华中师范大学伍伦贡联合研究院
Central China Normal University Wollongong Joint Institute

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

CSIT884
Web Development

Lecture 11B – Form Validation

Lecture slides are mainly taken from UOW

Objectives

- Built-in form validation
- Use JavaScript to perform form validation

Form validation

What is form/data validation?

- Server-side validation
- Client-side validation
  - built-in form validation
  - using JavaScript

Built-in form validation

- uses HTML5 form validation features
- generally doesn't require much JavaScript
- has better performance, but it is not as customizable as JavaScript validation

- HTML constraint validation is based on:
  - HTML Input Attributes (required, minlength/maxlength, min/max, type, pattern)
  - CSS Pseudo Selectors (:disabled, :invalid, :optional, :required, :valid)
  - DOM Properties and Methods (constraint validation API)

Built-in form validation – Example 1

```
<form . . . >
<h3>Insert data</h3> <i style="color:red">(*required)</i>
<p>
<label>Select colour </label><br />
<input type="radio" required name="chColour" value="black">
<label>Black</label>
<input type="radio" required name="chColour" value="white">
<label>White</label>
</p>
<p>
<label>Choose a 2-digit number</label>
<input type="number" min="10" max="99" step="1">
</p>
<p>
<label>Enter your e-mail address</label>
<input type="email" required name="email">
</p>
<p>
<label>Your message</label>
<textarea name="msg" maxlength="50" rows="3"></textarea>
</p>
<button>Submit</button>
</form>
```

Built-in form validation – Example 1

```
<style>
  input:invalid {
    box-shadow: 0 0 5px 1px red;
  }
  input:valid {
    box-shadow: 0 0 2px 1px green;
  }
  input:focus:invalid {
    box-shadow: none;
  }
</style>
```

Using JavaScript for form validation

- Using JavaScript the form's data can be validated on the client's computer before sending it to the web server
  - Basic Validation – make sure all the mandatory fields are filled in
  - Data Format Validation – if entered data is in correct form and value

Using JavaScript for form validation

```
<form action="myService" method="get" onSubmit="return validateForm()">
... your form goes here ...
</form>


<script>
function validateForm() {
  if (... something wrong ...) {
    return false;
  }
  return true;
}
</script>
```

Use form attribute **onSubmit** to check input before form submission

When function returns **false**, form will not be submitted

Form validation – Example 2

Example 2: Alert the user if the email field is empty

```
<form action="myService" method="get" onSubmit="return validateForm()">

  <h3>Insert email
  <span style="font-style: italic; color:red">(*required)</span>
  <h3>

  <p>
  <label>Enter your e-mail address</label>
  <input type="text" id="email" name="email">
  </p>

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

Form validation – Example 2

Example 2: Alert the user if the email field is empty

```
<script>
function validateForm() {
  var email = document.getElementById("email").value;
  if (email == null || email == "") {
    alert("Email must be filled out");
    return false;
  }
  return true;
}
</script>
```

Form validation – Example 3

Example 3: What if user enters only whitespaces?

```
<script>
function validateForm() {
  var email = document.getElementById("email").value;
  if (email == null || email.trim() == "") {
    alert("Email must be filled out");
    return false;
  }
  return true;
}
</script>
```

The `trim()` method removes whitespace from both sides of a string

Form validation – Example 4

Example 4: If user didn't fill out the email, we want to display an error message. We use a `span` element as a placeholder for the error message.

```
<form action="myService" method="get" onSubmit="return validateForm()">
  <h3>Insert email
  <span style="font-style: italic; color:red">(*required)</span>
  <h3>

  <p>
  <label>Enter your e-mail address</label>
  <input type="text" id="email" name="email">
  <span id="emailError" style="color:red"></span>
  </p>

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

Form validation – Example 4

Example 4: If user didn't fill out the email, we want to display an error message. We use a `span` element as a placeholder for the error message.

```
<script>
  function validateForm() {
    var email = document.getElementById("email").value;
    if (email == null || email == "") {
      document.getElementById("emailError").innerHTML =
                                        "Email must be filled out";
      return false;
    }
    return true;
  }
</script>
```

Form validation – Example 5

Example 5: We want to have two input fields. One for email and another one for email confirmation. User has to fill in the same email for both input fields.

```
<form action="myService" method="get" onSubmit="return validateForm()">
  <h3>Insert email
   <span style="font-style: italic; color:red">(*required)</span>
  </h3>
  <p>
   <label>Enter your e-mail address</label>
   <input type="text" id="email" name="email">
   <span id="emailError" ></span>
  </p>
  <p>
   <label>Confirm your e-mail address</label>
   <input type="text" id="email2" name="email2">
   <span id="emailError2" ></span>
  </p>
  <input type="submit" value="Submit">
</form>
```

Form validation – Example 5

Example 5: We want to have two input fields. One for email and another one for email confirmation. User has to fill in the same email for both input fields.

```
<style>
#emailError{
  color: red;
}
#emailError2{
  color: red;
}
</style>
```

Or (by using class)

```
<style>
.errorMessage{
  color: red;
}
</style>

<span id="emailError" class="errorMessage"></span>
<span id="emailError2" class="errorMessage"></span>
```

Form validation – Example 5

Example 5: We want to have two input fields. One for email and another one for email confirmation. User has to fill in the same email for both input fields.

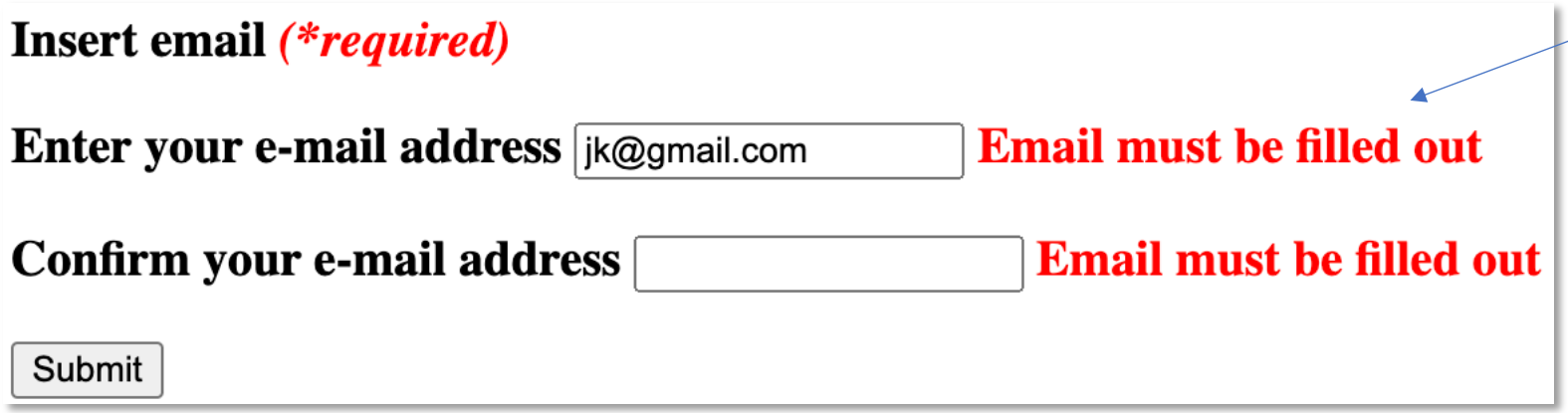
```
<script>
function validateForm() {
  var email = document.getElementById("email").value;
  if (email == null || email.trim() == ""){
    document.getElementById("emailError").innerHTML = "Email must be filled out";
    return false;
  }
  var email2 = document.getElementById("email2").value;
  if (email2 == null || email2.trim() == ""){
    document.getElementById("emailError2").innerHTML = "Email must be filled out";
    return false;
  }
  if(email.trim() != email2.trim()){
    document.getElementById("emailError2").innerHTML = "Email does not match";
    return false;
  }
  return true;
}
</script>
```

Form validation

Now suppose that user didn't fill out the email and click Submit, there will be a red error message next to the first input field.

Suppose that user fixed the error by filling out the first email, but leaving the second email field blank.

When the user clicks Submit, we will see that the error message next to the first input field still shows.



We don't want this

Form validation – Example 6

We should fix the JavaScript code

```
function validateForm() {
  var email = document.getElementById("email").value;
  if (email == null || email.trim() == ""){
    document.getElementById("emailError").innerHTML = "Email must be filled out";
    return false;
  }else{
    document.getElementById("emailError").innerHTML = "";
    document.getElementById("email").value = email.trim();
  }
...
```

remove all whitespaces in the input field before submit

Form validation – Example 6

We should fix the JavaScript code

```
function validateForm() {
  ...
  var email2 = document.getElementById("email2").value;
  if (email2 == null || email2.trim() == ""){
    document.getElementById("emailError").innerHTML = "Email must be filled out";
    return false;
  }else{
    document.getElementById("emailError2").innerHTML = "";
    document.getElementById("email2").value = email2.trim();
  }
...
```

remove all whitespaces in the input field before submit

Form validation – Example 6

We should fix the JavaScript code

```
function validateForm() {
  ...
  if(email.trim() != email2.trim()){
    document.getElementById("emailError2").innerHTML = "Email does not match";
    return false;
  }else{
    document.getElementById("emailError2").innerHTML = "";
  }
  return true;
}
```

Form validation – Example 7

Validate that the user is submitting the form. For example, ask user a simple math problem, only submit the form if user answers correctly

```
function validateForm() {
  ...
  var x = Math.floor(Math.random() * 10) + 1;
  var y = Math.floor(Math.random() * 10) + 1;
  var correctAnswer = x + y;
  var answer = prompt("What is " + x + " + " + y + " ?");
  if(answer == null || answer != correctAnswer){
    return false;
  }
  ...
}
```

Generate random question

References

- Robert W. Sebesta, Programming the World Wide Web, 8th edition, Pearson, 2015.
- http://www.w3schools.com/js
- http://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://www.w3schools.com/js/js_validation.asp
- https://developer.mozilla.org/enUS/docs/Learn/Forms/Form_validation