

Module 2 – Form object

Form Object

- The Form object represents an HTML form.
- For each `<form>` tag in an HTML document, a Form object is created.
- Forms are used to collect user input, and contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select menus, textarea, fieldset, legend, and label elements.
- Forms are used to pass data to a server.

Form Object Collections

- `elements[]` → Returns an array of all elements in a form

Form Object Properties

- **acceptCharset** → Sets or returns the value of the accept-charset attribute in a form.

Eg: Character encoding for Unicode (UTF-8) , Latin alphabet (ISO-8859-1)

- **action** → Sets or returns the value of the action attribute in a form
- **enctype** → Sets or returns the value of the enctype attribute in a form
- **length** → Returns the number of elements in a form
- **method** → Sets or returns the value of the method attribute in a form (get or post)
- **name** → Sets or returns the value of the name attribute in a form
- **target** → Sets or returns the value of the target attribute in a form (_blank, _top, _parent, _self, framename)

Form Object Methods

1. **reset()** → Resets a form

Eg : `document.getElementById("form1").reset();`

1. **submit()** → Submits a form

Form Object Events

1. **onreset** → The reset button is clicked
2. **onsubmit** → The submit button is clicked

```
<form onreset="alert('The form will be reset')">
```

```
Firstname: <input type="text" name="fname" value="Vijayan" ><br >
```

```
Lastname: <input type="text" name="lname" value="R" ><br ><br >
```

```
<input type="reset" value="Reset" >
```

```
</form>
```

The elements collection returns an array of all the elements in a form

```
<html> <body>
```

```
<form id="form1" action="">
```

```
  First name: <input type="text" name="fname" value="vijayan"> <br>
```

```
  Last name: <input type="text" name="lname" value="R"><br>
```

```
  <input type="button" value="Submit" name="submitbutton"> </form>
```

```
<script type="text/javascript">
```

```
  var x=document.getElementById("form1");
```

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

```
  {
```

```
    document.write(x.elements[i].name+": "+x.elements[i].value + "<br>");
```

```
  } </script> </body> </html>
```

```
<html> <head>
<script type="text/javascript">
function greeting()
{
alert("Welcome " + document.forms["form1"]["fname"].value + "!"); }
</script> </head> <body>
What is your name?<br>
<form name="form1" action="image.html" onsubmit="greeting()">
<input type="text" name="fname">
<input type="submit" value="Submit">
</form> </body> </html>
```

Form – reset () method

```
<html> <head>
<script type="text/javascript">
function formReset()
{ document.getElementById("frm1").reset(); }
</script> </head> <body>
<p>Enter some text in the fields below, then press the "Reset form"
  button to reset the form.</p>
<form id="frm1">
First name: <input type="text" name="fname" ><br >
Last name: <input type="text" name="lname" ><br ><br >
<input type="button" onclick="formReset()" value="Reset form" >
</form> </body> </html>
```

E-mail Validation

- The function below checks if the content has the general syntax of an email.
- This means that the input data must contain an @ sign and at least one dot (.). Also, the @ must not be the first character of the email address, and the last dot must be present after the @ sign, and minimum 2 characters before the end:

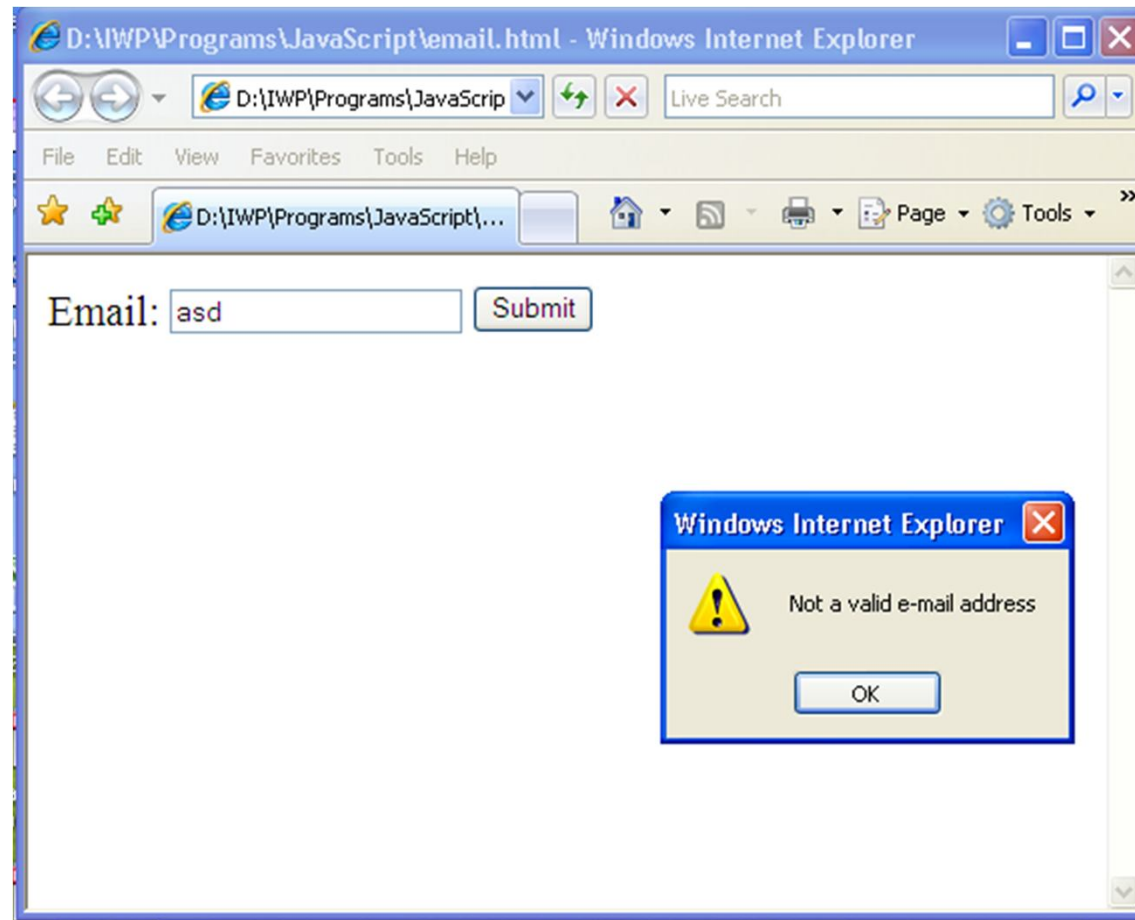
```
function validateForm() {  
  var x=document.forms["myForm"]["email"].value;  
  var atpos=x.indexOf("@");  
  var dotpos=x.lastIndexOf(".");  
  if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)  
    { alert("Not a valid e-mail address"); return false; } }
```

- The function above could be called when a form is submitted:

```
<form name="myForm" action="login.html" onsubmit="return  
validateForm();" method="post">
```

```
Email: <input type="text" name="email">
```

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

Registration Form

User id: Required and must be of length 5 to 12.

Password: Required and must be of length 7 to 12.

Name: Required and alphabates only.

Address: Optional.

Country: Required. Must select a country.

ZIP Code: Required. Must be numeric only.

Email: Required. Must be a valid email.

Sex: ☐ Male ☐ Female Required.

Language: ☒ English ☐ Non English Required.

About: Optional.

Submit

How would we set those validations

1. We will create JavaScript functions (one for each input field whose value is to be validated) which checks whether a value submitted by user passes the validation.
2. All those functions are called into another function.
3. It sets the focus to the input field until the user supplies a valid value.
4. When user does so, they may proceed and can supply value to the next available field.
5. The later JavaScript function created is called on onsubmit event of the form.

```

<html lang="en">
<head> <meta charset="utf8">
  <title>JavaScript Form Validation using a sample registration form</title>
<link rel='stylesheet' href='js-form-validation.css' type='text/css' />
  <script src="sample-registration-form validation.js">
</script> </head>
<body onload="document.registration.userid.focus();">
<h1>Registration Form</h1>
<p>Use tab keys to move from one input field to the next.</p>
<form name='registration' onSubmit="return formValidation();">
<ul> <li><label for="userid">User id:</label></li><li><input type="text" name="userid" size="12" />
</li>
<li><label for="passid">Password:</label></li><li><input type="password" name="passid" size="12" /></li>
<li><label for="username">Name:</label></li> <li><input type="text" name="username" size="50" /></li>
<li><label for="address">Address:</label></li> <li><input type="text" name="address" size="50" /></li>

```

```

<li><label for="country">Country:</label></li> <li><select name="country">
<option selected="" value="Default">(Please select a country)</option> <option v
alue="AF">Australia</option> <option value="AL">Canada</option> <option v
alue="DZ">India</option> <option value="AS">Russia</option> <option value
="AD">USA</option> </select></li>
<li><label for="zip">ZIP Code:</label></li> <li><input type="text" name="zip" /></
li>
<li><label for="email">Email:</label></li> <li><input type="text" name="email" siz
e="50" /></li>
<li><label id="gender">Gender:</label></li> <li><input type="radio" name="gend
er" value="Male" /><span>Male</span></li>
<li><input type="radio" name="gender" value="Female" /><span>Female</span>
</li> <li><label>Language:</label></li>
<li><input type="checkbox" name="en" value="en" checked /><span>English</sp
an></li> <li><input type="checkbox" name="nonen" value="noen" /><span>No
n English</span></li>
<li><label for="desc">About:</label></li> <li><textarea name="desc" id="desc"><
/textarea></li>
<li><input type="submit" name="submit" value="Submit" /></li> </ul> </form> </
body> </html>

```

js-form-validation.css

```
h1 { margin-left: 70px; }  
form li { list-style: none; margin-bottom: 5px; }  
form ul li label { float: left; clear: left; width: 100px; text-align: right; margin-right: 10px; font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 14px; }  
form ul li input, select, span { float: left; margin-bottom: 10px; }  
form textarea { float: left; width: 350px; height: 150px; }  
[type="submit"] { clear: left; margin: 20px 0 0 230px; font-size: 18px }  
p { margin-left: 70px; font-weight: bold; }
```

JavaScript function which is called on onSubmit

```
function formValidation()
{
var uid = document.registration.userid;
var passid = document.registration.passid;
var uname = document.registration.username;
var uadd = document.registration.address;
var ucountry = document.registration.country;
var uzip = document.registration.zip;
var uemail = document.registration.email;
var ugender = document.getElementsByName("gender");
if(userid_validation(uid,5,12)) {
if(passid_validation(passid,7,12)) {
if(allLetter(uname))
{
if(alphanumeric(uadd))
{
if(countryselect(ucountry))
{
if(allnumeric(uzip))
{
if(ValidateEmail(uemail))
{
if(validgender(ugender))
{
return true;
} } } } } } } }
return false; }
}
```

JavaScript function for validating userid

```
function userid_validation(uid,mx,my)
{
var uid_len = uid.value.length;
if (uid_len == 0 || uid_len >= my || uid_len < mx) {
alert("User Id should not be empty / length be between "+mx+" to "+
my);
uid.focus();
return false;
}
return true;
}
```


JavaScript function for validating password

```
function passid_validation(passid,mx,my)
{
var passid_len = passid.value.length;
if (passid_len == 0 || passid_len >= my || passid_len < mx)
{
alert("Password should not be empty / length be between "+mx+" to
    "+my);
passid.focus();
return false;
}
return true;
}
```

JavaScript code for validating user name

```
function allLetter(uname)
{
var letters = /^[A-Za-z]+$/;
if(uname.value.match(letters))
{
return true;
}
else {
alert('Username must have alphabet characters only');
uname.focus();
return false; } }
```

JavaScript code for validating user address

```
function alphanumeric(uadd)
{
var letters = /^[0-9a-zA-Z]+$/;
if(uadd.value.match(letters))
{
return true;
} else
{
alert('User address must have alphanumeric characters only');
uadd.focus();
return false; } }
```

JavaScript code for validating country

```
function countryselect(ucountry)
{
if(ucountry.value == "Default")
{
alert('Select your country from the list');
ucountry.focus();
return false;
}
else
{
return true; } }
```

JavaScript code for validating ZIP code

```
function allnumeric(uzip)
{
var numbers = /^[0-9]+$/;
if(uzip.value.match(numbers))
{
return true;
}
else
{
alert('ZIP code must have numeric characters only');
uzip.focus();
return false;
}
}
```

JavaScript code for validating email format

```
function ValidateEmail(uemail)
{
var mailformat = /^[a-zA-Z0-9_\. \-]+\@((([a-zA-Z0-9 \-])+\.)+([a-zA-Z0-9]{2,4})+)$/;
if(uemail.value.match(mailformat))
{
return true; }
else {
alert("You have entered an invalid email address!");
uemail.focus();
return false; } }
```

JavaScript code for validating gender

```
function validgender(ugender)
{
var isChecked = false;
  for (var i = 0; i < ugender.length; i++) {
    if (ugender[i].checked) {
      isChecked = true; // found one element checked
      break;
    }
  }
  if(isChecked==false)
  {
    alert("Select male/female");
  }
  return isChecked; }
```

Registration Form

Use tab keys to move from one input field to the next.

User id:

Password:

Name:

Address:

Country:


ZIP Code:


Email:

Sex: ☐ Male
☐ Female

Language: ☒ English
☐ Non English

About:

Message from webpage 

 User Id should not be empty / length be between 5 to 12

Registration Form

Use tab keys to move from one input field to the next.

User id:

Password:

Name:

Address:

Country:

ZIP Code:


Email:

Sex: ☐ Male
☐ Female

Language: ☒ English
☐ Non English

About:

Message from webpage

 Select your country from the list

OK

The Computer Gal Account Sign Up

First Name



Last Name

User Name
(4 to 15 letters)

Password
(up to 10 letters)

Email Address

Sign Up!

[Home](#)

Write validation functions and modify the onSubmit event Handler in the form code to validate the following form fields:

- Member number
 - Must be entered
 - Must be a number
- Password
 - Must be entered
 - Must be longer than 4 characters
- If an error occurs, use an alert box to display an error message for the first field in error and place the cursor in the first field in error.

Logon to Discount Delight's

Logon to our secure Web server to access Discount Delight's amazing deals!

The current date is: 11/14/2008

Member Number (from your membership card)
Password

1. Create a form that includes 2 textboxes (one for name, one for email address), a textarea, a dropdown box (<select> tag), a set of radio buttons, a set of checkboxes, and a button that submits the form input and a button that clears it. Format the form using a table. The ACTION attribute of the form tag should be to email the form results to your email address.
2. Add a JavaScript event handler to the form to determine whether the user input on the form is valid in the following ways. Feel free to do add additional criteria. Form validation should occur when the user clicks the button to submit the user input.
 - a) Validate that an entry has been made in each form element - textbox, textarea, dropdown box , radio buttons, checkboxes
 - b) If the user input for a particular form element fails the test in part 2, display a message in red text in the page (not a separate window) indicating the problem. When the user clicks the submit button, all the deficiencies in the form should be displayed.
 - c) If there are deficiencies in the form, the form input should not be emailed.
 - d) If the user changes the form input and clicks the submit button again, the validation described in part two above should occur again, with only the remaining deficiencies displayed.

Form with Name Validation

Please enter your name:

Please enter your birth year:

Where do you live?

What is your class year?

- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☐ 2013

What were your favorite things about Wellesley? (Choose 1 to 3):

- 0. ☐ close friends
- 1. ☐ awesome professors
- 2. ☐ exciting classes
- 3. ☐ great sports teams
- 4. ☐ clubs, orgs, and other extra-curriculars
- 5. ☐ proximity to Boston (Harvard, MIT ...)
- 6. ☐ women's college

Personal Information

Name:

First

Last

Gender:

☐ -M

☐ -F

Contact Information

Address:

No.

Street

City

State

Zip

Choose a state

Telephone:

validate form