### JavaScript Object Model

Working with images and forms



### images

- images[] is a property of document object representing array of window objects
- Each array element represents an object of type Image
- Properties:
  - align, alt, border, height, width, hspace, vspace, name, src
- Events
  - Onabort
  - Onerror
  - Onload
    - + HTMLElement Object Events



### **Example-Image rollover**

```
<html><head>
<title>Status</title>
<script>
function change(src)
document.images[0].src=src;
                        Since there is only one
</script>
                          image in the document
</head>
                          Or document.eye.src
<body>
<img src="close.bmp" name="eye"</pre>
  onMouseOver="change('open.bmp')"
  onMouseOut="change('close.bmp')">
</body>
</html>
```

### forms

- document object contains an array of form objects called forms.
- Form object corresponding to the <form> tag.

### Properties:

```
action, method, name, elements[], target + all the form elements listed in the hierarchy like button, checkbox etc.
```

### **Events:**

onSubmit, onReset

#### Methods:

reset() - Used to reset the form elements to their default values.

**submit()** - Submits the form as though the submit button were pressed by the user.

### Form elements

- Properties:
  - For all form elements → name, type, value (except select)
  - radio, checkbox: checked, defaultChecked
  - select: length, options[], selectedIndex
  - text, password, textarea:defaultValue
- Methods:
  - All form elements: focus(), blur()
  - For button, submit, reset: click()
- Events
  - For all form elements → onBlur, onFocus
  - For select, text, textarea→ onChange
  - For text, textarea→ onSelect
  - For button, submit, reset, radio, checkbox: onClick
  - For button: onMouseDown, onMouseUp,
  - For textArea: onKeyDown, onKeyUp,onKeyPress

## Example: text fields, radio button and checkbox

```
<html><head><title>Validate</title>
<script>
function check(){
 with (document.forms[0]) {
 if ((name.value=="") || (address.value=="")) {
  alert("Please ensure that all fields are
  filled up"); return false;
s=like[0].checked?like[0].value:like[1].value;
s="Thankyou,"+name.value+". You "+ s +" our
  site.\n";
s=s+"Your suggesstion (";
for (i=0;i<better.length;i++)</pre>
 if (better[i].checked) s=s+
     better[i].value+";"
```

```
s=s+" ) are recorded ";
alert(s);
                        When check() returns false the form does
                        not get submitted → the page does not go
return true;
                        to thank html
</script>
</head>
<body>
<form action="thank.html" onSubmit="return</pre>
check()">
Name: <input type=text name="name"><br><br>
Address: <input type=text
name="address"><br><br>
Do you like our site
<input type=radio name="like" value="like"</pre>
checked>Yes
<input type=radio name="like" value="don't</pre>
like">No<br><br>
```

```
Tell us how we can make this site better for
you:<br>
<input type=checkbox name="better"</pre>
value="Change the bg color">Change the bg
color
<br>
<input type=checkbox name="better"</pre>
value="Change the fg color">Change the fg
color
<br>
<input type=checkbox name="better"</pre>
value="Change the layout">Change the layout
<br>
<input type=checkbox name="better"</pre>
value="Include more services">Include more
services
<br><input type=submit></form> </body>
</html>
```

### **Example: working with select**

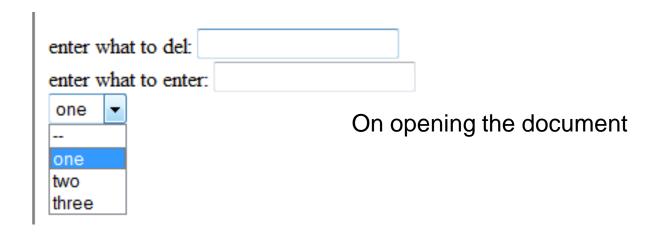
```
<html><head>
<title>Validate</title>
<script>
function check(){
 i=document.fl.choose.options.selectedIndex;
 if(i==0)
 alert("Correct");
else
 alert("Your choice, "+
 document.f1.choose.options[i].text +"- is
  incorrect");
</script>
</head>
<body>
```

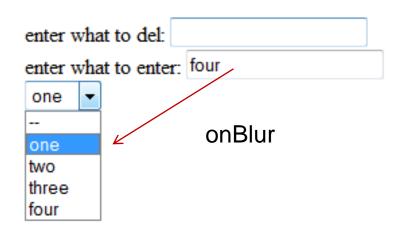


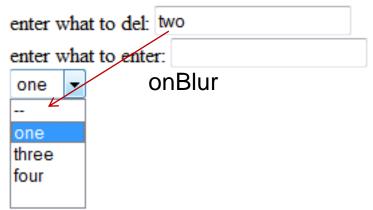
```
<form name=f1>
Which of the following is not true about
 JavaScript?
<select name="choose" onChange="check()">
<option>JavaScript is object-oriented
 language</option>
<option>JavaScript is loosely typed
 language
<option>JavaScript is used for client side
 validations
<option>JavaScript is platform
 independent
</select></form>
</body></html>
```



# Another way to work add and remove Option element









```
<html><head>
<script type="text/javascript">
function remSelOpt(inp1, sel1) {
 len1 = sel1.options.length;
 for (i=0;i<len1;i++) {
  if (sel1.options[i].value == inp1.value)
   sel1.options[i] = null;
    break;
function addSelOpt(inp1, sel1) {
len1 = sel1.options.length;
option0 = new Option(inp1.value,inp1.value);
sel1.options[len1]=option0;
</script>
</head>
```

```
<body><form>
enter what to del:
<input type=text name="removeOption"</pre>
onblur="remSelOpt(this,this.form.selectList)">
  <br>
enter what to enter:
<input type=text name="removeOption"</pre>
onblur="addSelOpt(this,this.form.selectList)">
<br>
<select id="selectList" name="selectList">
 <option value="--">--</option>
 <option value="one"</pre>
selected="selected">one</option>
 <option value="two">two</option>
 <option value="three">three</option>
</select>
</form>
</body></html>
```

### links

- Corresponds to the <a href="url"> tag of the document
- Document object contains an array of link objects → links
- Properties
  - hash The URL anchor part including the leading hash mark if one exists.
  - host The URL hostname and port.
  - hostname The URL hostname section.
  - href The URL.
  - pathname The URL pathname section.
  - port The URL port section.
  - protocol The URL protocol section including the colon after the protocol name.
  - search The URL query string section. This is the section after and including the question mark.
  - target The URL link's target name.
- Events: onClick ,onMouseOut , onMouseOver



### **Example**

```
<html><head>
<title>links</title>
<script>
function change() {
i=document.forms[0].choice.options.selectedIn
 dex;
document.links[0].href="http://www." +
 document.forms[0].choice.options[i].text
  +".com";
document.links[1].href= "http://"
  +document.forms[0].choice.options[i].value;
</script>
</head>
```



```
<body>
<form><select name="choice"</pre>
OnChange="change()">
<option value="#">--select--</option>
<option</pre>
value="www.mail.yahoo.com">yahoo</option>
<option</pre>
value="www.rediffmail.com">rediff</option>
</select>
</form>
<a href="">Home page</a><br>
<a href="">Mail Page</a>
</body></html>
```



### history

- The JavaScript history object is property of the window object.
- It maintains list of url/links visited during the current session.
- Properties
  - current The current document URL.
  - length The number of entries in the history object.
  - next The URL of the next document in the history object.
  - previous The URL of the last document in the history object.
- Methods
  - back()
  - forward()

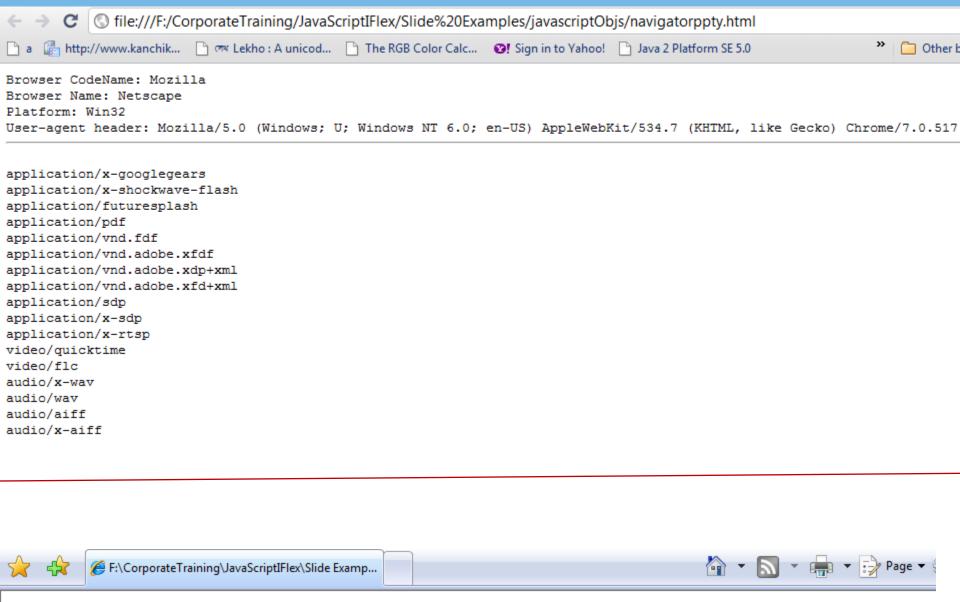
### navigator

- The JavaScript history object is property of the window object.
- It has the information about the client browser window.
- Properties:
  - appCodeName: the code name of the browser
  - appName: version of the browser
  - cookieEnabled: Boolean that indicates whether the browser has cookies enabled
  - mimeTypes: array of mimeTypes supported by the browser ( NS and Firefox only)
  - platform: OS(operating system)
  - userAgent: user agent string( the information that is added in the
     HTTP protocol for data transfer from browser to the server.

```
<html>
 <body>
  <
<script type="text/javascript">
  document.writeln("Browser CodeName: " +
  navigator.appCodeName);
  document.writeln("Browser Name: " +
  navigator.appName);
  document.writeln("Platform: " +
  navigator.platform);
  document.writeln("User-agent header: " +
  navigator.userAgent);
  document.writeln("<hr/>");
  browsername=navigator.appName;
```



```
if (browsername.indexOf("Netscape")!=-1)
 {browsername="NS"}
else
{if (browsername.indexOf("Microsoft")!=-1)
{browsername="MSIE"}
else {browsername="N/A"};
if (browsername!="MSIE") {
if (navigator.mimeTypes) {
  for(var p in navigator.mimeTypes) {
document.writeln(navigator.mimeTypes[p].type);
</script></body></html>
```



Browser CodeName: Mozilla

Browser Name: Microsoft Internet Explorer

Platform: Win32

User-agent header: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.5.30729;

