Javascript

Need for Scripting Languages

- Mathematical calculations
- Checking if a value matches another
- Finding a subsection of text
- Checking how long a piece of text
- Performing different actions
- Repeating an action a certain number of times

Define server side scripting and client side scripting

client side scripting

- Run scripts in a browser
- Processing on the end users computer
- code is transferred from the web server to the users
- **security risks** they may switch off
- Must be **enabled** on the client computer
- message usually pops up to alert the user

server side scripting

- Runs a scripting language is a web server
- user's request send to Web Server
- Result is client browser
- Web Server connect to Client

Examples of Client side scripting languages

- Javascript
- VB script
- AJAX
- jQuery
- ActionScript
- Dart

Illustrate how JavaScript is used in an HTML page

- <script> inside body tag
- <script> inside head tag
- External javascript file

<script> inside body tag

```
<html>
<body>

<script type="text/javascript">
document.write("My first JavaScript");
</script>

</body>
</html>
```

<script> inside head tag

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
document.write("My first
JavaScript");
</script>
</head>
<body>
</body>
</html>
```

External javascript file

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>
<script src="ht.js"></script>
</body>
</html>
```

External javascript file

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript"src="ht.js">
</script>
</head>
<body>

</body>
</html>
```

Programming elements in JavaScript

- Variable
- Operators
- Function
- Loop
- built in JavaScript objects

Assigning a Value to a Variable

```
var userName = “Bob Stewart”;
userName = “Robert Stewart”;
Num=9;
```

Lifetime of a Variable

- Local variables
- Global variables

Operators

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Logical operators
5. String operators

Arithmetic Operators

Symbol	Description	Example (x = 10)	Result
+	Addition	x+5	15
-	Subtraction	x-2	8
*	Multiplication	x*3	30
/	Division	x/2	15
%	Modulus (division remainder)	x%3	1
++	Increment (increments the variable by 1 — this technique is often used in counters)	x++	11
--	Decrement (decreases the variable by 1)	x--	9

Assignment operators

Symbol	Example Using Shorthand	Equivalent Without Shorthand
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Comparison Operators

Operator	Description	Example
==	Equal to	1==2 returns `false` 3==3 returns `true`
!=	Not equal to	1!=2 returns `true` 3!=3 returns `false`
>	Greater than	1>2 returns `false` 3>3 returns `false` 3>2 returns `true`
<	Less than	1<2 returns `true` 3<3 returns `false` 3<1 returns `false`
>=	Greater than or equal to	1>=2 returns `false` 3>=2 returns `true` 3>=3 returns `true`
<=	Less than or equal to	1<=2 returns `true` 3<=3 returns `true` 3<=4 returns `false`

Logical or Boolean Operators

Operator	Name	Description	Example (where x=1 and y=2)
&&	And	Allows you to check if both of two conditions are met	(x < 2 && y > 1) Returns `true` (because both conditions are met)
??	Or	Allows you to check if one of two conditions are met	(x < 2 ??y < 2) Returns `true` (because the first condition is met)
!	Not	Allows you to check if something is not the case	! (x > y) Returns `true` (because *x* is not more than *y*)

String Operator

- You can also add text to strings using the + operator

```
firstName = “Bob”
lastName = “Stewart”
name = firstName + lastName
```

Functions

- How to Define a Function
  - Define a name for it.
  - Arguments
  - Add statements.
- How to Call a Function
- Return Statement

```
<!DOCTYPE html>
<html>
<head>

<script>
var x = myFunction(4, 3);

function myFunction(a, b) {
    return a * b;
}
document.write("the result is="+x)
</script>
</head>
<body>

</body>
</html>
```

Conditional Statements

- if Statements
- if . . . else Statements
- A switch Statement

Looping

- While
- do . . . While
- For
- Infinite Loops and the break Statement

Events

Event	Purpose	Applies To
onload	Document has finished loading	<body> <frameset>
onunload	Document is unloaded, or removed	<body> <frameset>
onclick	Button on mouse has been clicked over the element.	Most elements
ondblclick	Button on mouse has been double-clicked over the element.	Most elements
onmousedown	Button on mouse has been depressed (but not released) over the element.	Most elements

Events

Event	Purpose	Applies To
onmouseup	Button on mouse has been released over the element.	Most elements
onmouseover	Button on mouse has been moved onto the element.	Most elements
onmousemove	Button on mouse has been moved while over the element.	Most elements
onmouseout	Button on mouse has been moved off the element	Most elements
onkeypress	A key is pressed and released over the element	Most elements

Events

Event	Purpose	Applies To
onkeydown	A key is held down over an element.	Most elements
onkeyup	A key is released over an element.	Most elements
onfocus	Element receives focus either by mouse clicking it, tabbing order giving focus to that element, or code giving focus to the element.	<a> <area> <button> <input> <label> <select> <textarea>
onblur	Element loses focus	<a> <area> <button> <input> <label> <select> <textarea>
onsubmit	A form is submitted.	<form>

Events

Event	Purpose	Applies To
onreset	A form is reset.	<form>
onselect	User selects some text in a text field.	<input> <textarea>
onchange	A control loses input focus and its value has been changed since gaining focus.	<input> <select> <textarea>

Built-in Objects

length	
big(), small()	strike()
bold(), italics()	sub()
charAt(index)	
fontsize(fontsize)	substr(start}, [length])
alert()	date()
toLowerCase(), toUpperCase()	getDay()
getMonth()	getFullYear()
getMinutes()	getSeconds()

Form Validation

- you have not entered a value or
- when you have entered the wrong kind of value
- Check whether the text you have entered
- choices you have made match some rules

➢ ***When to Validate***

➢ ***How to Validate***

When to Validate

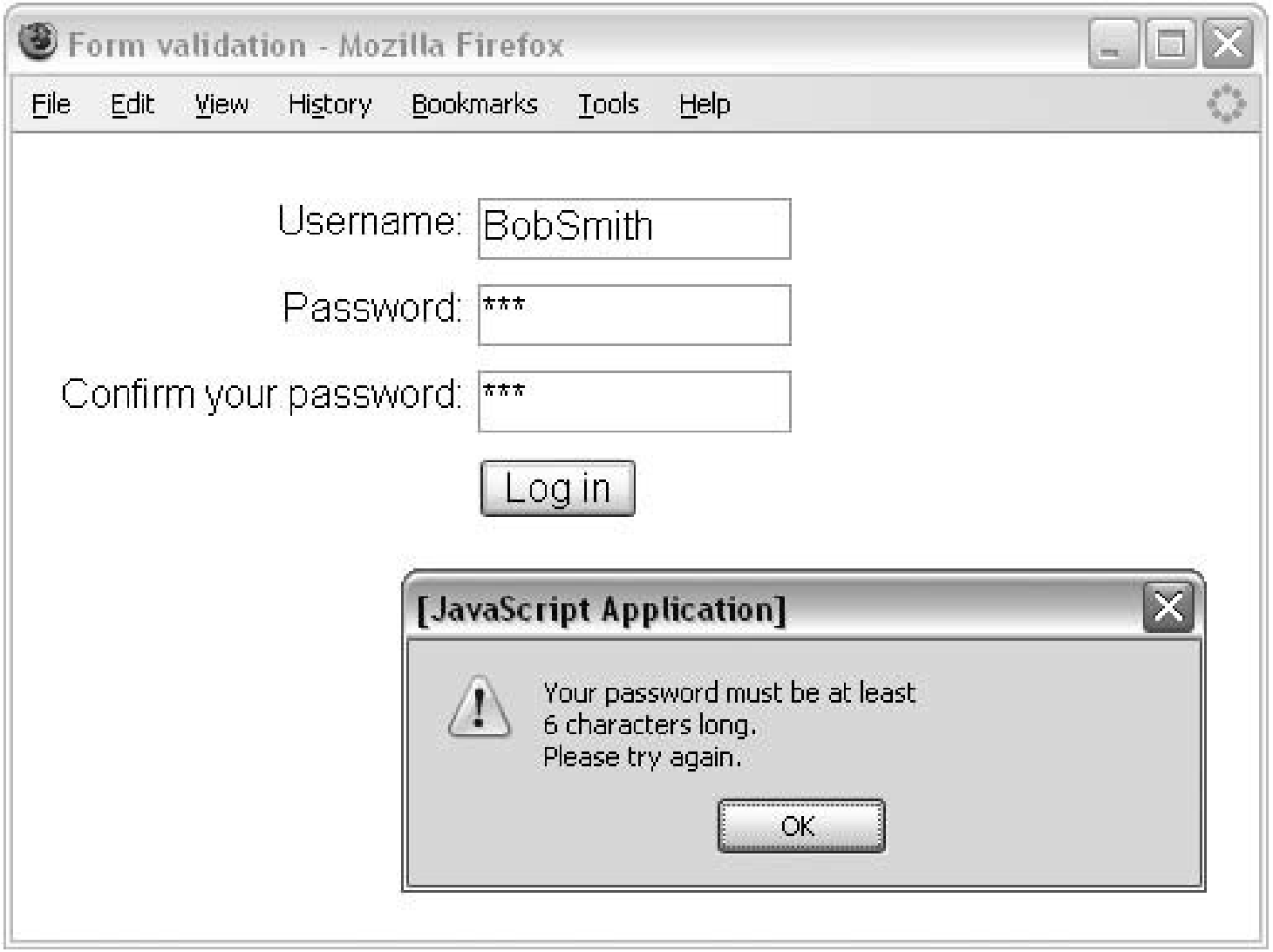
- Validation can happen in two places
  - Client Side
  - Server Side
- Client helps the user enter the correct data required for the job without the form being sent to the server
- Validation with database is done in server side

How to Validate

- **Checking Text Fields**

```
<form name=”frmRegister” method=”post”

action=”server.php”

onsubmit=”return validate(this);”>

<label for=”txtUsername”>Username:</label>

<input type=”text” name=”txtUserName” size=”12” />

<label for=”txtPassword”>Password: ></label>

<input type=”password” name=”txtPassword”

size=”12” />
```

```
<label for="txtPassword2">

Confirm your password:</label>

<input type="password"

name="txtPassword2" size="12" />

<input type="submit" value="Log in"

/></form>
```


Form validation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Username: BobSmith
Password: ***
Confirm your password: ***
Log in
[JavaScript Application]
Your password must be at least
6 characters long.
Please try again.
OK

<script type=”text/JavaScript”>

</script>

```
function validate(form) {

var returnValue = true;

var username =

frmRegister.txtUserName.value;

var password1 =

frmRegister.txtPassword.value;
```

```
if(username.length < 6) {
        returnValue = false;
        alert(“Your username must be at
least\n6
         characters long.\n
          Please try again.”);
         frmRegister.txtUserName.focus();
}
```

```
if (password1.length < 6) {
        returnValue = false;
        alert("Your password must be at
least\n6
        characters long.\n
       Please try again.");
frmRegister.txtPassword.value = "";
frmRegister.txtPassword2.value = "";
frmRegister.txtPassword.focus();
```

```
if (password1.value !=
password2.value) {
returnValue = false;
alter(“Your password entries did not
match.\nPlease try again.”);
frmRegister.txtPassword.value = “”;
frmRegister.txtPassword2.value = “”;
frm Register.txtPassword.focus();
```

```
return returnValue;
}
```

Select Box

```
function validate(form) {
var returnValue = true;
var selectedOption =
form.selCards.selectedIndex;
if (selectedOption==0)
{
returnValue = false
alert(“Please select a suit of cards.”);
}
return returnValue;
}
```

Radio Buttons

```
function validate(form) {
var radioChosen = false;
var radioButtons = form.radSuit;
for (var i=0; i<radioButtons.length; i++) {
if (radioButtons[i].checked)
{
radioChosen=true;
returnValue=true;
}
}
```

```
if (radioChosen == false) {
returnValue = false;
alert(“You did not select a suit of cards”);
}
return returnValue;
}
```

Checkboxes

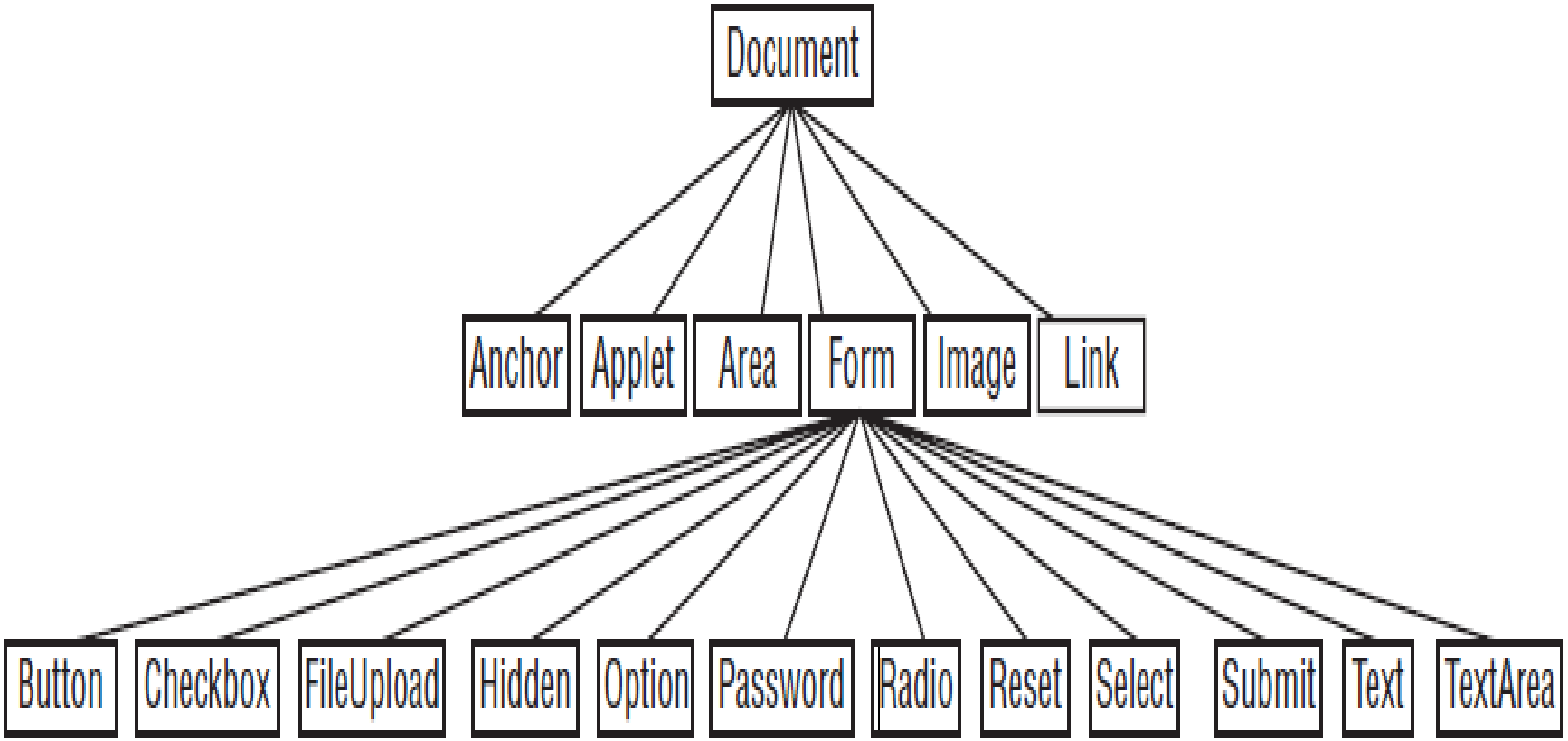
```
function countCheckboxes(field) {
var intCount = 0
for (var i = 0; i < field.length; i++)
{
if (field[i].checked)
intCount++; }
alert(“You selected “ + intCount +
“ checkbox(es)”);
}
```

DOM(Document Object Model)

- JavaScript by itself doesn’t do much more than allow you to perform calculations or work with basic strings.

- DOM defines the logical structure of documents and the way a document is

- Document object model specifies how you can retrieve values users have entered into a form.
- Retrieved these values, you can use JavaScript to ensure appropriate value


Document
Anchor
Applet
Area
Form
Image
Link
Button
Checkbox
FileUpload
Hidden
Option
Password
Radio
Reset
Select
Submit
Text
TextArea

- *DOM is not part of JavaScript*
- *It just explains how all programming languages should be able to access the properties of documents.*

document.frmLogin.pwdPassword.value

- The **document** comes first again as it is the top-level object.
- The name of the **form**, frmLogin.
- This is followed by the name of the **form control**, pwdPassword.
- The **value** of the password box

Objects, Methods, and Properties

- DOM Made up of several objects
- Each object can have *properties* and *methods*
- ***Property:***

Eg : ***<body bgcolor="green">***

- ***Method***

Dynamic documents with JavaScript

- DHTML can be changed while it is being displayed by a browser
- DHTML is a collection of technologies that allows dynamic changes to documents
- HTML document whose tag attributes, tag contents, or element style properties can be changed
- by user interaction or the occurrence of a browser event

- changes can be made with an embedded script that accesses the elements of the document as objects in the associated DOM structure.

DHTML

- POSITIONING ELEMENTS
- ELEMENT VISIBILITY
- MOVING ELEMENTS
- CHANGING COLORS AND FONTS

POSITIONING ELEMENTS

- Fixed Positioning
- Static Positioning
- Relative Positioning
- Absolute Positioning

Fixed Positioning

- Fix the position of an element to a particular spot
- Regardless of scrolling
- Use two values ***top*** and ***left*** along with the *position* property to move an HTML element

Static Positioning

- HTML elements are positioned static by default
- Static positioned elements are not affected by the top, bottom, left, and right properties.

Relative Positioning

- An element with **position: relative;** is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties
- element will cause it to be adjusted away from its normal position

Absolute Positioning

- An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.
- You can use two values *top* and *left* along with the *position* property
- move an HTML element anywhere in the HTML document

ELEMENT VISIBILITY

- Document elements can be specified to be visible or hidden with the value of their visibility property.
- two possible values for visibility are, quite naturally,

MOVING ELEMENTS

```
function moveIt (movee, newTop, newLeft) {
dom = document.getElementById
(movee).style;
dom.top=newTop + "px";
dom.left=newLeft + "px";
}
```

CHANGING COLORS AND FONTS

- <h1 style = "color:purple;"
  onmouseover="this.style.color='red';
  this.style.font='italic 20pt Jokerman';"
  onmouseout ="this.style.color='blue';
  this.style.font='bold 17pt Times';" >

s5 CSE </h1>

x-coordinate: <input type = "text" id = "leftCoord" size = "3" >

y -coordinate: <input type = "text" id = "topCoord" size = "3" >

- <input type="button" value="Move It" onclick = "moveIt('nit', document.getElementById('topCoord').value, document.getElementById('leftCoord').value)" >

```
<div id = "nit" style =
"position:absolute;top:115px;left:0;">
<img src = "Icon.png" alt = "SMPTC" /
>
</div>
```