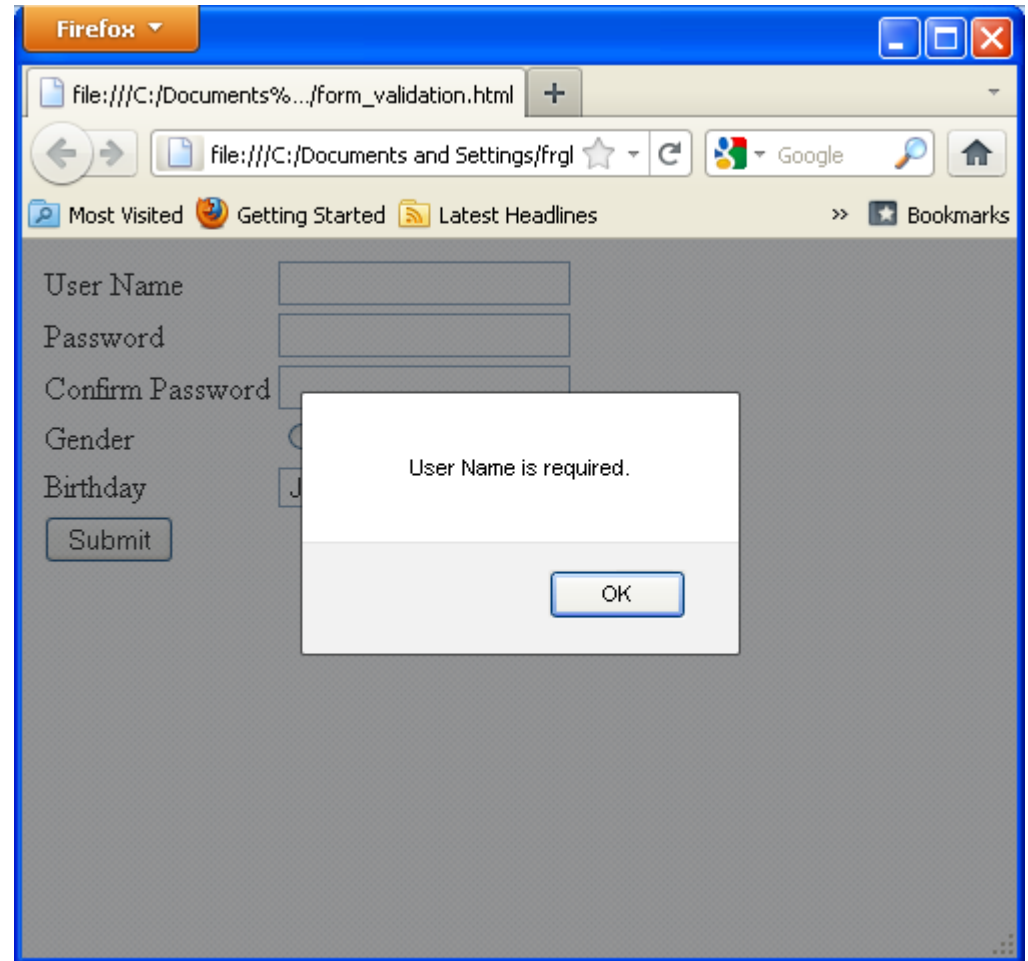


One Meeting With Javascript

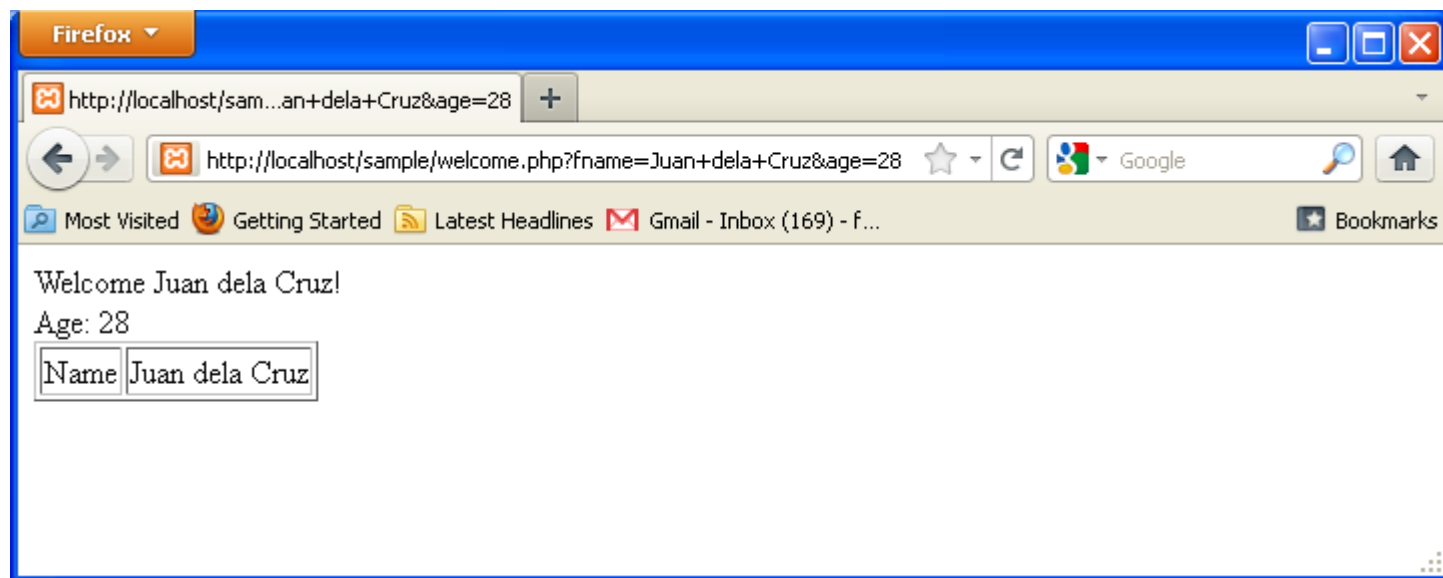
Client-Server Programming

- Client side programming/scripting
 - a program or a script is executed at the client side
 - in a web application this is usually UI-related/page-content modification tasks



Client-Server Programming

- Server side programming/scripting
 - a program or a script executed on OR by the (web) server.
 - in a web application, server side programs usually represent business logic and data processing.



Scripting Languages

- **AKA extension languages**
- Languages designed to provide 'control' over a predefined system.
- i.e. Operating systems have shell scripting languages
 - Allows you to issue OS-related commands.

Client Side Scripting

- ECMAScript / Javascript
 - Created by Brendan Eich for Netscape
 - Originally Mocha, then LiveScript, then Javascript
 - Included in Netscape Navigator 2.0



Client Side Scripting

- ECMAScript / Javascript
 - Microsoft came out with Jscript in MSIE 3
 - Submitted to ECMA for standardization
 - ECMA-262 Specification
 - ECMAScript is a compromise name.



Client Side Scripting

- VBScript
 - in MSIE
- PerlScript*

Javascript

- Popular in client-side programming although several dialects exists that are used in server-side programming/scripting.
- Other dialects:
 - ActionScript (in Adobe Flash and Flex)
 - DMDScript
 - InScript

Javascript Syntax

- Pseudo-Java syntax.
- Weakly/dynamically typed – type is assigned to values not variables.
- Prototype-based language
 - It is possible to declare “classes” through **prototypes**.
- Functions are first-class citizens
 - They can be parameters to other functions
 - Inner functions
 - Functions used in declaring prototypes and constructors.

Javascript and Browsers

- Javascript is executed by Web Browsers in a sandboxed environment
 - Operations limited to and within the web page only.
 - No client side I/O operations possible.
- Objects in Browsers
 - document (DOM, i.e. the data structure that represents the web page being viewed)
 - window
 - browser properties
 - Some built in object types (i.e. Date, Math, String)

Javascript Embedding

```
<html>
```

```
  <head>
```

```
    ...
```

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

```
    <!--
```

```
      function myFunction() {
```

```
      }
```

```
    -->
```

```
  </script>
```

Javascript Embedding

```
<html>
```

```
...
```

```
<head>
```

```
<link rel="javascript" href="jsfile.js"  
      type="text/javascript" />
```

```
</head>
```

```
...
```

Javascript Embedding

`<form onsubmit="script">`

`...`

`<input type='button' onclick='script'/>`

`...`

` ...`

Directions of the Language

- New features are to be added to the next version
 - Classes
 - Structural types
 - packages/namespaces
 - optional type annotations
 - etc.

Things of Note

- AJAX
 - Asynchronous Javascript And XML
 - A means of communicating with the server asynchronously.
- JSON
 - Javascript Object Notation
- Javascript DOM (Document Object Model)
 - This is a standard API.

Examples: JS DOM

```
function changeStyle(objID) {  
    //standard DOM method  
    var el = document.getElementById(objID);  
  
    //accessing web page element  
    //properties  
    el.style.backgroundColor = '#00FF22';  
  
}
```


Example: JS DOM

```
<form name="myform">
```

```
  <input type="checkbox" checked="false"  
    name="chk" />
```

```
  <input type="button" onclick=  
    "document.myform.chk.checked='true'"  
  />
```

```
</form>
```

Example: JSON

- Object representation using associative array.

```
{
```

```
  "firstName" : "Jose",
```

```
  "lastName" : "Jalapeno",
```

```
  "address" {
```

```
    "streetAddress" : "#13 Balet Drive",
```

```
    "city" : "Mexico City",
```

```
    "province" : "Pampanga"
```

```
  }
```

```
}
```

Example: AJAX

```
var the_object;  
  
var http_request = new XMLHttpRequest();  
  
http_request.open( "GET", url, true );  
  
http_request.onreadystatechange = function () {  
    if ( http_request.readyState == 4 ) {  
        if ( http_request.status == 200 ) {  
            the_object =  
                eval( "(" + http_request.responseText + ")" );  
        } else {  
            alert( "There was a problem with the URL." );  
        }  
        http_request = null;  
    }  
}
```