

# **Lecture 2.4**

## **Working with form Object, Form Validation**

**Course Instructor**  
**Engr. Madeha Mushtaq**

# Form Object

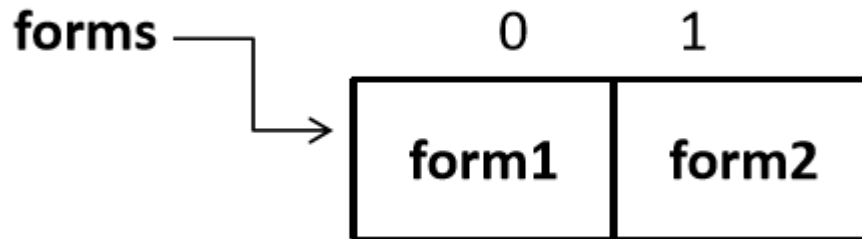
- The Form object represents an HTML form.
- The browser creates a 'forms array' which keeps the number of form objects in the HTML document.
- The first form object in the HTML file being held as array index [0], the second as index [1] and so on.

# Form Object

- The 'forms array' also holds information about each element used within <FORM> and </FORM> tags
- Elements array keeps information about form elements.

# Form Object

- `<body>`
- `<form name="form1">`
- `</form>`
- `<form name="form2">`
- `</form> </body>`



# Accessing form elements

- `<body>`
- `<form name="form1">`
- `<input type="text" name="name">`
- `<input type="text" name="email"> </form>`  
`</body>`
- `document.forms[0].name.value` or  
`document.form1.elements[0].value`  
`document.forms[0].email.value` or  
`document.form1.elements[1].value`

# Accessing form elements

```
<html>
<head>
  <title> Form Object </title>
</head>

<body>
  <form name = "my form" action = "mailto: mmushtaq@gwu.edu" method = "post">
    Name: <input type = "text" name = "name" id = "1">
    Password: <input type = "password" name = "password" id = "2">
    Email: <input type = "text" name = "email" id = "3">

    <p> Which languages do you know? (Select all that applies) </p>

    English <input type = "checkbox" name = "English" id = "4">
    Urdu <input type = "checkbox" name = "Urdu" id = "5">
    Pashto <input type = "checkbox" name = "Pashto" id = "6">
    <input type = "Submit" value = "Submit" name = "Submit" id = "7">
  </form>

  <script type = "text/javascript">
    var x = document.forms[0].elements[3].name;
    alert(x);
  </script>
</body>
</html>
```

# Accessing form elements

- What will be the output?

```
<script type = "text/javascript">  
var x = document.forms[0].length;  
alert(x);  
</script>
```

```
<script type = "text/javascript">  
var x = document.forms[0].elements[3].name;  
alert(x);  
</script>
```

# Setting form elements

- `<body> <form name="form1">`
- `<input type="text" name="name">`
- `<input type="text" name="email">`
- `</form> </body>`
- `document.forms[0].name.value = "ali" or  
document.form1.element[0].value = "ali"  
document.forms[0].email.value =  
"ali@gmail.com" or  
document.form1.elements[1].value =  
"ali@gmail.com"`



# Validating form elements

- One of the most common tasks performed using JavaScript.
- Validation can happen in two places, either in the browser using JavaScript or on the server.
- Onsubmit event handler, which triggers a validation function.
- The function returns true in order for the form to be sent. If the function returns false the user's form will not be sent.

# Validating form elements

```
<body>
<form name = "myform" action = "mailto: mmushtag@gwu.edu" method = "post" onsubmit = "return validation()">
  Name: <input type ="text" name = "name" id= "1">
  Password: <input type ="password" name = "password" id= "2">
  Email: <input type ="text" name = "email" id= "3">

  <p> Which languages do you know? (Select all that applies)</p>

  English <input type = "checkbox" name = "English" id = "4">
  Urdu <input type = "checkbox" name = "Urdu" id = "5">
  Pashto <input type = "checkbox" name = "Pashto" id = "6">
  <input type = "Submit" value = "Submit" name = "Submit" id = "7">
</form>

<script type = "text/javascript">
function validation(){
  if((document.myform.name.value == "")){
    alert("Please enter your name");
    return false;
  }
  else{
    alert("Thanks for your input")
    return true;
  }
}
</script>
</body>
</html>
```

# Validating form elements



The screenshot shows a web browser window with the title 'Form validation - Mozilla Firefox'. The browser's menu bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The main content area displays a login form with the following elements:

- A label 'Username:' followed by a text input field.
- A label 'Password:' followed by a text input field.
- A label 'Confirm your password:' followed by a text input field.
- A 'Log in' button located below the input fields.

- In this form we might want to check a few things:
  - That the username is of a minimum length
  - That the password is of a minimum length
  - That the two passwords match

# Checking Text Fields

## JS Validation Function

```
FormValidation_4.html x
1
2 <html>
3 <head>
4   <title>Form validation</title>
5   <script type="text/JavaScript">
6
7   function validate(form) {
8
9     var returnValue = true;
10
11     var username = form.txtUserName.value;
12     var password1 = form.txtPassword.value;
13     var password2 = form.txtPassword2.value;
14
15     if(username.length < 6) {
16       returnValue = false;
17       alert("Your username must be at least\n6 characters long.\nPlease try again.");
18       frmRegister.txtUserName.focus();
19     }
20
21     if (password1.length < 6) {
22       returnValue = false;
23       alert("Your password must be at least\n6 characters long.\nPlease try again.");
24       frmRegister.txtPassword.value = "";
25       frmRegister.txtPassword2.value = "";
26       frmRegister.txtPassword.focus();
27     }
28
29     if (password1 != password2) {
30       returnValue = false;
31       alert("Your password entries did not match.\nPlease try again.");
32       frmRegister.txtPassword.value = "";
33       frmRegister.txtPassword2.value = "";
34       frmRegister.txtPassword.focus();
35     }
36     return returnValue;
37   }
38
```

# Checking Text Fields

```
<link rel="stylesheet" type="text/css" href="forms.css" />
</head>
<body>

<form name="frmRegister" method="post" action="register.aspx" onsubmit="return validate(this);">

<div>
  <label for="txtUsername">Username:</label>
  <input type="text" name="txtUserName" id="txtUserName" size="12" />
</div>
<div>
  <label for="txtPassword">Password: </td></label>
  <input type="password" name="txtPassword" id="txtPassword" size="12" />
</div>
<div>
  <label for="txtPassword2">Confirm your password:</label>
  <input type="password" name="txtPassword2" id="txtPassword2" size="12" />
</div>
<div>
  <input type="submit" value="Log in" />
</div>
</form>
</body>
</html>
```

**HTML Code**

# Select Box Options

- Select boxes, also known as drop-down list boxes, save space on the screen, especially when there are lots of options.
- `selectedIndex` property of the select object represents the select box.
- By default, the value will be 0 for a standard select box.
- The `defaultSelected` property returns the default value if any set by the user.

# Select Box Options

```
<body>
<form name="frmCards" action="cards.asp" method="get"
      onsubmit="return validate(this)" >
  <select name="selCards" id="selCards">
    <option>Select a suit of cards</option>
    <option value="hearts">Hearts</option>
    <option value="diamonds">Diamonds</option>
    <option value="spades">Spades</option>
    <option value="clubs">Clubs</option>
  </select>
  <input type="submit" value="Send selection" />
</form>
</body>
```

# Select Box Options

```
<head>
  <title>Form validation</title>
  <script type="text/JavaScript" language="JavaScript">
    function validate(form) {
      var returnValue = true;
      var selectedOption = form.selCards.selectedIndex;
      if (selectedOption=="0")
      {
        alert("Please select a suit of cards.");
        returnValue = false;
      }

      return returnValue;
    }
  </script>
</head>
```



# Radio Buttons

```
<body>

<form name="frmCards" action="cards.aspx" method="post"
      onsubmit="return validate(this)" >
  <p>Please select a suit of cards.</p>
  <p><input type="radio" name="radSuit" value="hearts" /> Hearts </p>
  <p><input type="radio" name="radSuit" value="diamonds" /> Diamonds </p>
  <p><input type="radio" name="radSuit" value="spades" /> Spades </p>
  <p><input type="radio" name="radSuit" value="clubs" /> Clubs </p>
  <p><input type="submit" value="Submit choice" /></p>
</form>

</body>
```

# Radio Buttons

```
<head>
  <title>Form validation</title>
  <script type="text/JavaScript" language="JavaScript">

function validate(form) {
  var radioButtons = form.radSuit;
  radioChosen = false;
  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;
}
</script>
</head>
```

# Checkboxes

- Checkboxes allow a user to select zero, one, or more items from a set of choices.
- A group of checkboxes share the same name but they are not mutually exclusive as radio buttons are.

# 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)");  
}  
</script>  
</head>  
  
<body>  
<form name="frmCards" action="cards.aspx" method="post">  
    <p>Please select a suit of cards.</p>  
    <p><input type="checkbox" name="chkSuit" value="hearts" /> Hearts </p>  
    <p><input type="checkbox" name="chkSuit" value="diamonds" /> Diamonds </p>  
    <p><input type="checkbox" name="chkSuit" value="spades" /> Spades </p>  
    <p><input type="checkbox" name="chkSuit" value="clubs" /> Clubs </p>  
    <p><input type="button" value="Count checkboxes" onclick="countCheckboxes(frmCards.chkSuit)" /></p>  
</form>  
</body>  
</html>
```

# Preventing a Form Submission Until a Checkbox Has Been Selected

- If you want to ensure that a checkbox has been selected — for example, if you want a user to agree to certain terms and conditions.
- Add a function to the onsubmit event handler.
- The function checks whether the checkbox has been checked.
- If the function returns true the form will be submitted.
- If the function returns false, the user would be prompted to check the box.

# Preventing a Form Submission Until a Checkbox Has Been Selected

```
function checkCheckBox(myForm) {  
  if (myForm.agree.checked == false )  
  {  
    alert('You must agree to terms and conditions to continue');  
    return false;  
  } else  
    return true;  
}
```

# Preventing a Form Submission Until a Checkbox Has Been Selected

```
CheckBoxConfirmation.html x
1 <html>
2 <head>
3   <title>Form validation</title>
4 </head>
5
6 <body onload="document.frmAgree.btnSubmit.disabled=true">
7 <form name="frmAgree" action="test.aspx" method="post">
8
9   I understand that this software has no liability:
10  <input type="checkbox" value="0" name="chkAgree" id="chkAgree"
11    onclick="document.frmAgree.btnSubmit.disabled=false" />
12  <input type="submit" name="btnSubmit" value="Go to download" /><br />
13  <p>You will not be able to submit this form unless you agree to the
14    <a href="terms.html">terms and conditions</a> and check the terms and
15    conditions box.</p>
16 </form>
17 </body>
18 </html>
```

# References

- **Chapter 12** Beginning HTML, XHTML, CSS, and JavaScript, by Jon Duckett, Wiley Publishing; 2009, ISBN: 978-0-470-54070-1.