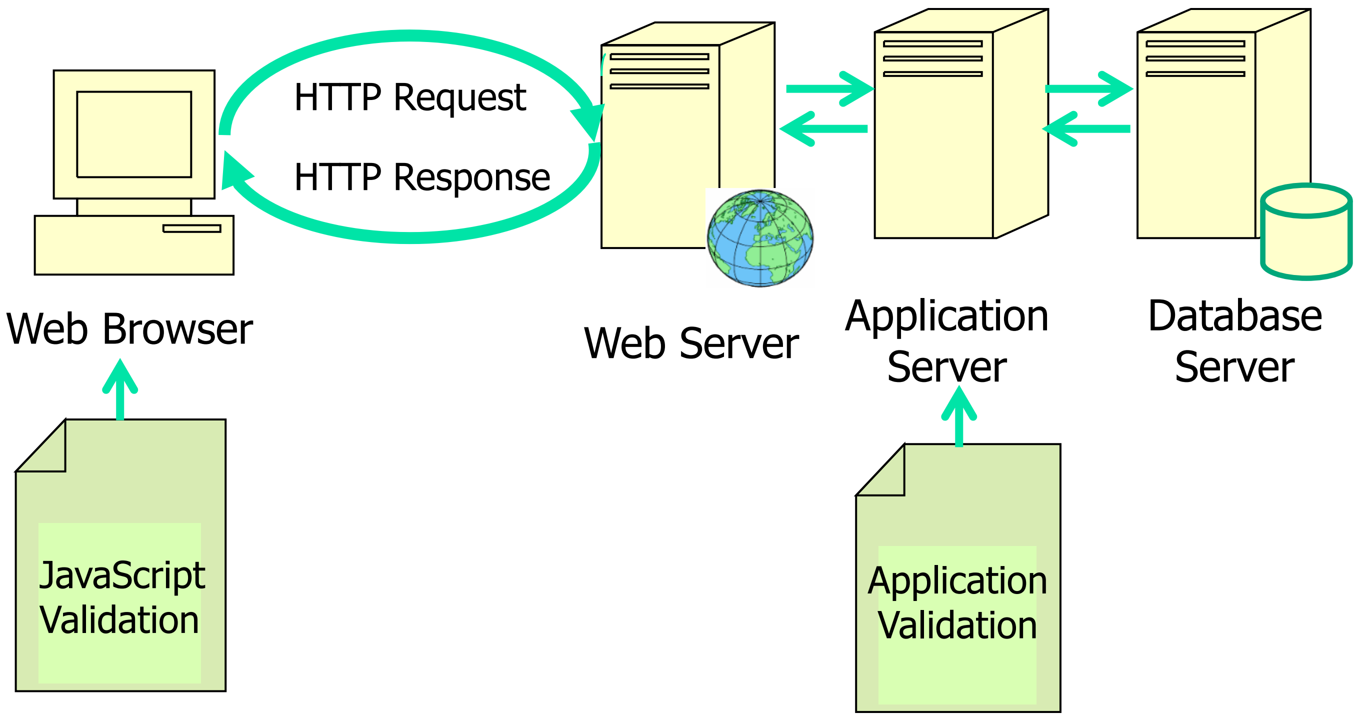
JavaScript Form Validation

Lecture#17

JavaScript Form Validation

- Before an HTML form is submitted, JavaScript can be used to provide client-side data validation
- This is more user-friendly for the user than server-side validation because it does not require a server round trip before giving feedback
- If the form is not valid, the form is not submitted until the errors are fixed

Client-Side Validation



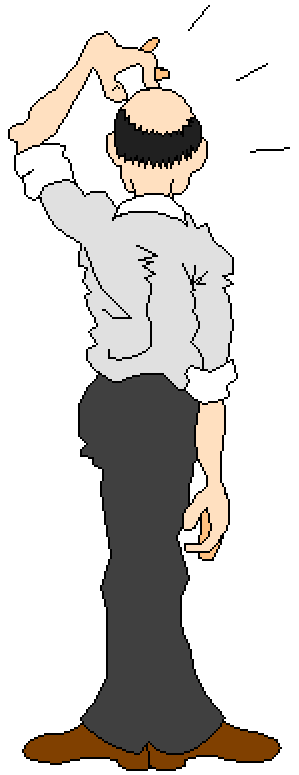
- JavaScript data validation happens before form is submitted
- Server-side application validation happens after the form is submitted to the application server

Client-Side vs Server-Side

- If creating a Web form, make sure the data submitted is valid and in the correct format
- Client-side validation gives the user faster feedback
- If possible, allow for server-side validation if the JavaScript is turned off
  - Covered in future INFO courses

What to Validate on a Form?

- Form data that typically are checked by a JavaScript could be:
  - were required fields left empty?
  - was a valid e-mail address entered?
  - was a valid date entered?
  - was text entered in a numeric field?
  - were numbers entered in an text field?
  - did the number entered have a correct range?



onchange Validation

- To force the browser to check each field immediately, we add an **onchange** event to each of the **<input>** tags in the form
- For example: if we wanted to check if the value of a certain text field had a valid e-mail address we would add this:

```
<input type="text" name="EMail" size="20"
        onchange="emailvalidation(this);" >
```

<form> onsubmit Event

- Your form must have a Submit button the user clicks when completing the form
- An **onsubmit** event will be raised and you should put this code in the **<form>** tag
- Call your event handler to go through and test each form field as needed
- If the event handler returns false, the form submission will be cancelled, if true the form will submit and the form action will be executed

```
<form action="Process.php" onsubmit="return validate(this);" >
```

onsubmit Event Handler

- Pass the form object as the **this** parameter
  - onsubmit=“return validate(this);”
- This function should create variables for each field that needs validation
- Set a inputvalid = true to begin with
- Use a series of if statements to perform each validation test, if test fails set inputvalid =false and set error message and/or alert message
- Finally, return inputvalid from the event handler

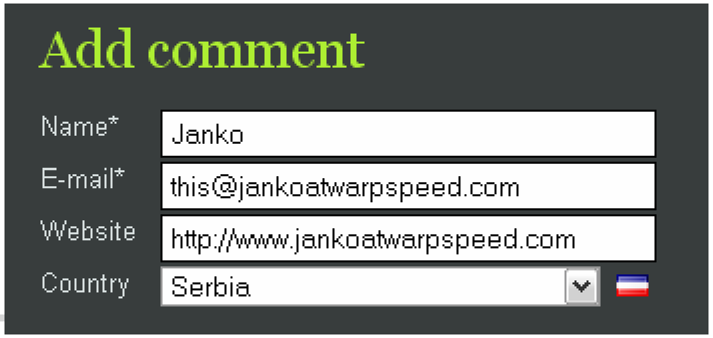
General Event Handler Structure

```
function validate(form) {
// Set each of needed form variables

var input valid=true;
var message="Please fix the following errors \n";

    if (!testFunction(text)) {
      // something is wrong
      // message =+ "new error \n";
      // validinput= false;
     }
     if (!testFunction2(text)) {
        // something else is wrong
        // message =+ "new error \n";
        // validinput= false;
      }
     // do each validation test
     if(!validinput) {
      alert (message);
     }
return validinput;
}
```

Required Fields


Add comment

Name*	Janko
E-mail*	this@jankoatwarpspeed.com
Website	http://www.jankoatwarpspeed.com
Country	Serbia

- What fields on a Web form should be required?
- Good usability practice suggests the form designer only make the user fill out necessary information
  - This information may be required when sent to a database such as non-null data
- Additional good practice would mark which fields are required on the form
  - Often marked with an *
  - May spell out the “Required” or style differently

Testing for Required Entry

- Checking a textbox field could be done with a simple test for **text.length == 0**
- Checking a select field to see if an option has been selected use **selectedIndex>0**
  - Place instruction text as the first <option>
- Checking a checkbox **checked==true**
- Checking a radiobutton need to loop through array and test each **checked==true**
  - May want to always set a default radio button as **selected="selected"**

Testing for Valid Input

- If a textbox asks for an email, test the text entered is a valid email
- If a textbox asks for a date, test the text entered is a valid date
  - Very complex to text format and validity
- If a textbox asks for a zipcode, test the text entered is a valid zipcode

Regular Expressions

- You can use a symbol representation of a string value called a **regular expression** to test your input text
- There are many regular expressions available for common tests
  - U.S. Phone: /^\(?(\d{3})\)?[-]?(\d{3})[-]?(\d{4})$/
  - Email: /^[0-9a-zA-Z]+@[0-9a-zA-Z]+[\.]{1}
    [0-9a-zA-Z]+[\.]?[0-9a-zA-Z]+$/
  - Currency: /^\s*(\+|-)?((\d+(\.\d\d)?)|(\.\d\d))\s*$/

Example Regular Expression

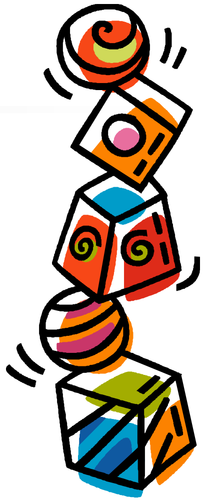
- Test to see if text is a number
- Returns true if number, false otherwise
- Does the text match the pattern?

```
// check if the text is a number
function IsNumber(fData) {
   var reg = /^[0-9]+[\.]?[0-9]+$/;
   return reg.test(fData)
}
```

Multiple Validations ???

- A Web form field may need more than one validation test
- For example a textbox for age:
  - It may be a required field
  - It must be a number
  - It must be in a range 0 - 120
- Order your validation tests from general to most specific

Modularize Your Functions



- It is a good idea to write a separate function for each type of validation
- Generalize each function
- Place these frequently used validation functions in their own external JavaScript file

Fine Tune Error Messages

- Help users to successfully complete the Web form
- Provide hints on formatting by the fields on the form rather than wait for a submission error
- Be professional and helpful with the error messages
- Try to provide both error message by each field and summary text in an alert box

Testing the Web Form

- First test that required fields must be provided
- Then test fields that need valid input such as:
  - Phone number
  - Email address
  - Dates
- Make sure error messages are appropriate and specific for each error

JavaScript Form Validation Summary

- JavaScript can be used for client-side data validation of a Web form
- Modularize validation functions to be reusable for future work
- Begin by making form user-friendly with instructions and hints
- Provide helpful error messages