checkboxes: provides users a means to select one or more items from previously listed items

```
Languages:<br/>
    English <input type="checkbox" name="languages" value="English"/><br/>
    French <input type="checkbox" name="languages" value="French"/><br/>
  Japanese <input type="checkbox" name="languages" value="Japanese"/><br/>
  Chinese <input type="checkbox" name="languages" value="Chinese"/><br/>
```

Form Elements

Submit and Reset buttons:

```
<form name="signin">
  Username: <input type="text" name="email"/> <br/>
  Password: <input type="password" name="passwd"/> <br/>

  <input type="reset" name="reset" value="Reset"/>
  <input type="submit" name="submit" value="Submit"/>
</form>
```



Form Elements

Textarea: get multiple-line text input

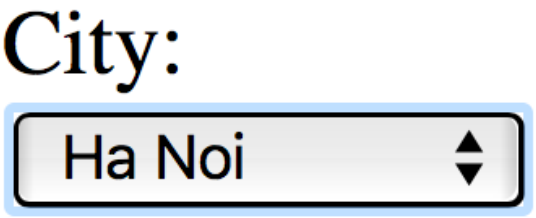
```
Comment: <br/>
<textarea name="comment" rows="10" cols="20"></textarea>
```

Comment:

Form Elements

A dropdown list allows users to select one item from a list of items.

```
City:<br/>
<select size="1">
    <option value=“HN">Ha Noi</option>
    <option value="H">Hue</option>
    <option value=“HCM”>Ho Chi Minh</option>
</select>
```



Form Processing: Getting values

- Form text field:

  value=document.formName.textfieldName.value;

- Form radio buttons:

  checked=document.formName.radioButtonName[index].checked;

  if (checked==true)

  value=document.formName.radioButtonName[index].value;

Form Processing: Validation

```
<html><head>
<script>
        function checkForm() {
           var unamef = document.forms["validateForm"]["uname"].value;
           if (unamef == null || unamef == "") {
              alert("Name must be filled out");
              return false;
           }
        }
</script>
</head>
<body>

<form name="validateForm"  action="check_form.asp" onsubmit="return checkForm()"
method="post">
      Username: <input type="text" name="uname"> <br />
       <input type="submit" name="submit" value="Submit">
</form>

</body></html>
```

JavaScript Frameworks

- A collection of Javascript functions for common JavaScript tasks like animations, DOM manipulation, and Ajax handling
- Examples:
  - Prototype
  - Foundation
  - jQuery