WSA | Forward looking IT finishing school

Form Handling

JavaScript

www.webstackacademy.com

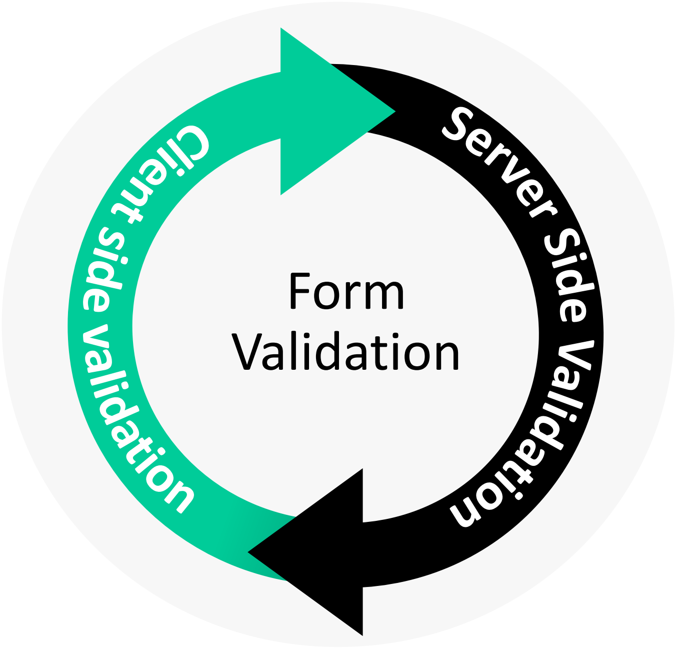
Introduction to Forms

(Basic form validation)

Form Processing

- Forms are one of the key interaction elements, where the user submits information into the website
- Upon submission of data into the form (in form of various fields) validation of the form happens at two levels as follows

Input validation in the browser itself by scripting languages like JavaScript



User input is sent via network and validated at the server side using server side languages like PHP

Form Validation – What?

- Form validation is the process of making sure that data supplied by the user using a form, meets the criteria set for collecting data from the user. Some examples are:
  - *Has the user left required field empty ?*
  - *Has the user entered valid email address ?*
  - *Has the user entered text in numeric field ?*
  - *Has the user entered number in correct range ?*
  - *Has the user entered valid date ?*

Primitive version of form – Text boxes

- We were using some primitive version of form in form of text boxes
- Combining input boxes with button events can be used to implement forms

```
<script>
function validate() {
    var numValue,formMessage;
    numValue = document.getElementById("num").value;
    if (isNaN(numValue) || (numValue < 5 || numValue > 20)) {
        formMessage = “Not valid!";
    } else {
        formMessage = "OK";
    }
    document.getElementById("ex").innerHTML = formMessage;
}
</script>
```

Primitive version of form – Text boxes

```
<body>


<p>Please input a number between 5 and 20:</p>
<input id="num">
<button type="button" onclick="validate()">Submit</button>
<p id="ex"></p>


</body>
```

Accessing Data from forms

Elements of a form can be accessed by `document.formName.formElement`

Attribute	Description
`formName`	Name attribute of the form element
`formElement`	Value of the name attribute

```
<form name ="form1" onsubmit="return validate(this);">

        Username: <input type="text" name="Username"><br>
        Password:  <input type="password" name="Password"><br>

</form>
```

Accessing Data from forms

```
<script>
function validateForm(form){
   var Username = document.form1.Username.value;
   var Password = document.form1.Password.value;
    if (Username.length == 0) {
        alert("You must enter a username.");
        return false;
    }
    else if(Password.length < 8){
        alert("Password must be at least 6 characters long.");
        return false;
    }
}
</script>
```

Form validation – Text fields

Property	Description
maxLength	Maximum number of character that can be entered in the text field
Placeholder	Place default text (ex: some recommendation to the user)
size	Specify the field width
disabled	Specify whether or not the field is disabled

```
<input type =”text” name=”tx1” maxLength=”30”>

<input type =”text” name=”tx2” size=”10”>

<input type=”text” name=”tx3” placeholder=”Default Text”>
```

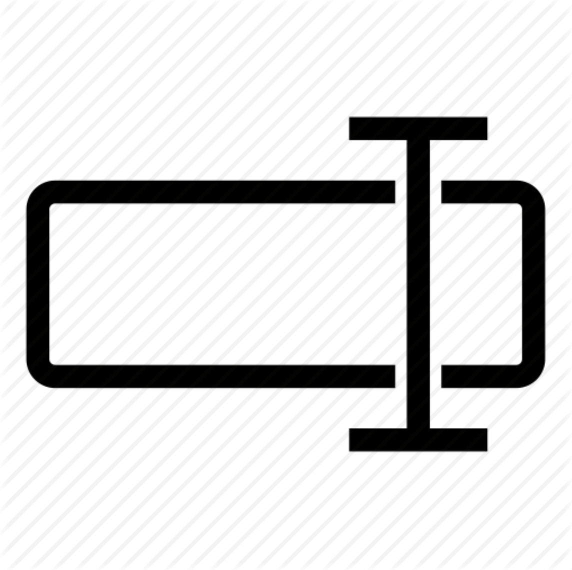
Form validation – Semantic text fields

Property	**Description**
`<input type=”email”>`	Text field with no line breaks that should hold an email address
`<input type=”number”>`	Single-Line text field for number input
`<input type=”url”>`	Text field with no line breaks that should hold an url.
`<input type=“password”>`	Text field is a password, hence don’t display on the screen
`<input type=“submit”>`	Submit button. Trigger the function that is attached with the event `onsubmit`
`<input type=“reset”>`	Reset input fields and take input again

Form validation – Text area

Text Area - Syntax

```
<textarea name=”field name” id =”id” rows =”nrows” cols =”ncolumns”>
            Default text for the field
</textarea>
```

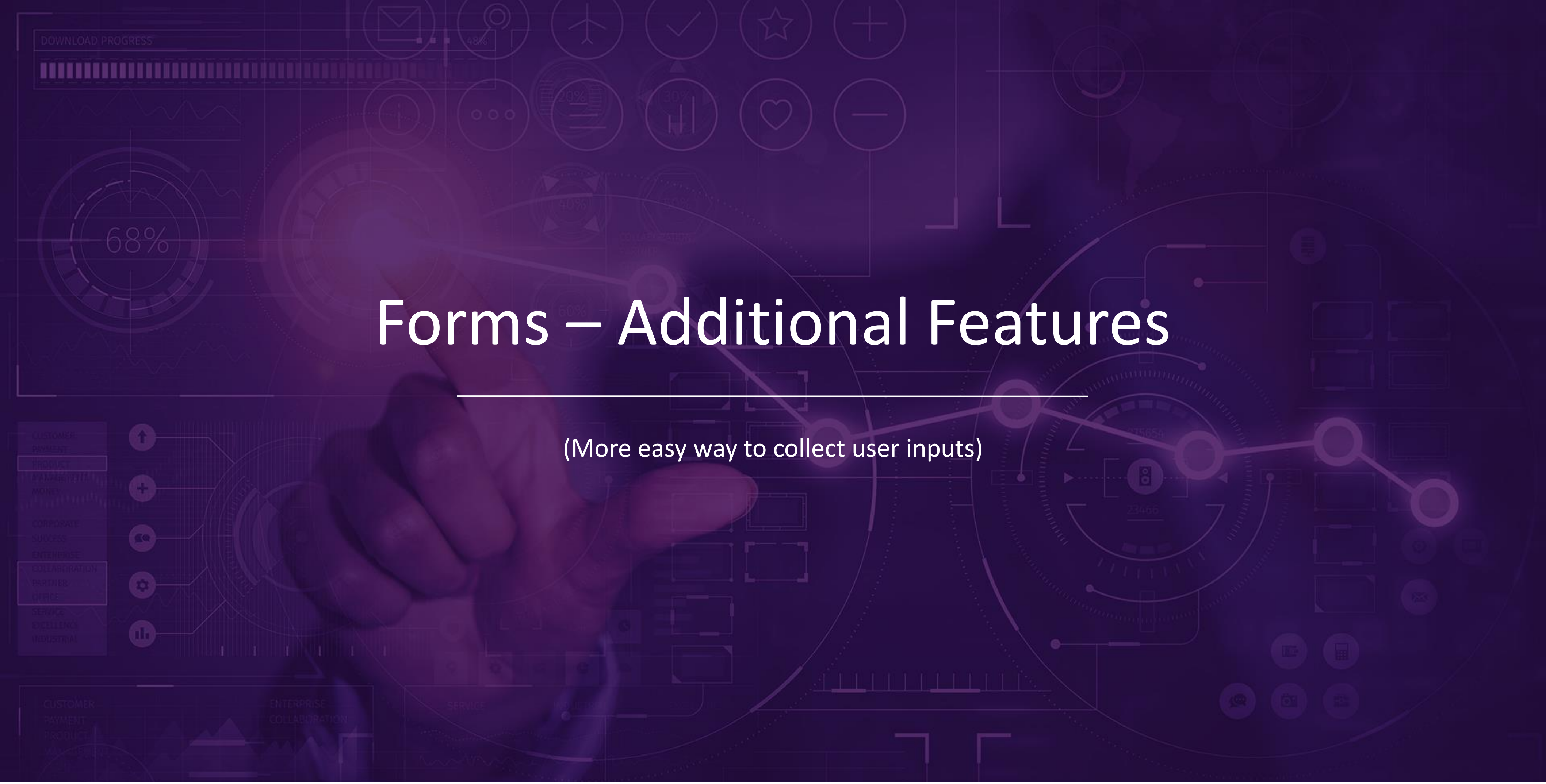


Exercise

- Enhance the customer enquiry form with the following:
  - ○ Add text a text box to enter their enquiry text
  - ○ Add additional validation points as follows:
    - ✓ Text field should not be left blank
    - ✓ Better email ID validation
    - ✓ Check there are 10 digits entered for the mobile number

Some useful tips:

- **field.value.match(regexp)** will match the regular expression against the given value



Forms – Additional Features

(More easy way to collect user inputs)

Additional form facilities & validation

Radio button provides option to choose an item from multiple:

```
<input type="radio" id=“male">Male<br>
<input type="radio" id="female">Female<br>
```

Validation code:

```
if ((form.male.checked == false) && (form.female.checked == false) )
   {
        alert ( "Please choose your Gender: Male or Female" );
        return false;
   }
```

Additional form facilities & validation

Checkbox validation (ex: Terms & Condition):

```
<input type =”checkbox” id=”license”>
```

Validation code:

```
if ( document.form1.license.checked == false )
{
      alert ( "Please check the Terms & Conditions box." );
      return false;
}
```

Additional form facilities

Select from given set of values

```
<select id="mySelect">
  <option value="apple">Apple</option>
  <option value="orange">Orange</option>
</select>
```

Validation code:

```
if ( document.form1.city.selectedIndex == 0 )
    {
        alert ( "Please select your City." );
        return false;
    }
```

Exercise

- Write a JavaScript program to create a Pizza order form:
  - ✓ Create radio button to select pizza (Simply Veg / Veg Supreme / Veg Delight)
  - ✓ Create radio button to select pizza size (Personal Pan / Medium / Large)
  - ✓ Create radio button to select crust (Thin / Normal / Special)
  - ✓ Create a drop down list to select payment option (COD / Credit Card / PayTM / Debit Card)
  - ✓ Create check box to agree terms & conditions
  - ✓ Facility to enter discount code with limited size of the field (Have some code like PIZDIS090)
  - ✓ Take customer details (Name, Email ID and Phone number)
  - ✓ Display complete details entered by the user
  - ✓ Add appropriate validation checks as necessary

Validation APIs

(Functions provided by JavaScript for form validation)

JavaScript – Validation APIs

Additional validation APIs provided by JavaScript

Property	Description
`checkValidity()`	Returns a Boolean indicating whether or not the current value of the element is valid. It contains automatically populated error message in the field named “validationMessage”
`rangeOverflow`	Returns true if the element's value is larger than the element's max value.
`rangeUnderflow`	Returns true if the element's value is less than the element's min value.

```
<input id="id1" type="number" min="100" max="300"
<button onclick="myFunction()">OK</button>
```

Check Validity example

```
<script>
function rangeFunction() {
var myInput = document.getElementById("id1“);
if (!myInput.checkValidity())
{
        msg = myInput.validationMessage;
}
else
{
        msg = "Input OK";
}
    document.getElementById("demo").innerHTML = msg;
}
</script>
```

Range Overflow example

```
<script>
function rangeFunction() {
var msg = "";
if (document.getElementById("id1").validity.rangeOverflow)
{
        msg = "Value too large";
}
else
{
        msg = "Input OK";
}
    document.getElementById("demo").innerHTML = msg;
}
</script>
```

Thank you

WebStack Academy

#83, Farah Towers,
1st Floor, MG Road,
Bangalore – 560001

M: +91-809 555 7332
E: training@webstackacademy.com

WSA in Social Media: