WEB :: JavaScript

TalentSprint

Licensed To Skill

July 29, 2016

Learning Objectives

The content in this presentation is aimed at teaching learners to:

- Work on regular expressions
- Validate HTML WebPages

Java Script

Form Validation

- Used to occur at the server, after the client had entered all necessary data and then pressed the Submit button.
- The server would have to send all the data back to the client and request that the form be resubmitted with correct information.

JavaScript

Form Validation

- This was really a lengthy process and over burdening server.
- JavaScript provides a way to validate form's data on the client's computer.

JavaScript

Form Validation Functions

Basic Validation:

- Make sure that data was entered into each form field that required it.
- We will call validate() function to validate data when onsubmit event is occurring.

JavaScript

Form Validation Functions

Data Format Validation:

- Entered data must be checked for correct form and value.
- We can RegExp for validation of data.

JavaScript

Regular Expressions and RegExp Object

- An object that describes a pattern of characters.
- The JavaScript RegExp class represents regular expressions, and both String and RegExp define methods that use regular expressions to perform powerful pattern-matching and search and replace functions on text.

JavaScript

Regular Expressions and RegExp Object

- A regular expression could be defined with the RegExp() constructor:

```
var pattern = new RegExp(pattern, attributes);
var pattern = /pattern/attributes;
```

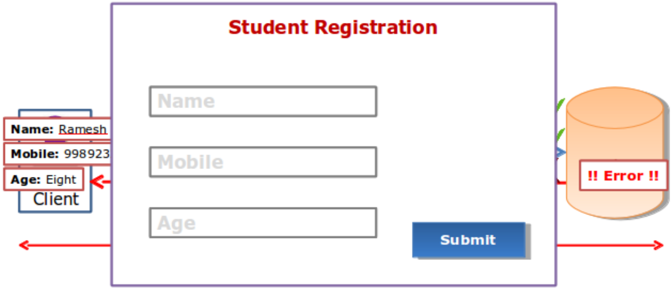
JavaScript

Browsers Compatibility

- Understand the differences between different browsers in order to handle each in the way it is expected.
- To get information about the browser your Web page is currently running in, use the built-in navigator object.

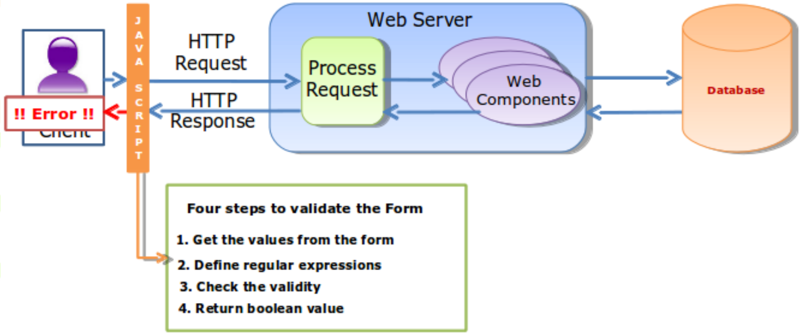
JavaScript

Validation



JavaScript

Validation



Four steps to validate the Form

1. Get the values from the form
2. Define regular expressions
3. Check the validity
4. Return boolean value

JavaScript

Validation

<**form name** = “reg” **action** = “TargetServer” onsubmit = “
return validate()”>

if **validate()**

- returns TRUE form will be submitted
- returns FALSE form will NOT be submitted

JavaScript

Student Registration

Name

```
<input type = “text” placeholder = “Name” name = “
sname !”>!
```

Should accept only alphanumeric characters, dot and space Minimum 6 and Maximum 20 characters

JavaScript

Mobile

<**input type** = “**text**” placeholder = “Mobile” **name** = “ mob !”>!

Should accept exactly 0 digits (0-9), Should starts with 9 or 8 or 7

JavaScript

Age

<**input type** = “**text**” placeholder = “Age” **name** = “ age !”>!

Should accept only max 2 digits (0-9)

Submit

</**form**>

JavaScript

Validation

Step - 1 Get the Value of text fields

Syntax

```
var value = document.<form_name>.<field_name>.value
    or
var value = document.getElementById("element_id").value
```

Snippet

```
var name = document.reg.sname.value;
var mob = document.reg.mob.value;
var age = document.reg.age.value;
```

JavaScript

Step - 2 Define Regular Expression

Syntax

```
var exp = new RegExp(“pattern”);
```

Snippet

```
var rname = new RegExp(“^[a–zA–Z .]{6,10}$”);
var rmob = new RegExp(“^[987][0–9]{9}$”);
var rage = new RegExp(“^[0–9]{1,2}$”);
```

JavaScript

Brackets

[...]	Any one character between the brackets.
[∧...]	Any one character not between the brackets.
[0-9]	It matches any decimal digit from 0 through 9.

JavaScript

Brackets

[a-z]	It matches any character from lowercase a through lowercase z.
[A-Z]	It matches any character from uppercase A through uppercase Z.
[a-Z]	It matches any character from lowercase a through uppercase Z.

JavaScript

Quantifiers

p+	It matches any string containing at least one p.
p*	It matches any string containing zero or more p's.
p?	It matches any string containing one or more p's.
pN	It matches any string containing a sequence of N p's.

JavaScript

Quantifiers

p2,3	It matches any string containing a sequence of two or three p's.
p2,	It matches any string containing a sequence of at least two p's.
p$	It matches any string with p at the end of it.
∧p	It matches any string with p at the beginning of it.

JavaScript

Metacharacters

.	a single character
\s	a whitespace character (space, tab, newline)
\S	non-whitespace character
\d	a digit (0-9)
\D	a non-digit
\w	a word character (a-z, A-Z, 0-9, _)

JavaScript

Metacharacters

\W	a non-word character
[\b]	a literal backspace (special case)
[aeiou]	matches a single character in the given set
[^aeiou]	matches a single character outside the given set

JavaScript

Step - 3 Check the validity

Syntax: regexp.test(data)

Snippet

```
if (rname.test(name))
    if (rmob.test(mob))
        if (rage.mob(age))
            return true;
        else
            return false;
    else
        return false;
else
    return false;
```

JavaScript

Step - 4 Return boolean value

JavaScript

RegExp Methods

exec()	executes a search for a match in its string parameter.
test()	tests for a match in its string parameter.
toSource()	returns an object literal representing the specified object; you can use this value to create a new object.
toString()	returns a string representing the specified object.

JavaScript

Complete Program

```
function validate(){
    var name = document.reg.sname.value;
    var mob = document.reg.mob.value;
    var age = document.reg.age.value;
    var rname = new RegExp("^[a–zA–Z .]{6,10}$");
    var rmob = new RegExp("^[987][0–9]{9}$");
    var rage = new RegExp("^[0–9]{1,2}$");
```

Complete Program - Cont...

```
    if (rname.test(name))
        if (rmob.test(mob))
            if (rage.mob(age))
                return true;
            else
                return false;
        else
            return false;
    else
        return false;
}
```

JavaScript