ACCP I7.1 – SEMESTER 3

BEGINNING ASP.NET

Module 3 & 4

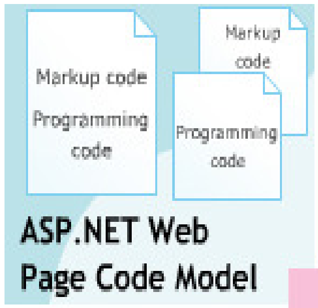
Session 2 - Basics of ASP.NET

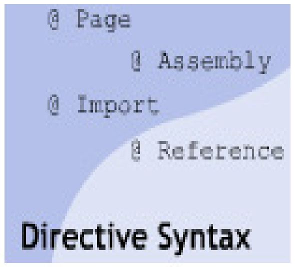
Session 1 Review

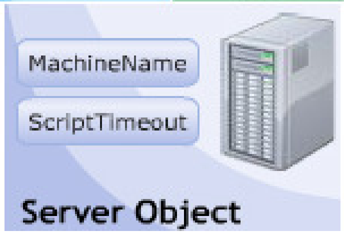
- WebForm and ASP.NET 2.0
- ASP.NET Application Development
- Working with Visual Studio 2005 IDE
- Configure ASP.NET Application with IIS
- Features of the New Web Development Environment

Objectives

- ASP.NET Web Page Code Model
- Directive Syntax
- Basic Event Handling in Web Pages
- Request Object
- Response Object
- Server Object

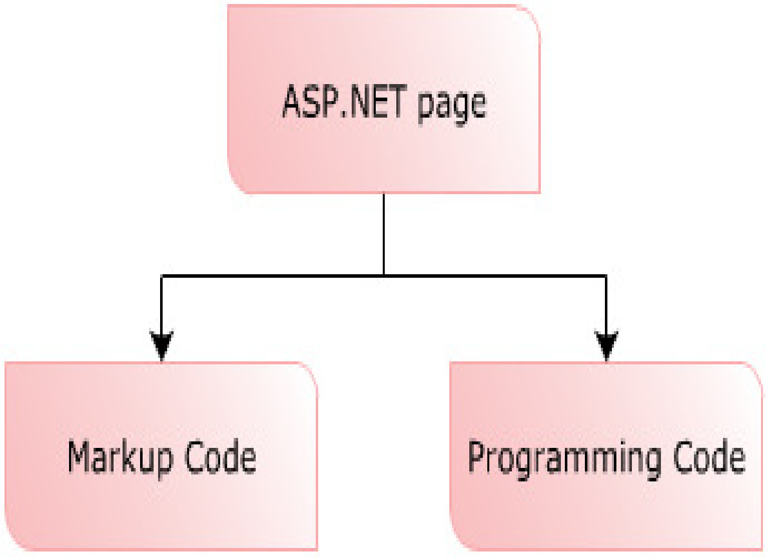






ASP.NET Web Page Code Model

- Web page is created using markup and programming code
  - Layout : markup
  - Logic : programming code
- Markup and code can either be in same file or different files → there are two type of Web page models : single-file page model and code-behind model



ASP.NET Code Page Model

Single-File Page model

Advantages

- Easier to maintain
- Easier to deploy
- Easier to rename the single-file page

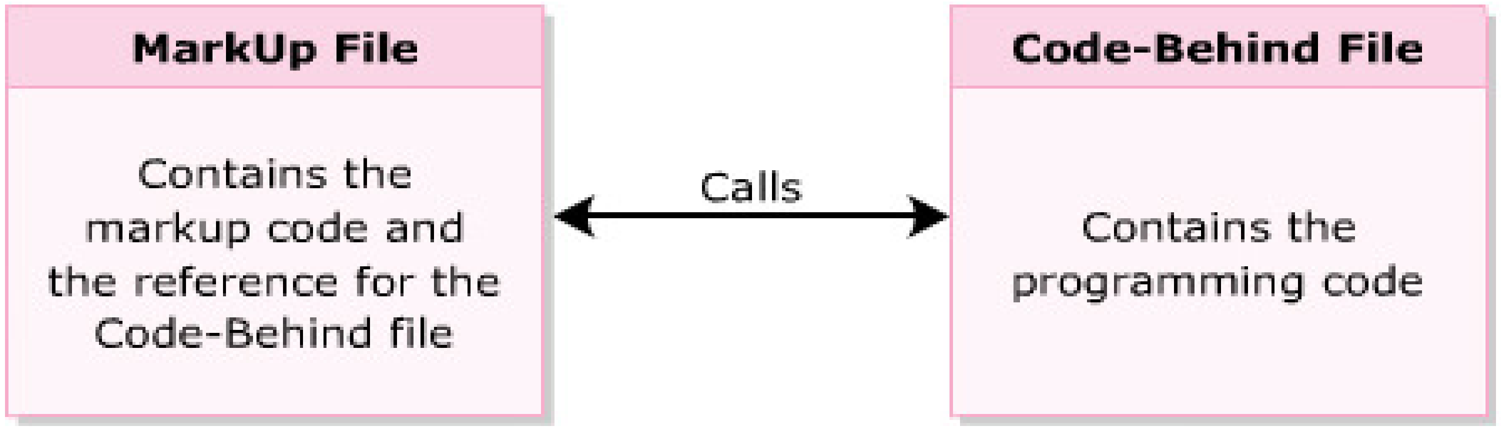
Limitations

- A single-file page cannot be directly created in VS
- HTML editor has limited coding support
- The event-handler cannot be created easily

ASP.NET Code Page Model

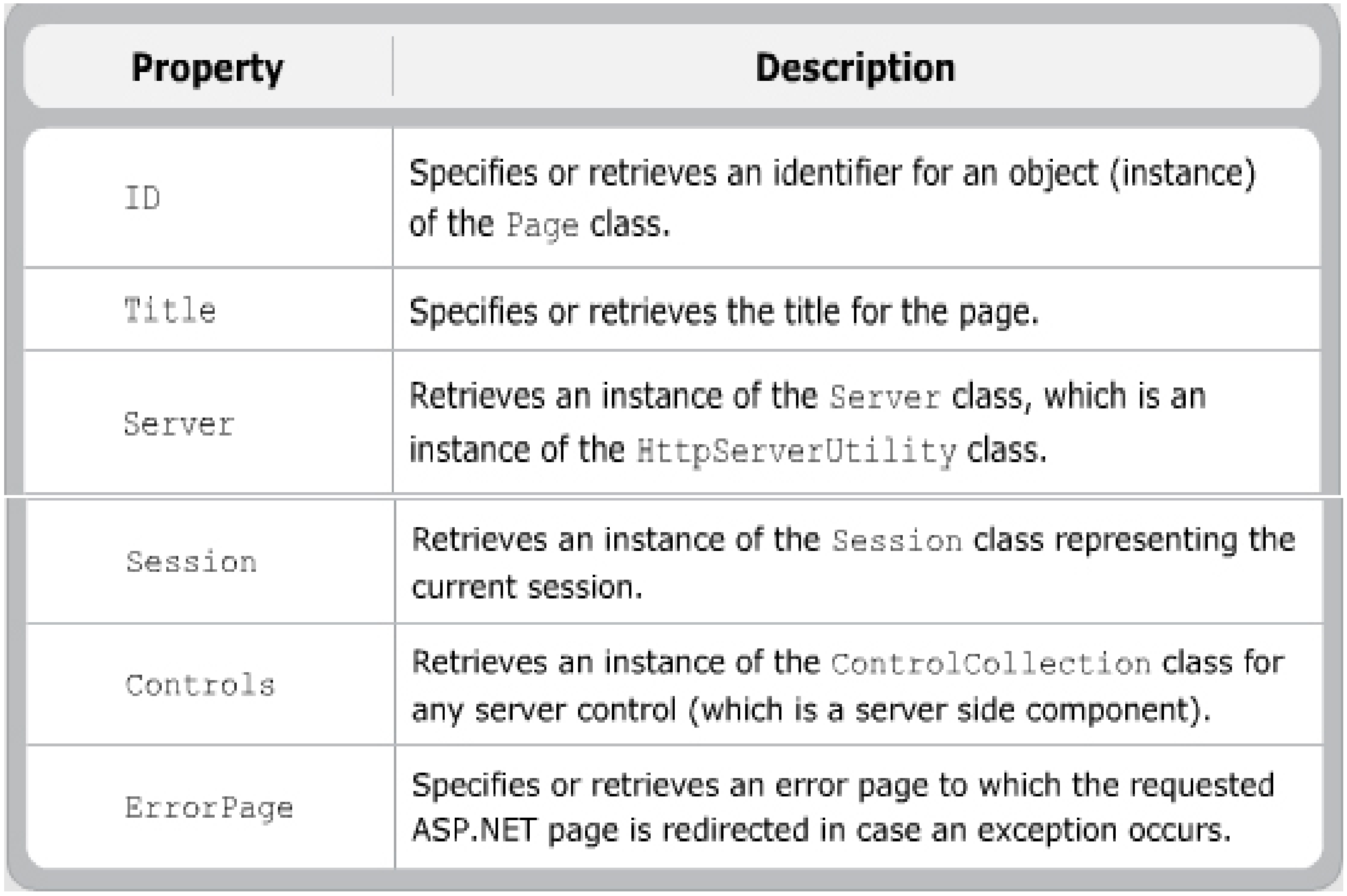
Code-Behind Page model

- Markup and code are separated
- Markup file extension: .aspx
- Programming code file: .aspx.cs


MarkUp File
Contains the markup code and the reference for the Code-Behind file
Calls
Code-Behind File
Contains the programming code

The **System.Web.UI.Page** class

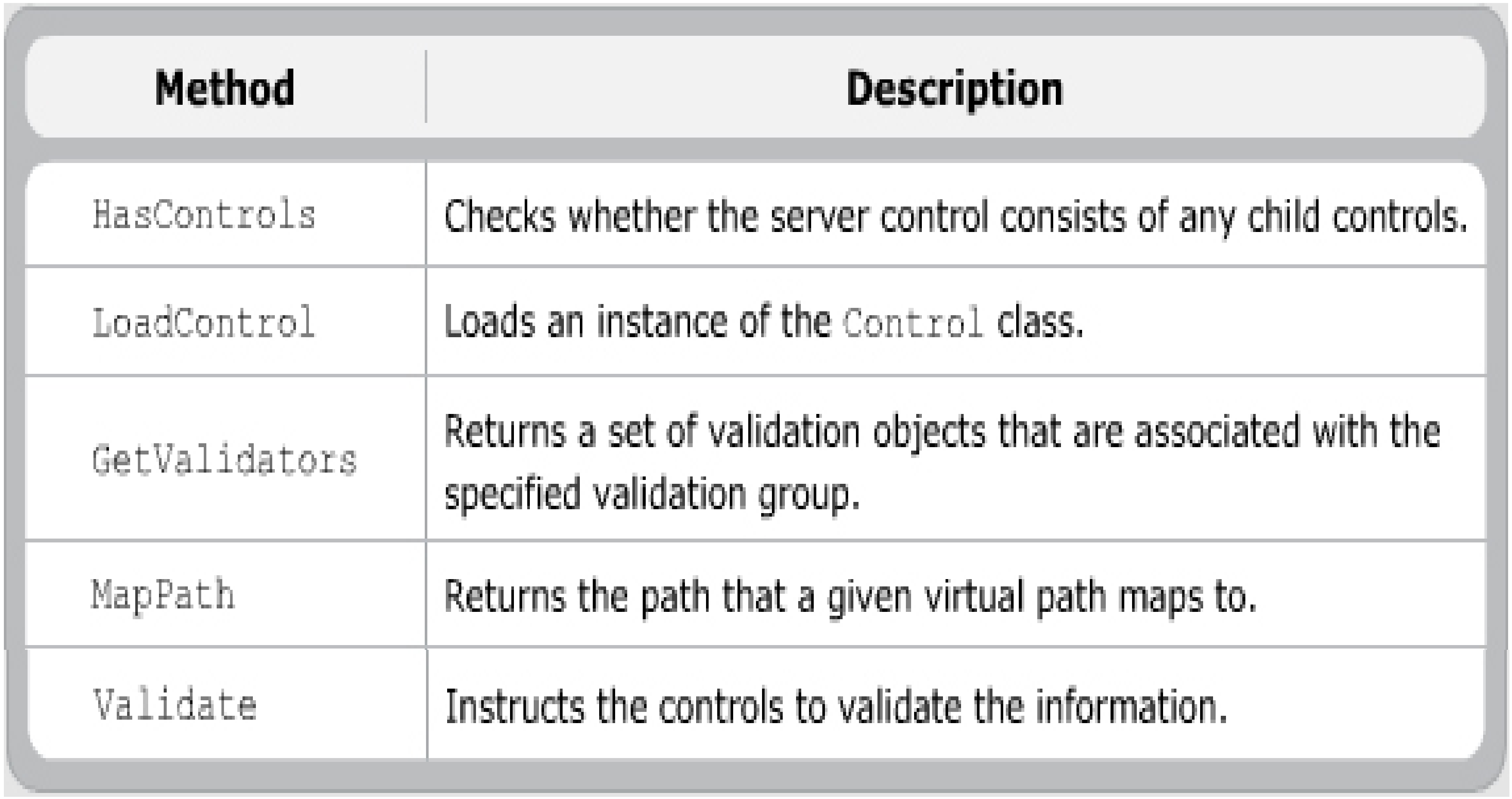
Properties



Property	Description
`ID`	Specifies or retrieves an identifier for an object (instance) of the `Page` class.
`Title`	Specifies or retrieves the title for the page.
`Server`	Retrieves an instance of the `Server` class, which is an instance of the `HttpServerUtility` class.
`Session`	Retrieves an instance of the `Session` class representing the current session.
`Controls`	Retrieves an instance of the `ControlCollection` class for any server control (which is a server side component).
`ErrorPage`	Specifies or retrieves an error page to which the requested ASP.NET page is redirected in case an exception occurs.

The **System.Web.UI.Page** class

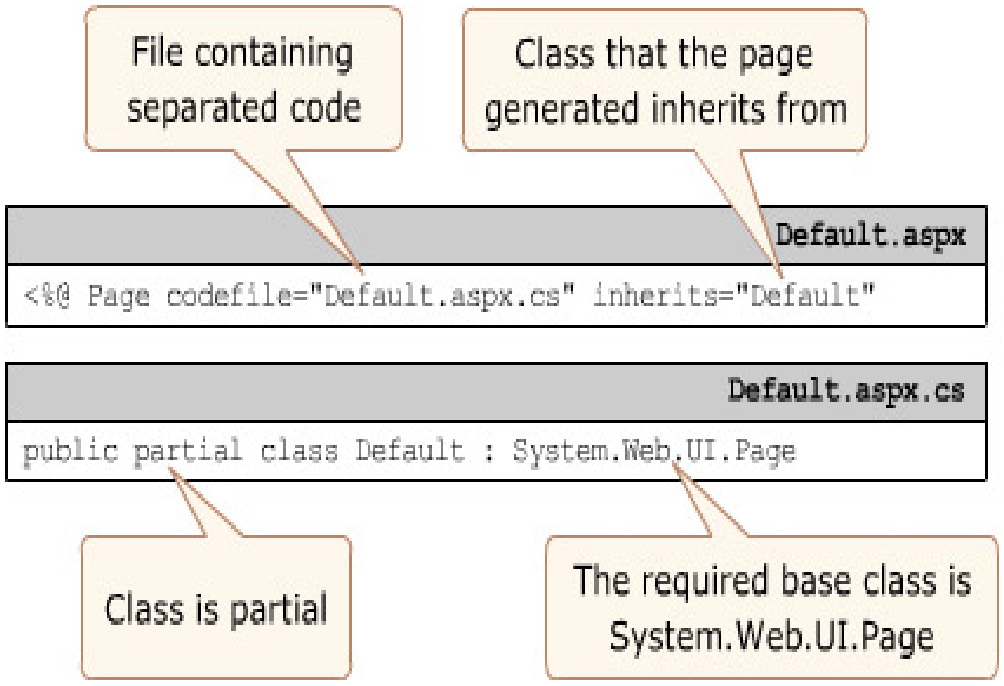
Methods



Method	Description
`HasControls`	Checks whether the server control consists of any child controls.
`LoadControl`	Loads an instance of the `Control` class.
`GetValidators`	Returns a set of validation objects that are associated with the specified validation group.
`MapPath`	Returns the path that a given virtual path maps to.
`Validate`	Instructs the controls to validate the information.

Partial class

- Code behind files automatically creates a partial class
- Declaration with "partial" keyword
- Does not contain the complete implementation
- Inherited from the Page class


File containing separated code
Class that the page generated inherits from
Default.aspx
Default.aspx.cs
Class is partial
The required base class is System.Web.UI.Page

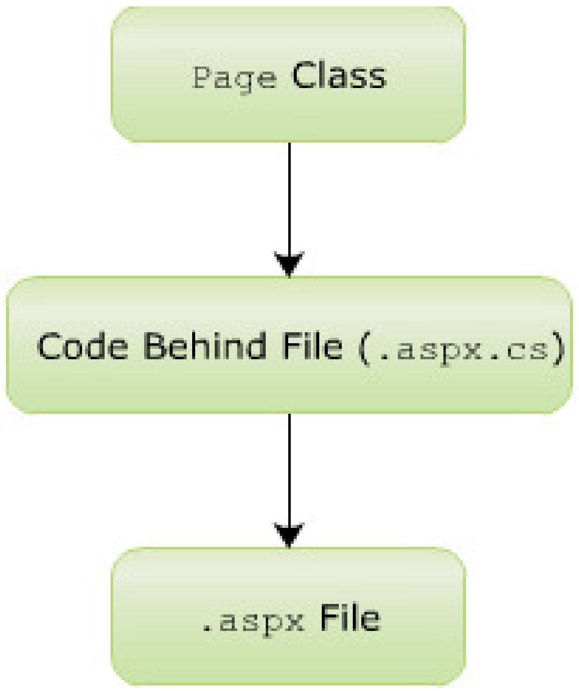
```
<%@ Page codefile="Default.aspx.cs" inherits="Default"
```

```
public partial class Default : System.Web.UI.Page
```

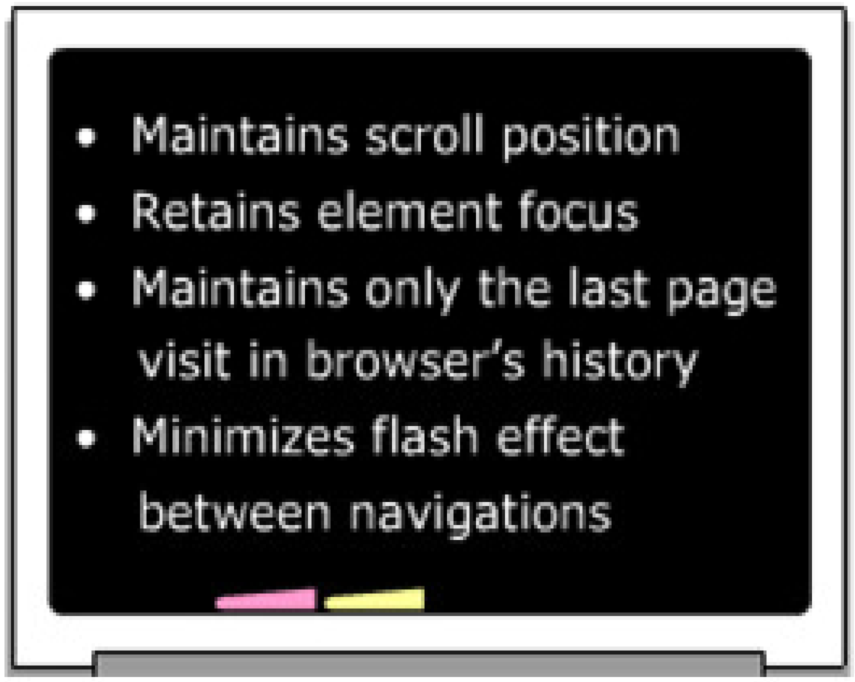
ASP.NET Code Page Model

Advantages of Code-Behind Files

- Code separation
- Reusability of code across multiple pages.
- Hiding of application logic
- Providing browser incompatibility



Smart Navigation

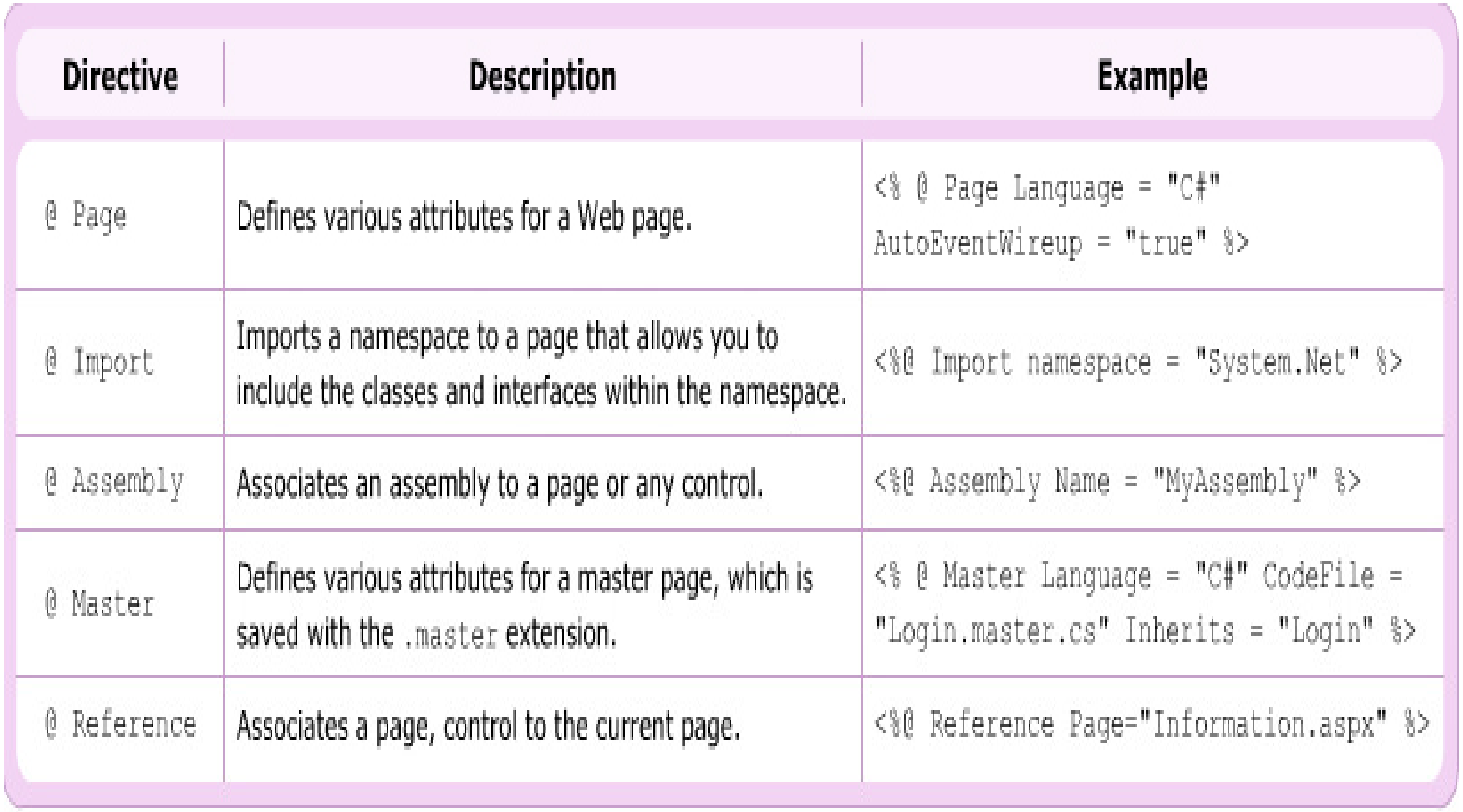


- Maintains scroll position
- Retains element focus
- Maintains only the last page visit in browser's history
- Minimizes flash effect between navigations

- Smart navigation feature was implemented by using SmartNavigation property (obsolete)
- In ASP.NET 2.0, smart navigation can be implemented by using SetFocus() method and MaintainScrollPositionOnPostBack property

ASP.NET Directives

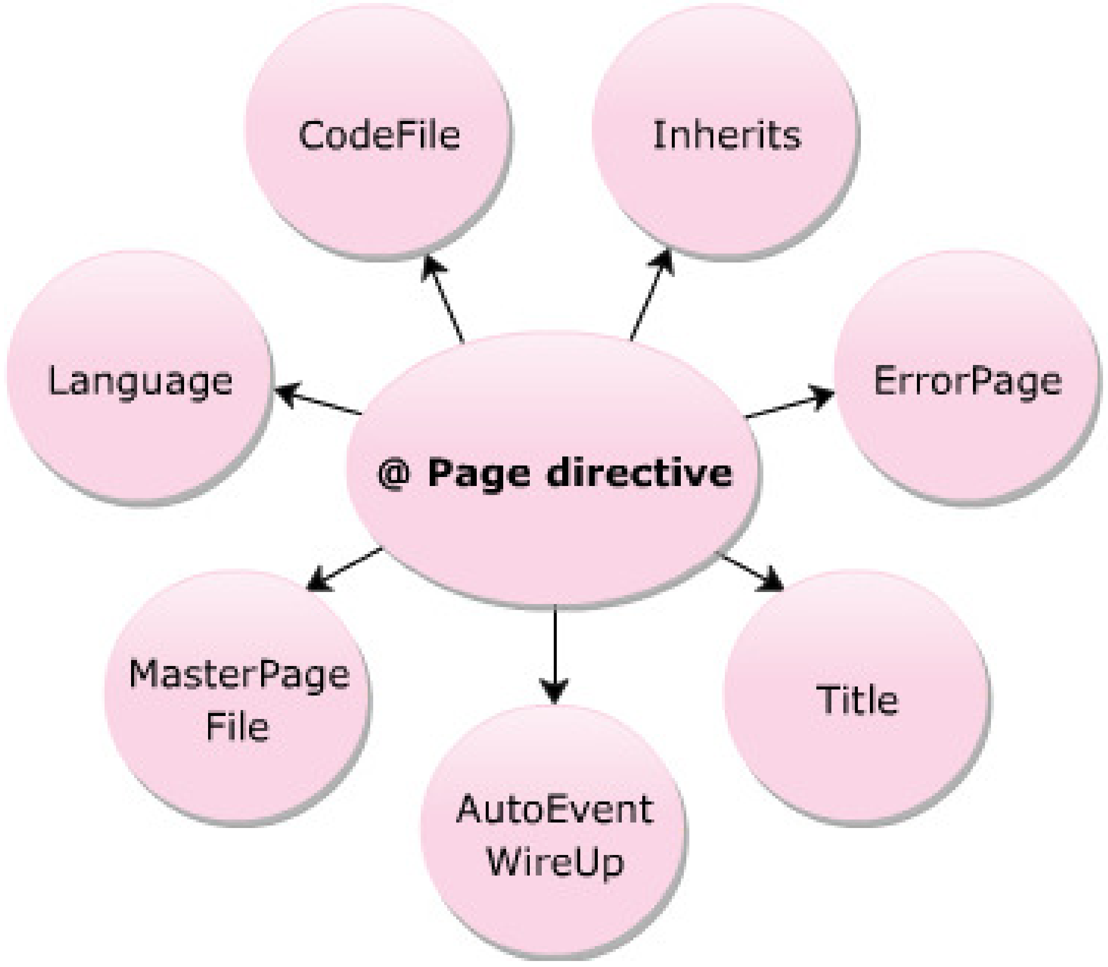
Directive are commands that describe how an ASP.NET compiled and processed



Directive	Description	Example
`@ Page`	Defines various attributes for a Web page.	`<% @ Page Language = "C#" AutoEventWireup = "true" %>`
`@ Import`	Imports a namespace to a page that allows you to include the classes and interfaces within the namespace.	`<%@ Import namespace = "System.Net" %>`
`@ Assembly`	Associates an assembly to a page or any control.	`<%@ Assembly Name = "MyAssembly" %>`
`@ Master`	Defines various attributes for a master page, which is saved with the `.master` extension.	`<% @ Master Language = "C#" CodeFile = "Login.master.cs" Inherits = "Login" %>`
`@ Reference`	Associates a page, control to the current page.	`<%@ Reference Page="Information.aspx" %>`

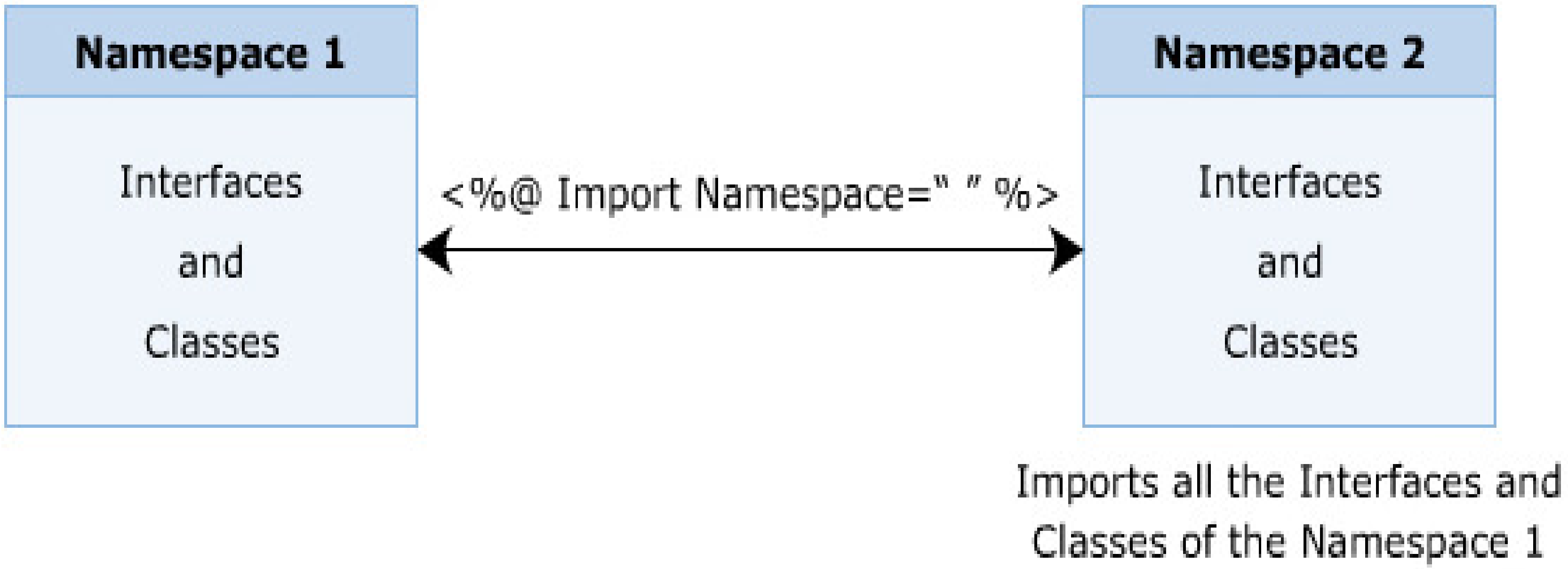
“@Page” Directive

This directive allows you to specify different attributes for a page



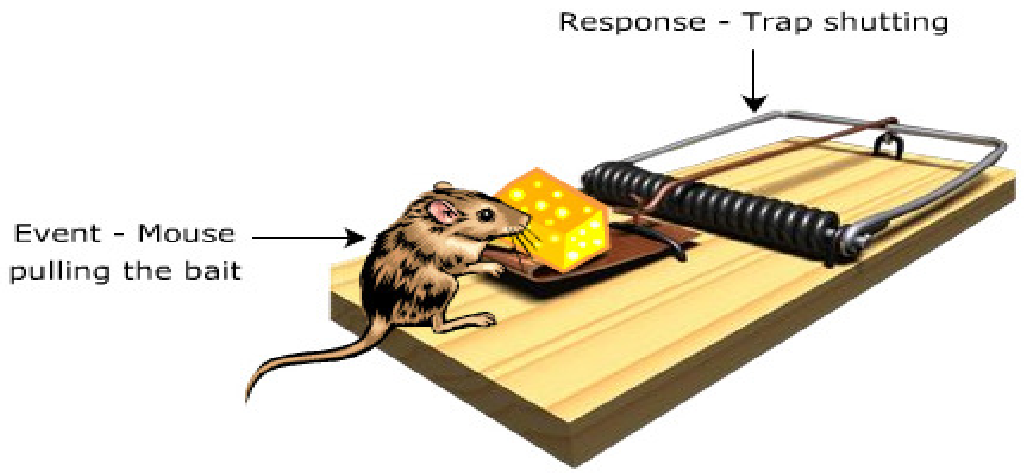
"@Import" Directive

- Allows you to explicitly include in your Web Page different functionalities that are declared in other namespaces
- Allows one attribute : **namespace**


Namespace 1
Interfaces
and
Classes
<%@ Import Namespace=" " %>
Namespace 2
Interfaces
and
Classes
Imports all the Interfaces and
Classes of the Namespace 1

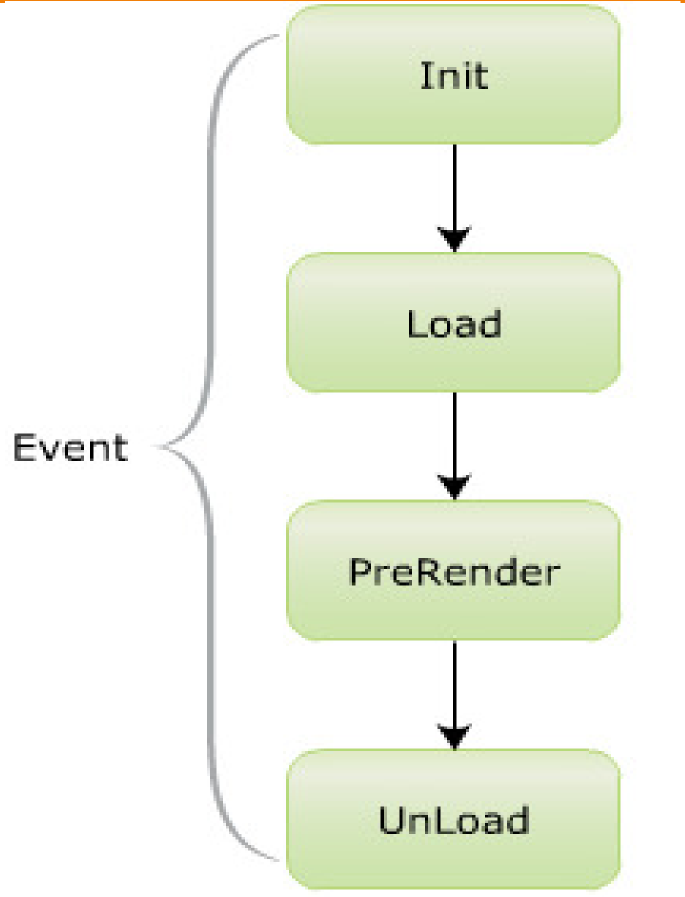
Event Handling

- **Event:**
  - actions that are fired while the application is running
- **Event handler:**
  - block of code which is executed when an event occurs
- Events can be handled manually and automatically



Automatic Event Handling

- Each stage in a page life cycle can raise an event
- These events are handled by associated **handler**

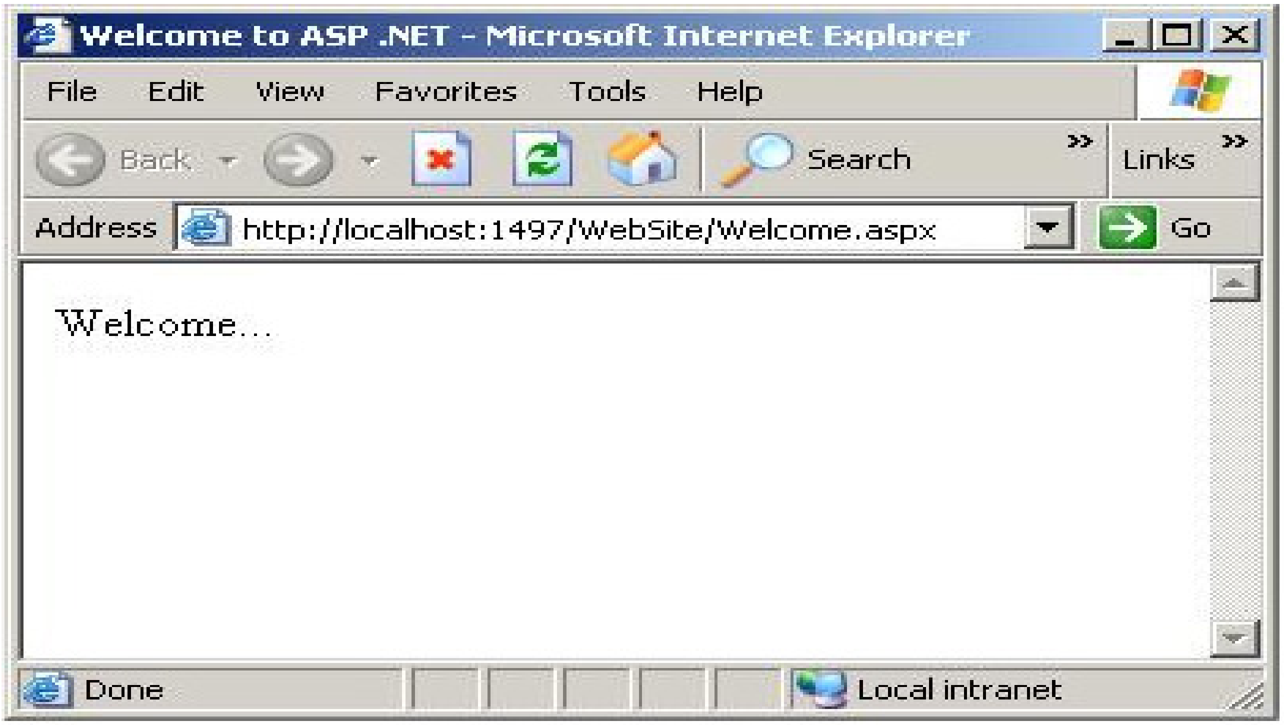


Web page life cycle

Basic Event Handling in Web Pages

Page_Load Event

- Is triggered each time a web page is requested

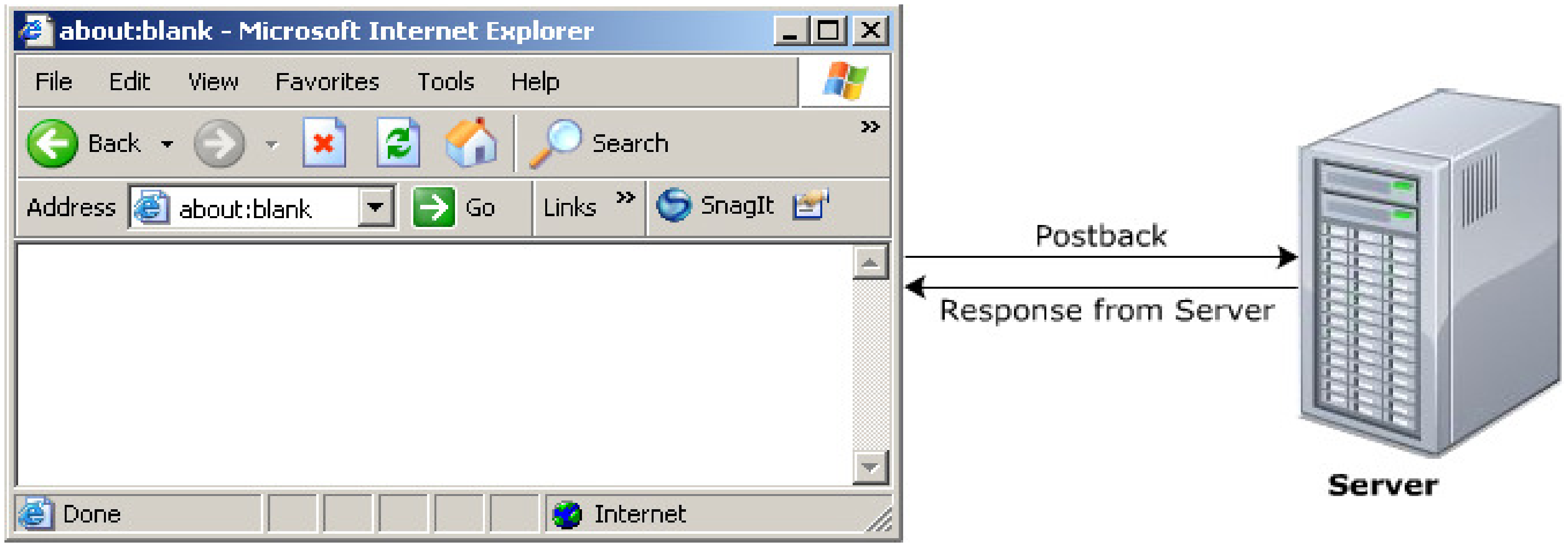

Welcome to ASP .NET - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Links
Address
http://localhost:1497/WebSite/Welcome.aspx
Go
Done
Local intranet

Welcome...

Basic Event Handling in Web Pages

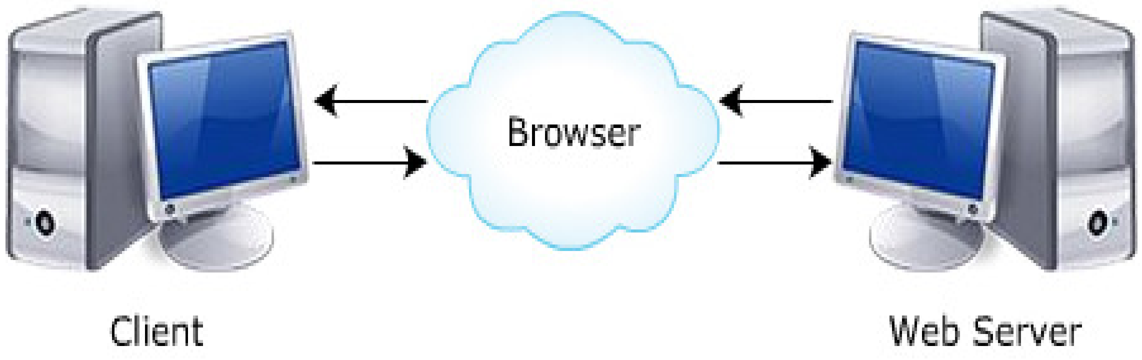
Postback

- **Postback** is the information submitted by the browser to the server for processing.
- The ***`IsPostBack`*** property checks whether the Web page is requested for the first time or is a result of a postback.


about:blank - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address about:blank Go Links SnagIt
Done Internet
Postback
Response from Server
Server

“Request” ,“Response” and “Server” Object

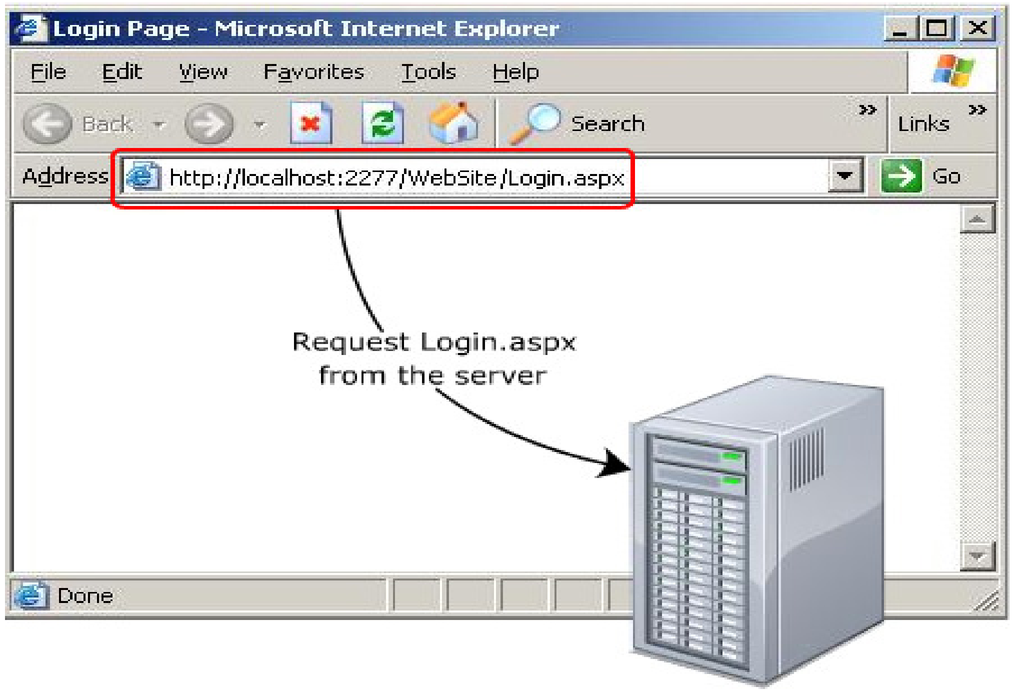
- **Request** objects represent the incoming information
- **Response** objects represent the information going out of the Web server
- **Server** objects provide the utility methods for web application



“Request” Object and HttpRequest class

- **Properties**
  - ApplicationPath
  - Browser
  - ContentLength
  - FilePath
  - **QueryString**
  - ReguestType

- **Methods**
  - SaveAs
  - MapImageCoordinates
  - MapPath


Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Links
Address http://localhost:2277/WebSite/Login.aspx Go
Request Login.aspx from the server
Done

Getting User's Request

QueryString Property

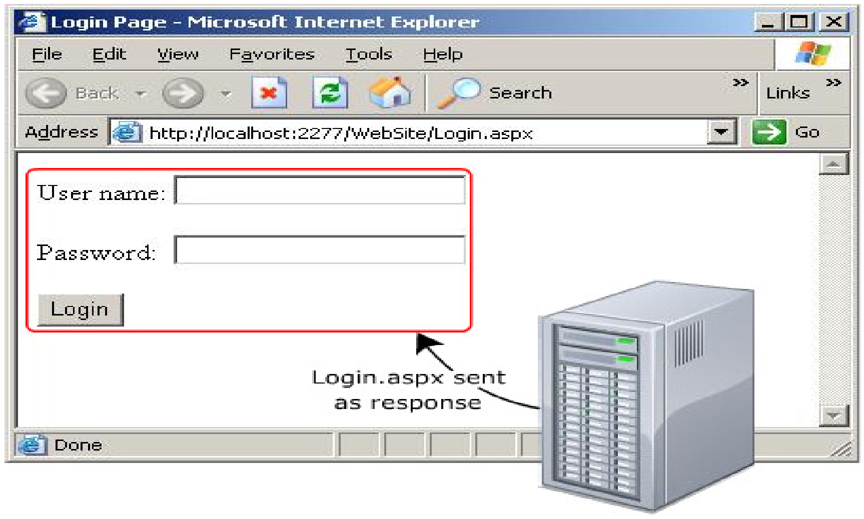
- The QueryString property returns a collection of name-value pairs that represent the elements of a form.
- *Syntax*:
  - **`Request.QueryString["varName"]`**
- *Example*: *(getting values of all input element)*

```
foreach ( string varNames in Request.QueryString ) {
   Response.Write { Request.QueryString[varNames] + <br />);
}
```

Response Object and HttpResponse class

- **Properties**
  - BufferOutput
  - Charset
  - ContentEncoding
  - ContentType
  - *Cookies*
  - **IsClientConnected**
- **Methods**
  - Clear
  - ClearContent
  - Close
  - *End*
  - ***Redirect***
  - **Write**

Response object stores information related to the server's response


Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Links
Address http://localhost:2277/WebSite/Login.aspx Go
User name:
Password:
Login
Login.aspx sent as response
Done

Server Object and HttpServerUtility Class

- **Properties**
  - MachineName
  - ScriptTimeout

- **Methods**
  - Execute
  - HtmlEncode
  - MapPath
  - UrlEncode

Server object exposes various utility methods that can be used to transfer control between pages, decode HTML text, get error information

Summary – Workshop Activities

- Code Behind Model
- Directive Syntax
- Basic Event Handling in Web Pages
- Properties and method of the Request Object
- Properties and method of the Response Object
- Properties of the Server Object

Next session..

- Web Server Controls