CS108: Advanced Database
- ASP.NET Programming

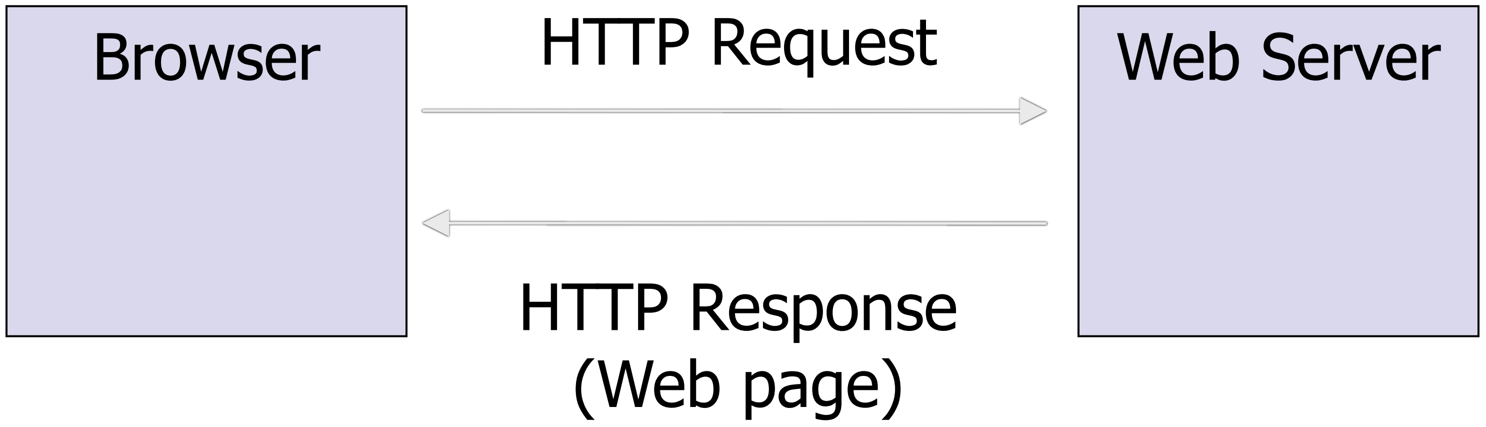
Lecture 01:

Introduction to ASP.NET

The Web Architecture and ASP.NET

Review of the Web

- It began with HTTP and HTML, which delivers static Web pages to browsers which would render those pages



- index.html written, stored, put on server, displayed when it’s url is requested

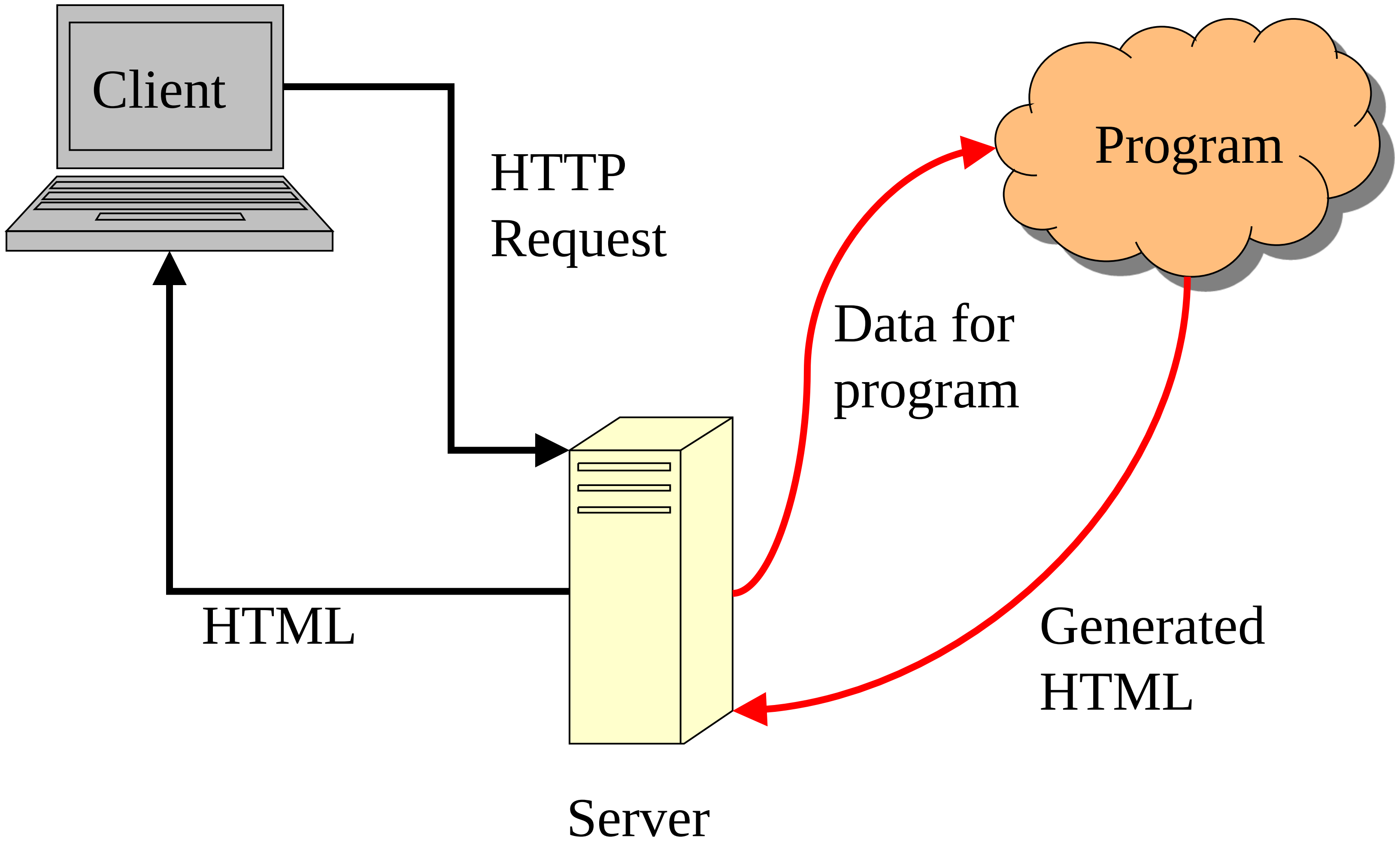
Dynamic Webpages

- Upon url request
  - Somehow decide to dynamically generate an html page (from scratch)
  - Send back new html page to user
- No html file exists on server, just created on demand
- CGI/Perl, Java servlets, etc.

CGI

- CGI: Common Gateway Interface
  - Not a programming language!
  - Just an interface (connection) between the webserver and an outside program
  - “Webserver” = webserver software, e.g., Apache
- Very simple basic idea:
  - User chooses an url
  - Webserver runs that url’s program
  - Sends back the program’s output

On-the-fly content with CGI



■Image from http://www.scit.wlv.ac.uk/~jphb/cp3024/

CGI Input

- CGI programs must respond to input, there are two ways
- ***GET***: string is part of the URL, following a ?:

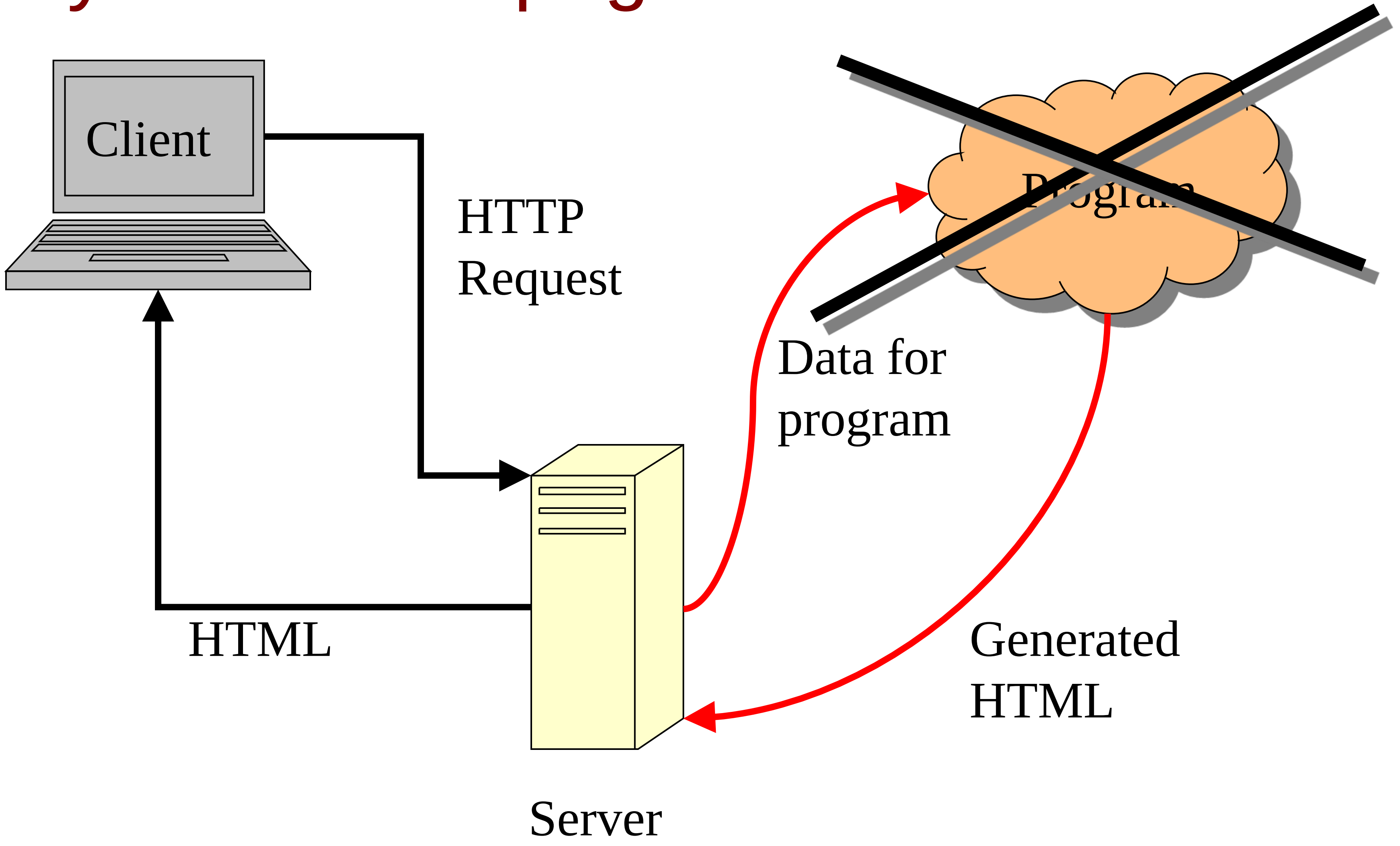
https://www.google.com/search?q=ASP.NET

- ***POST***: string can be read by program from an environmental variable
  - Vars not visible to the browser user
  - Not automatically put in server log, etc.

Dynamic Webpages

- Need webpages to respond to user inputs
  - Create a an html file *embedded* with special non-html code
  - Upon url request, execute embedded code to generate more html/fill in the file
  - Send back the modified html page to user
  - An incomplete html page exists on server
  - Examples: PHP, JSPs, ASPs, etc.

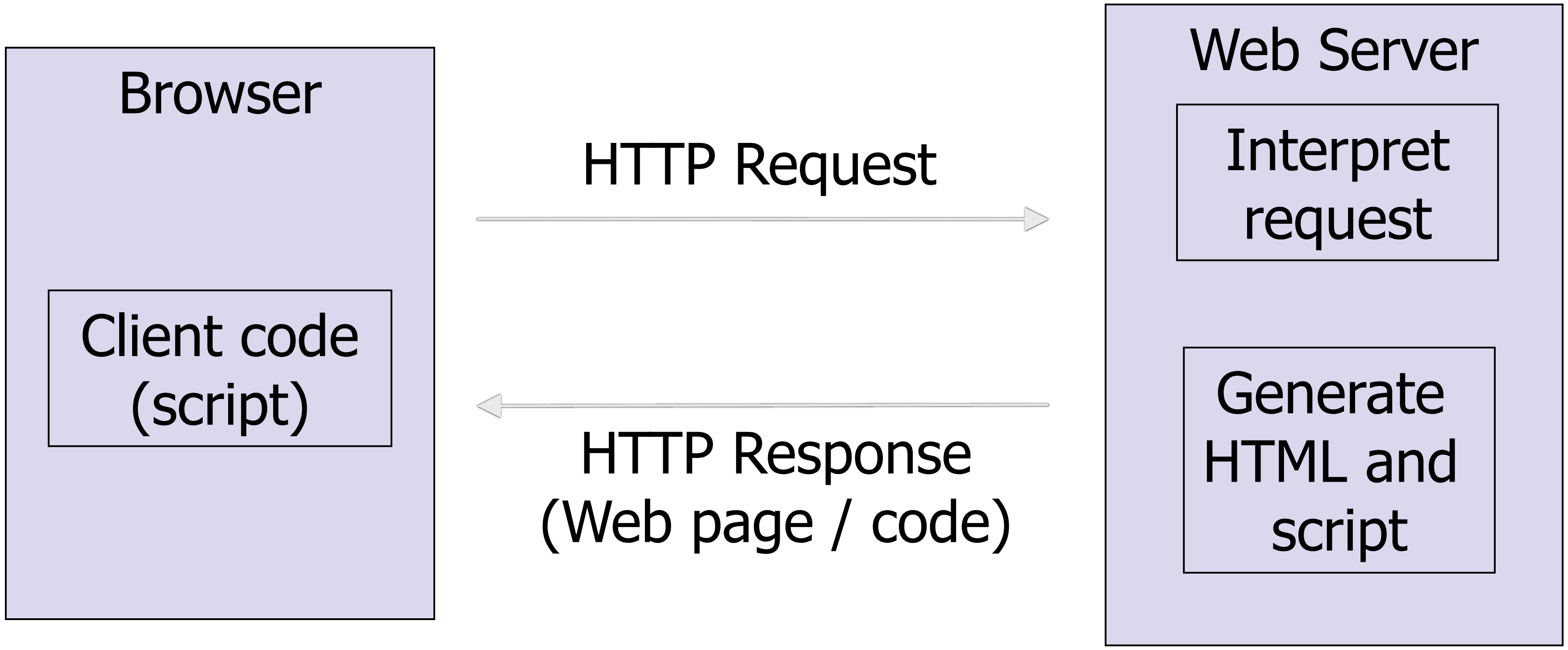
Dynamic Webpages



■Image from http://www.scit.wlv.ac.uk/~jphb/cp3024/

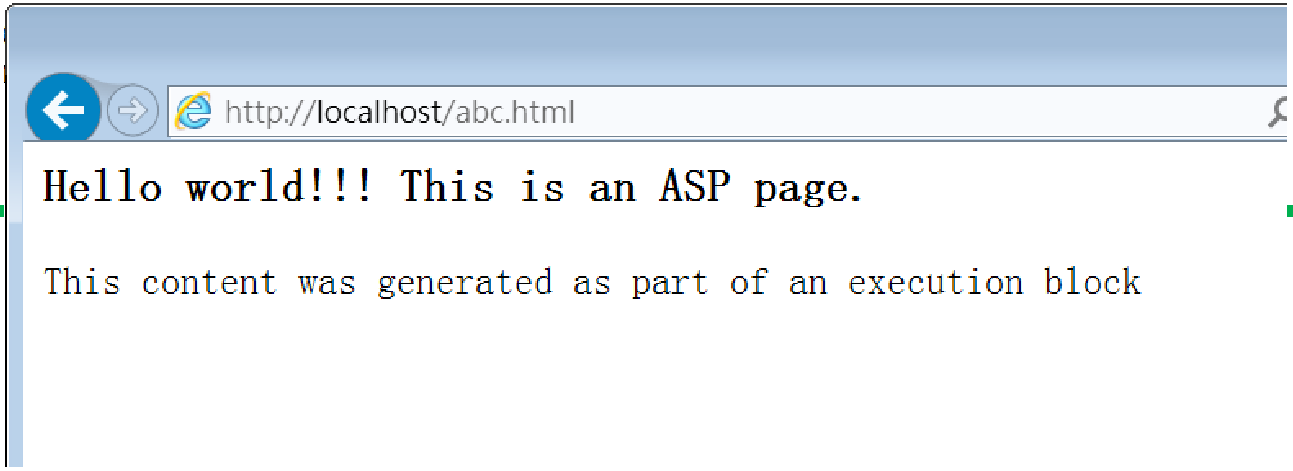
Dynamic Webpages

- Server-side applications were created to execute code and dynamically generate Web pages based on a particular request



Example: ASP

```
<%@ Language="javascript" %>
<html>
 <body>
  <form>
   <h3>Hello world!!! This is an ASP page.</h3>
    <% Response.Write("This content was generated ");%>
    <% Response.Write("as part of an execution block");%>
   </form>
 </body>
</html>
```



http://localhost/abc.html

```
Hello world!!! This is an ASP page.

This content was generated as part of an execution block
```

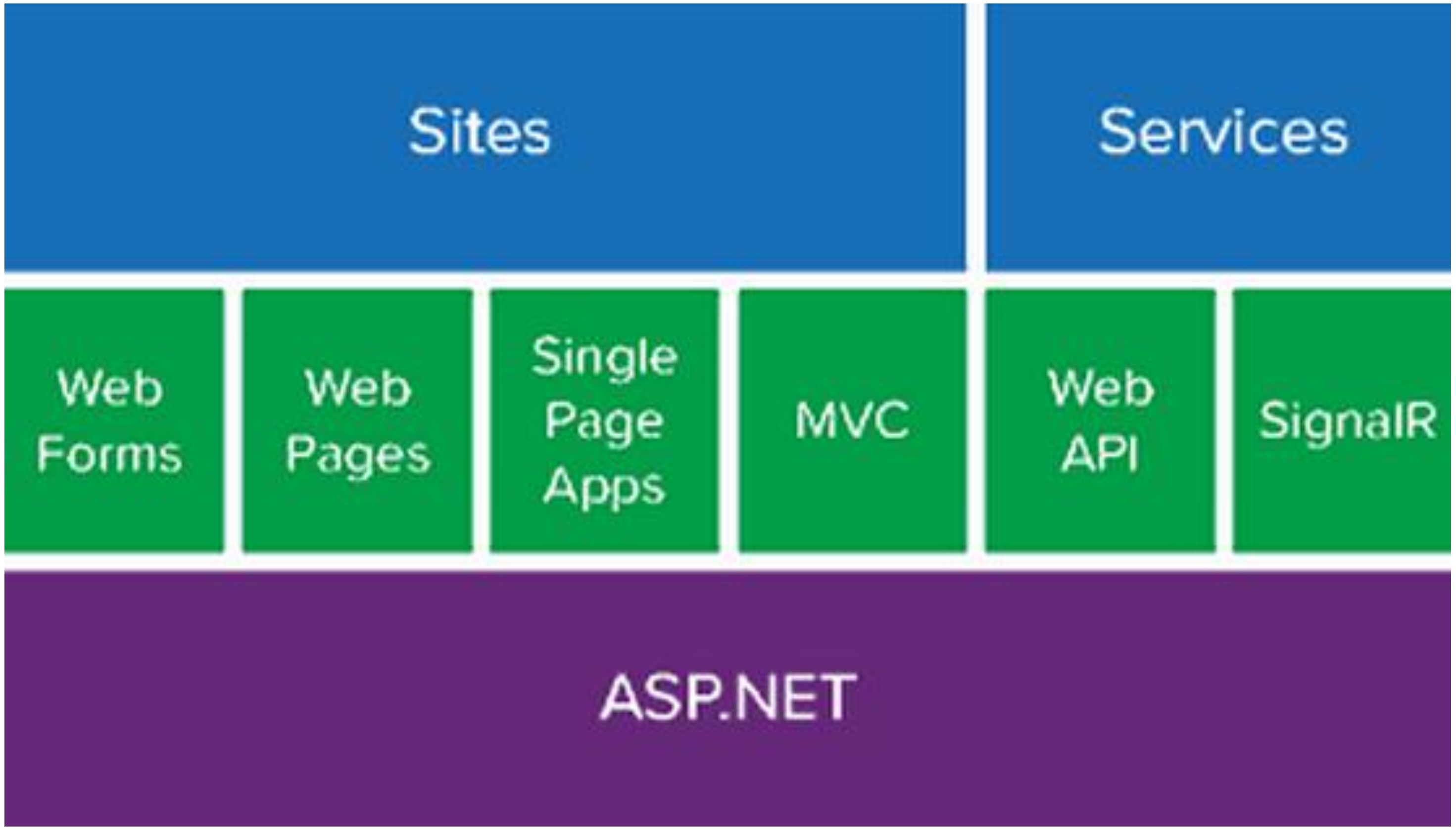
Technologies

- Current Client Technologies
  - Scripting details vary from browser to browser
  - HTML5 and CSS3 support varies but is pretty good
- Current Server Technologies
  - Apache
  - Oracle / Sun (doing ASP.NET, CGI and PHP)
  - Microsoft and IIS

ASP.NET (History)

- Introduced with the .NET Framework in 2002
  - It made Web development look like traditional Windows development
- ASP.NET 2.0 introduced in 2005 gave developers enhanced database tools
  - This included desktop improvements too
  - MVC was introduced
- ASP.NET 3.0 released with 2008

ASP.NET (Architecture)



What is a .NET Application

- It looks and works surprisingly like Java
- Source code (VB, C#, etc.) is compiled into a machine independent assembly language (MSIL or IL)
  - The executable file is called an assembly
  - IL can be disassembled using the IL Disassembler (ILDASM)

What is a .NET Application

- MSIL is translated into native executable code via the Just-in-Time (JIT) compiler
- All .NET applications run under the control of the Common Language Runtime (CLR)
- On the desktop, the application runs under the control of the Common Language Runtime
- With Respect to Web applications, all execution is handled on the server (IIS)

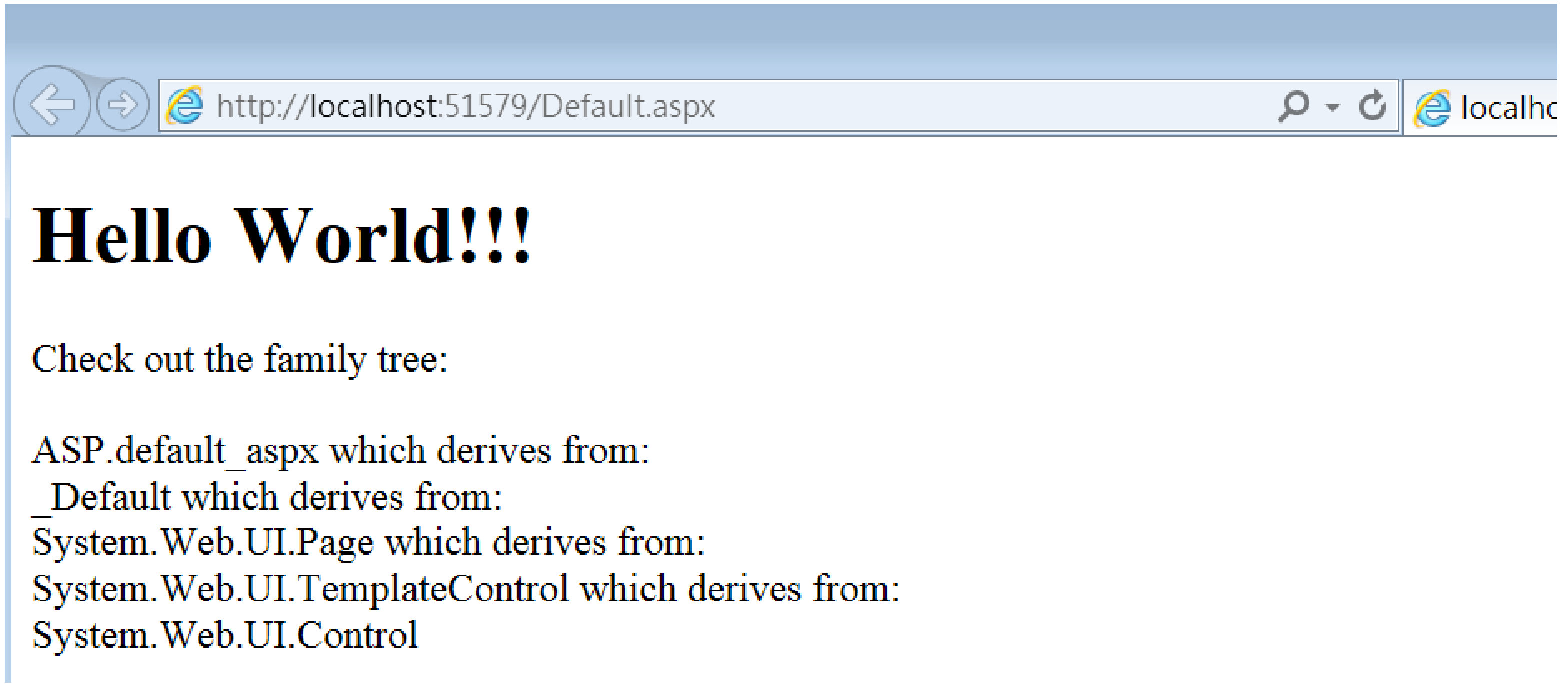
The ASP.NET Runtime

- When IIS sees an .aspx file, it loads aspnet_isapi.dll to create the Page object
- Application behavior is controlled by configuration files
  - Machine.config is the global machine configuration file
  - Web.config allows us to override default configuration items

Example: ASP.NET

```
<%@ Page Language="C#" CodeFile="Default.cs"
                            Inherits="_Default" %>
<html>
  <body>
    <h1>Hello World!!!</h1>
    <%
          // This block will execute
          // in the RenderControl method
          ShowLineage();
    %>
  </body>
</html>
```

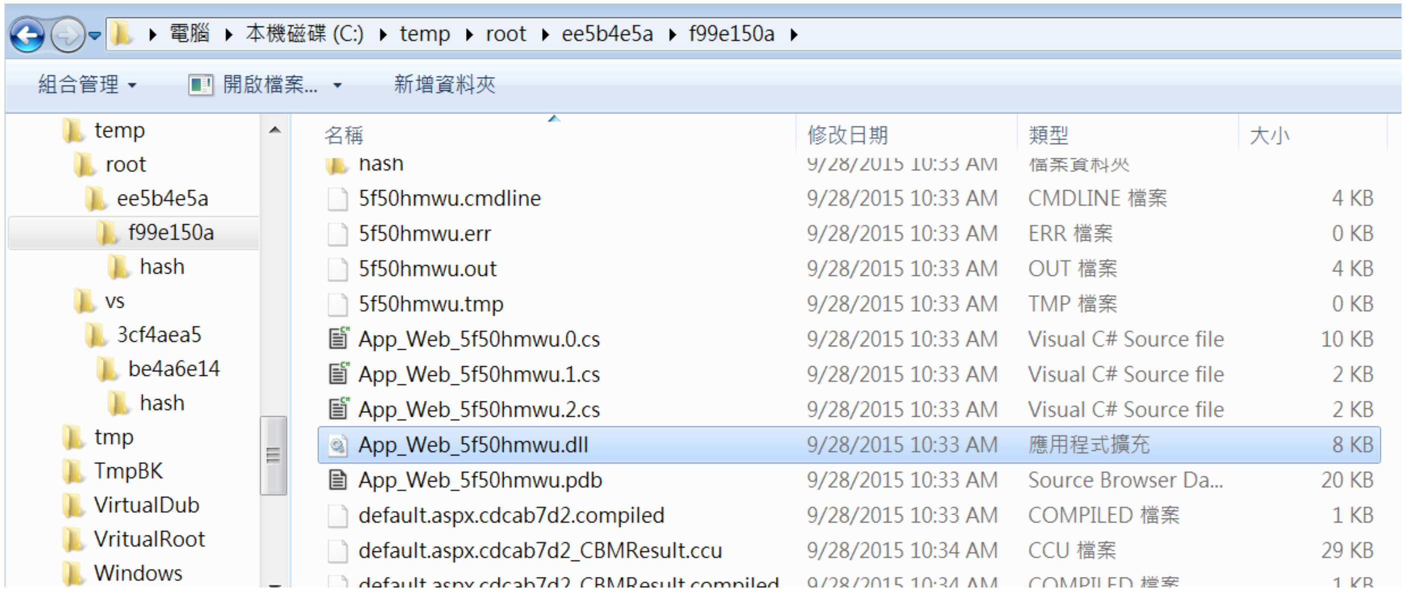
```
using System.Web;
public partial class _Default : System.Web.UI.Page
{
   public void ShowLineage()
   {
     Response.Write("Check out the family tree: <br/> <br/>");
     Response.Write(this.GetType().ToString());
     Response.Write(" which derives from: <br/> ");
     Response.Write(this.GetType().BaseType.ToString());
     Response.Write(" which derives from: <br/> ");
     Response.Write(
        this.GetType().BaseType.BaseType.ToString());
     Response.Write(" which derives from: <br/> ");
     Response.Write(
        this.GetType().BaseType.BaseType.BaseType.ToString());
     Response.Write(" which derives from: <br/> ");
     Response.Write(
this.GetType().BaseType.BaseType.BaseType.BaseType.ToString());
    }
});
```



Hello World!!!

Check out the family tree:

ASP.default_aspx which derives from:
_Default which derives from:
System.Web.UI.Page which derives from:
System.Web.UI.TemplateControl which derives from:
System.Web.UI.Control



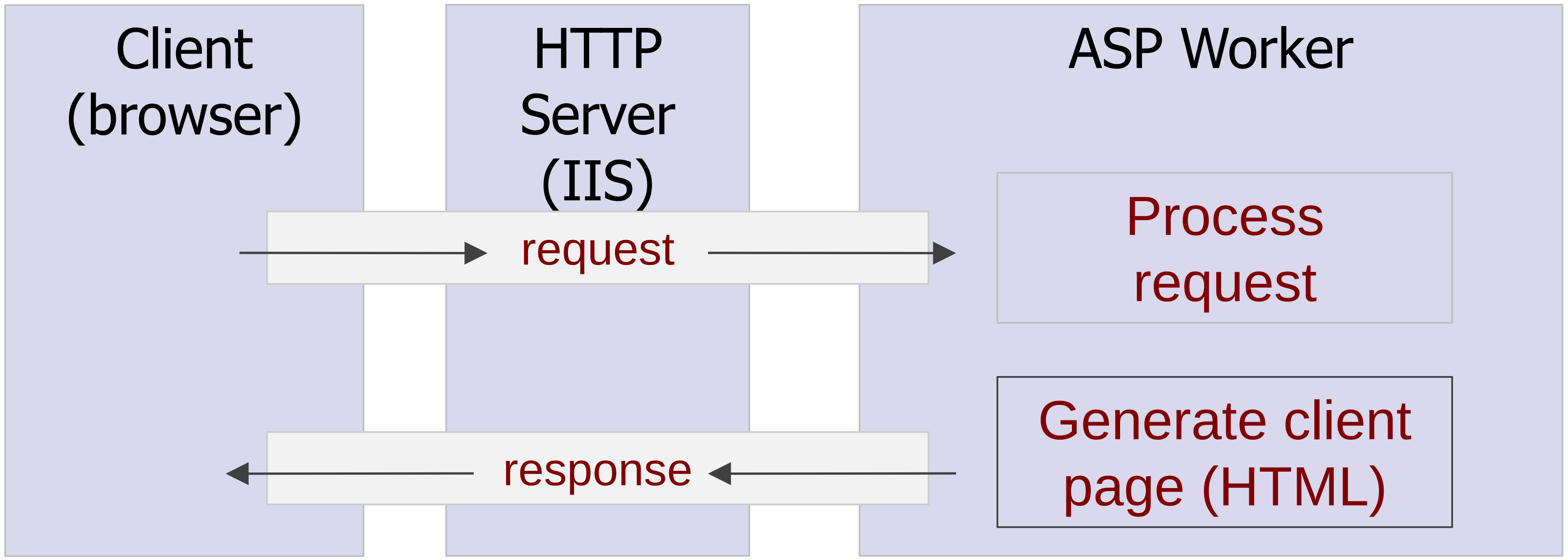
名稱	修改日期	類型	大小
hash	9/28/2015 10:33 AM	檔案資料夾	
5f50hmwu.cmdline	9/28/2015 10:33 AM	CMDLINE 檔案	4 KB
5f50hmwu.err	9/28/2015 10:33 AM	ERR 檔案	0 KB
5f50hmwu.out	9/28/2015 10:33 AM	OUT 檔案	4 KB
5f50hmwu.tmp	9/28/2015 10:33 AM	TMP 檔案	0 KB
App_Web_5f50hmwu.0.cs	9/28/2015 10:33 AM	Visual C# Source file	10 KB
App_Web_5f50hmwu.1.cs	9/28/2015 10:33 AM	Visual C# Source file	2 KB
App_Web_5f50hmwu.2.cs	9/28/2015 10:33 AM	Visual C# Source file	2 KB
App_Web_5f50hmwu.dll	9/28/2015 10:33 AM	應用程式擴充	8 KB
App_Web_5f50hmwu.pdb	9/28/2015 10:33 AM	Source Browser Da...	20 KB
default.aspx.cdcab7d2.compiled	9/28/2015 10:33 AM	COMPILED 檔案	1 KB
default.aspx.cdcab7d2_CBMResult.ccu	9/28/2015 10:34 AM	CCU 檔案	29 KB
default.aspx.cdcab7d2_CBMResult.compiled	9/28/2015 10:34 AM	COMPILED 檔案	1 KB

```
<compilation debug="true" tempDirectory="c:\temp\"/>
```

ASP.NET Web Forms

- Provides server-side tools for site navigation
  - Menus, site-map, multi-view, etc.
- Database connections
- Client and server executable code
- State management and caching
- Robust development tools (Visual Studio)

ASP.NET Web Forms



Web Forms and Web Programming

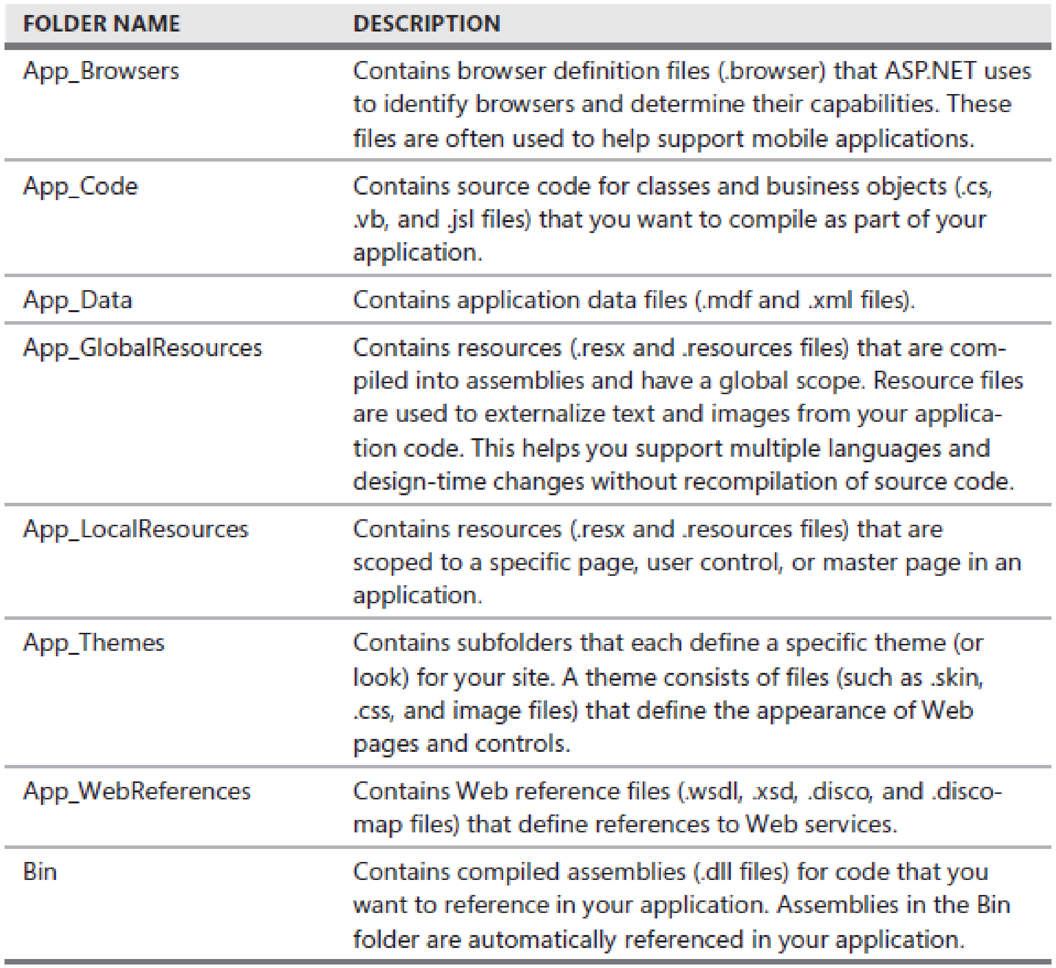
- Remember that the Web is stateless
  - Web servers to not persist state from one request to the next
- ASP.NET does allow stateful programming though
  - various hidden fields and serialized data
  - server side session objects
- Event-Driven Programming
  - ASP.NET, though serialization, simulates event-driven programming
  - ASP.NET events differ significantly from Windows Forms events

Web Forms Component Model

- Remember that .NET is an OOP environment and ASP.NET is no exception
  - As the developer, we create .aspx pages that make up a Web application
- When executed, the .aspx page is compiled into a custom class that inherits from System.Web.UI.Page
- As the developer, we can program against this class just as we would program against any class
- It all happens on the server!

ASP.Net Web Site Structure

- Major file types
  - Web form (.aspx file): a major form of ASP.Net pages
  - Class file (.cs file): user written class source file
  - User control (.ascx): web form (page) components to be included in a container page.
  - Web.config: Web site configuration file under the root directory of the web site
  - Other common web content files: HTML, image, script, style, etc.



FOLDER NAME	DESCRIPTION
App_Browsers	Contains browser definition files (.browser) that ASP.NET uses to identify browsers and determine their capabilities. These files are often used to help support mobile applications.
App_Code	Contains source code for classes and business objects (.cs, .vb, and .jsl files) that you want to compile as part of your application.
App_Data	Contains application data files (.mdf and .xml files).
App_GlobalResources	Contains resources (.resx and .resources files) that are compiled into assemblies and have a global scope. Resource files are used to externalize text and images from your application code. This helps you support multiple languages and design-time changes without recompilation of source code.
App_LocalResources	Contains resources (.resx and .resources files) that are scoped to a specific page, user control, or master page in an application.
App_Themes	Contains subfolders that each define a specific theme (or look) for your site. A theme consists of files (such as .skin, .css, and image files) that define the appearance of Web pages and controls.
App_WebReferences	Contains Web reference files (.wsdl, .xsd, .disco, and .discomap files) that define references to Web services.
Bin	Contains compiled assemblies (.dll files) for code that you want to reference in your application. Assemblies in the Bin folder are automatically referenced in your application.

Creating a Web Form

- Two web form programming model
  - Single-File Page (inline code): a single web page file contains both server-side code and HTML.
  - Code-Behind Page: this model physically separates the user interface layout markup and the server-side code into two distinct files.
  - A .aspx page contains the layout markup (HTML)
  - A related .aspx.cs file contains the associated code.
- Hybrid model
  - In code behind model, code blocks can also be place directly into the .aspx file.

ASPX Page Structure

- Each page is effectively a class that inherits from the "System.Web.UI.Page" class
- Each aspx page consists of general web page content (HTML, JavaScript, Style, Head, Body, etc.) and ASP.Net code blocks
- Embedded code blocks in the layout page (.aspx)
  - <%@ Page %>: the page directive is used to the environment, specifying how the page should be processed.
  - <script runat="server"> </server>: defining class level variables, functions and methods.
  - <% %>: for normal procedure code that are usually within a method; can be mixed with normal web page content, such as HTML and JavaScript
  - <%= %>: expression for a single value output; equivalent to Response.Write(); can be mixed with normal web page content, such as HTML and JavaScript
  - <%----%>: comment.

Code Behind Example

- “schedule.aspx” has this line of code at the top of the page

The code file is defined in the page directive.

```
<%@ Page Language="C#" CodeFile="schedule.aspx.cs"
     Inherits=“schedule" %>
```

Each page inherits from the “System.Web.UI.Page” class

- schedule.aspx.cs

```
public partial class schedule : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgse)
   {
     // code goes here;
   }
   // other class level variables, methods or functions go
here.
}
```

In this method, all code will be executed when the web page is loaded.

Web Development in Visual Studio

Introduction to Visual Studio

- All ASP.NET development and testing can take place inside of Visual Studio
- A local Web server is provided for testing purposes
- It’s part of Visual Studio
- Intellisense works for server code and JavaScript, XHTML, and HTML5

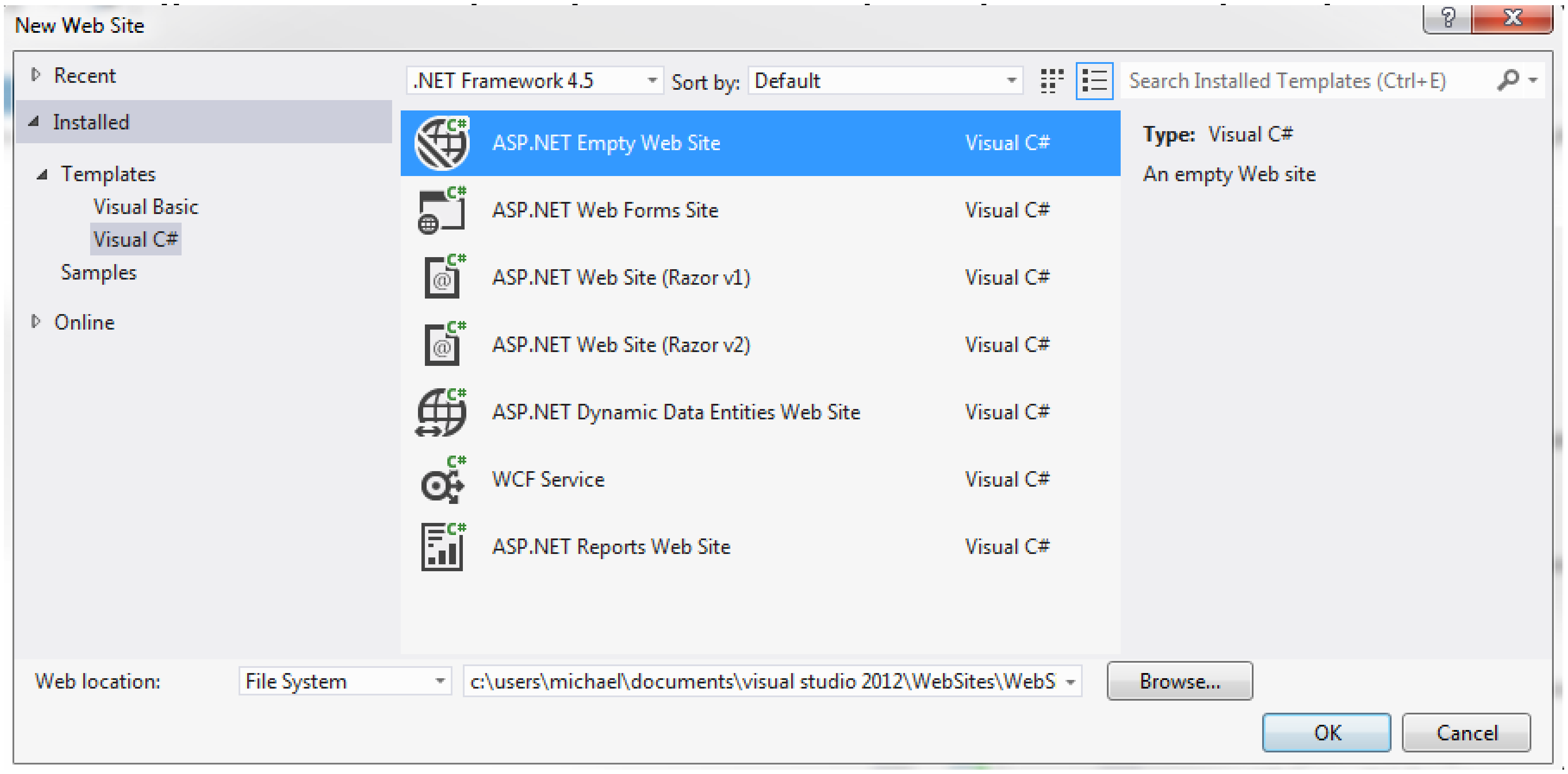
Visual Studio Windows

- The Toolbox contains HTML controls and ASP.NET controls
  - Drag control instances using Design or Source view
- The Solution Explorer lists the application’s files
- The Properties window is used to set object properties

Creating an ASP.NET Application

- Click **File / New / Web Site**
- In the New Web Site dialog box
  - Select ASP.NET Web site
  - Select the desired folder (The folder should be empty or not already exist)
  - Note that different framework versions are supported

New Web Site Dialog Box (Illustration)


New Web Site

.NET Framework 4.5 Sort by: Default

Search Installed Templates (Ctrl+E)

ASP.NET Empty Web Site	Visual C#
ASP.NET Web Forms Site	Visual C#
ASP.NET Web Site (Razor v1)	Visual C#
ASP.NET Web Site (Razor v2)	Visual C#
ASP.NET Dynamic Data Entities Web Site	Visual C#
WCF Service	Visual C#
ASP.NET Reports Web Site	Visual C#

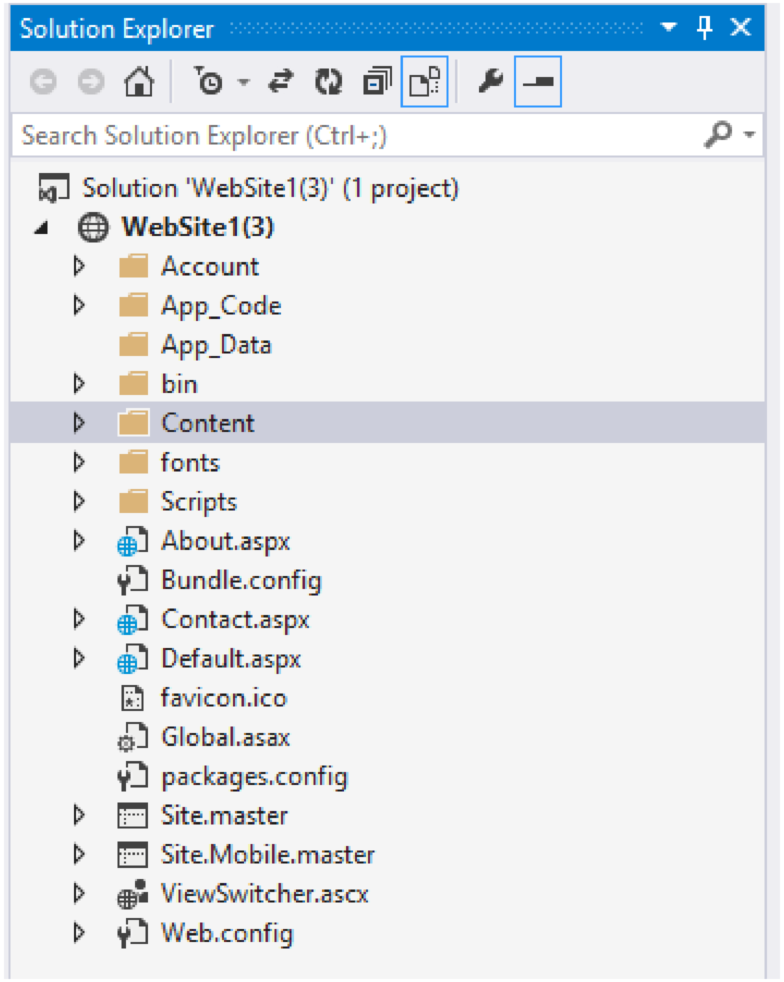
Type: Visual C#

An empty Web site

Web location: File System c:\users\michael\documents\visual studio 2012\WebSites\WebS

Solution Explorer Files

- Web forms (.aspx) make up the visual forms
  - They are HTML5 documents
  - There is a code behind file (.aspx.cs)
- Web.config contains configuration information
  - It's an XML document

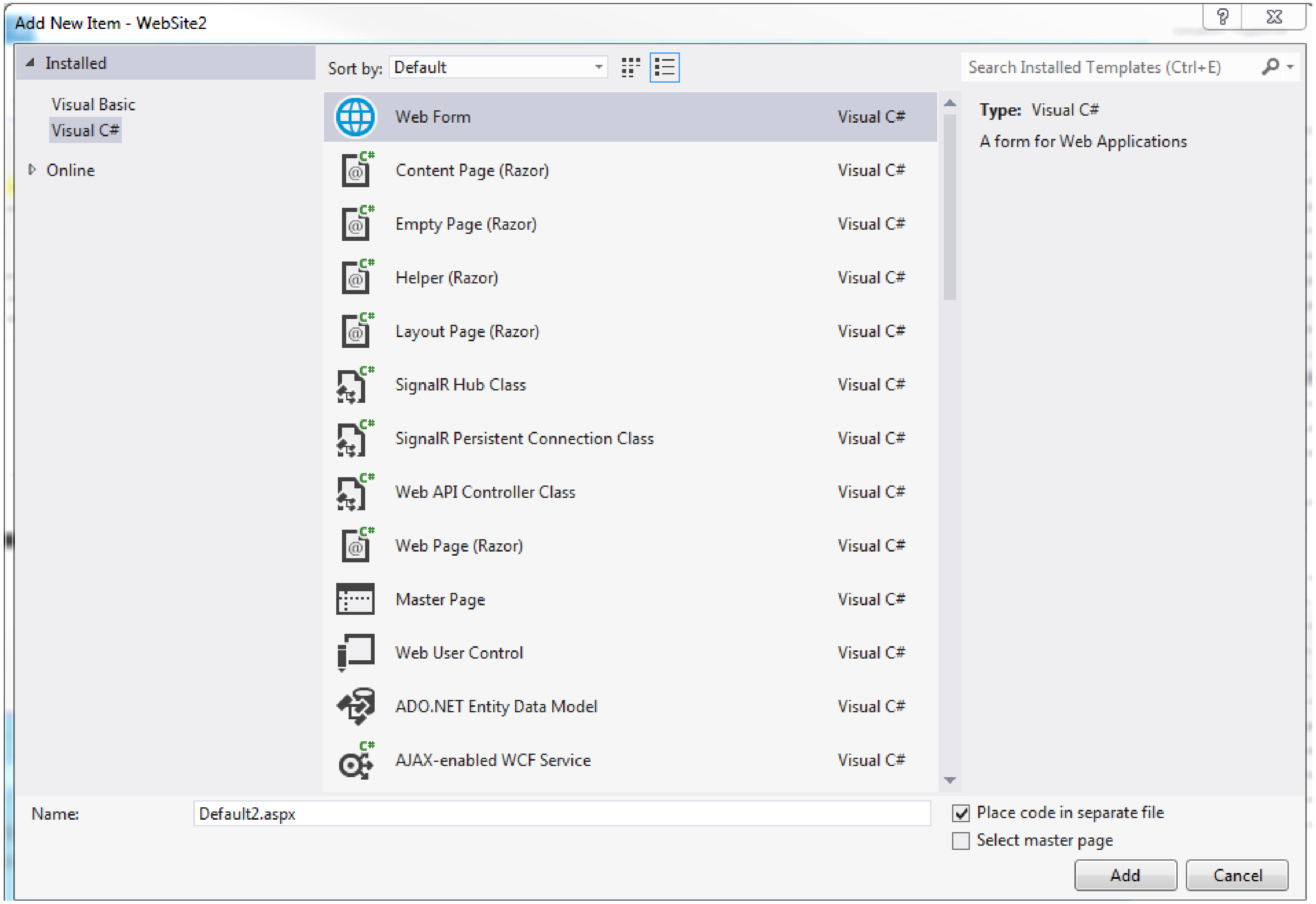

Solution Explorer
Search Solution Explorer (Ctrl+;)

```
Solution 'WebSite1(3)' (1 project)
◢ WebSite1(3)
    ▷ Account
    ▷ App_Code
      App_Data
    ▷ bin
    ▷ Content
    ▷ fonts
    ▷ Scripts
    ▷ About.aspx
      Bundle.config
    ▷ Contact.aspx
    ▷ Default.aspx
      favicon.ico
      Global.asax
      packages.config
    ▷ Site.master
    ▷ Site.Mobile.master
    ▷ ViewSwitcher.ascx
    ▷ Web.config
```

Adding a Web Form

- Every .aspx file corresponds to a Web page
- Click Website / Add New Item
  - Select Web Form
  - Select the master page, if desired
  - Set the name as desired

Adding a Web Form



Add New Item - WebSite2

Installed

Visual Basic

Visual C#

Online

Sort by: Default

Search Installed Templates (Ctrl+E)

Web Form	Visual C#
Content Page (Razor)	Visual C#
Empty Page (Razor)	Visual C#
Helper (Razor)	Visual C#
Layout Page (Razor)	Visual C#
SignalR Hub Class	Visual C#
SignalR Persistent Connection Class	Visual C#
Web API Controller Class	Visual C#
Web Page (Razor)	Visual C#
Master Page	Visual C#
Web User Control	Visual C#
ADO.NET Entity Data Model	Visual C#
AJAX-enabled WCF Service	Visual C#

Type: Visual C#

A form for Web Applications

Name: Default2.aspx

[✓] Place code in separate file

[ ] Select master page

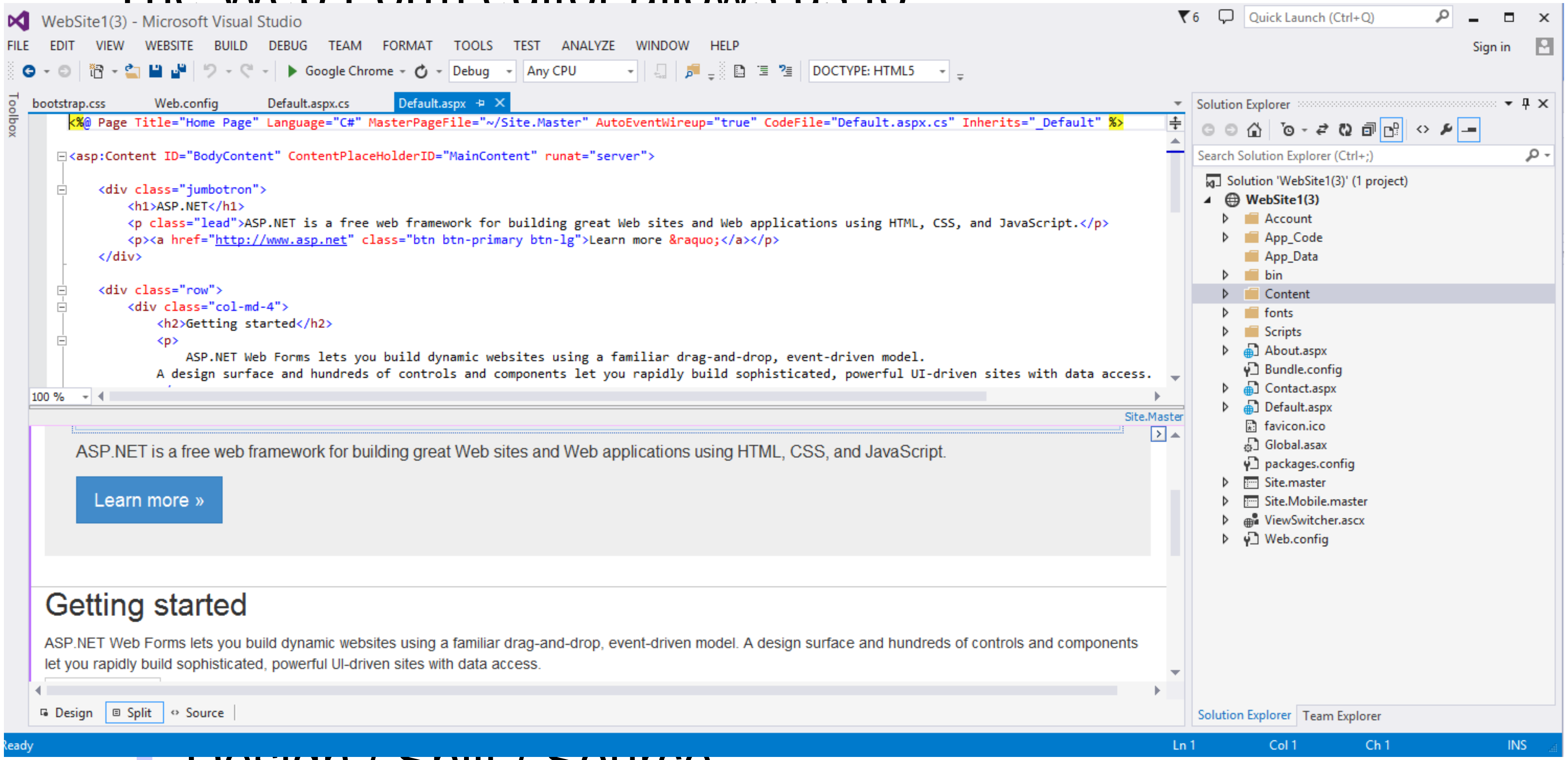
Add Cancel

The Web Form Editor

- The Web Form editor allows us to
  - Create HTML5
  - Create ASP.NET server side tags
  - Create the “code behind” the form in a language such as Visual Basic or C#
- The Web form editor has three views
  - Design / Split / Source

The Web Form Editor

- The Web Form editor allows us to



WebSite1(3) - Microsoft Visual Studio
FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM FORMAT TOOLS TEST ANALYZE WINDOW HELP
Google Chrome Debug Any CPU DOCTYPE: HTML5
bootstrap.css Web.config Default.aspx.cs Default.aspx
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'WebSite1(3)' (1 project)
WebSite1(3)
Account
App_Code
App_Data
bin
Content
fonts
Scripts
About.aspx
Bundle.config
Contact.aspx
Default.aspx
favicon.ico
Global.asax
packages.config
Site.master
Site.Mobile.master
ViewSwitcher.ascx
Web.config
Solution Explorer Team Explorer
Design Split Source
Ready Ln 1 Col 1 Ch 1 INS

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

    <div class="jumbotron">
        <h1>ASP.NET</h1>
        <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.</p>
        <p><a href="http://www.asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>
    </div>

    <div class="row">
        <div class="col-md-4">
            <h2>Getting started</h2>
            <p>
                ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model.
            A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.
```

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Learn more »

Getting started

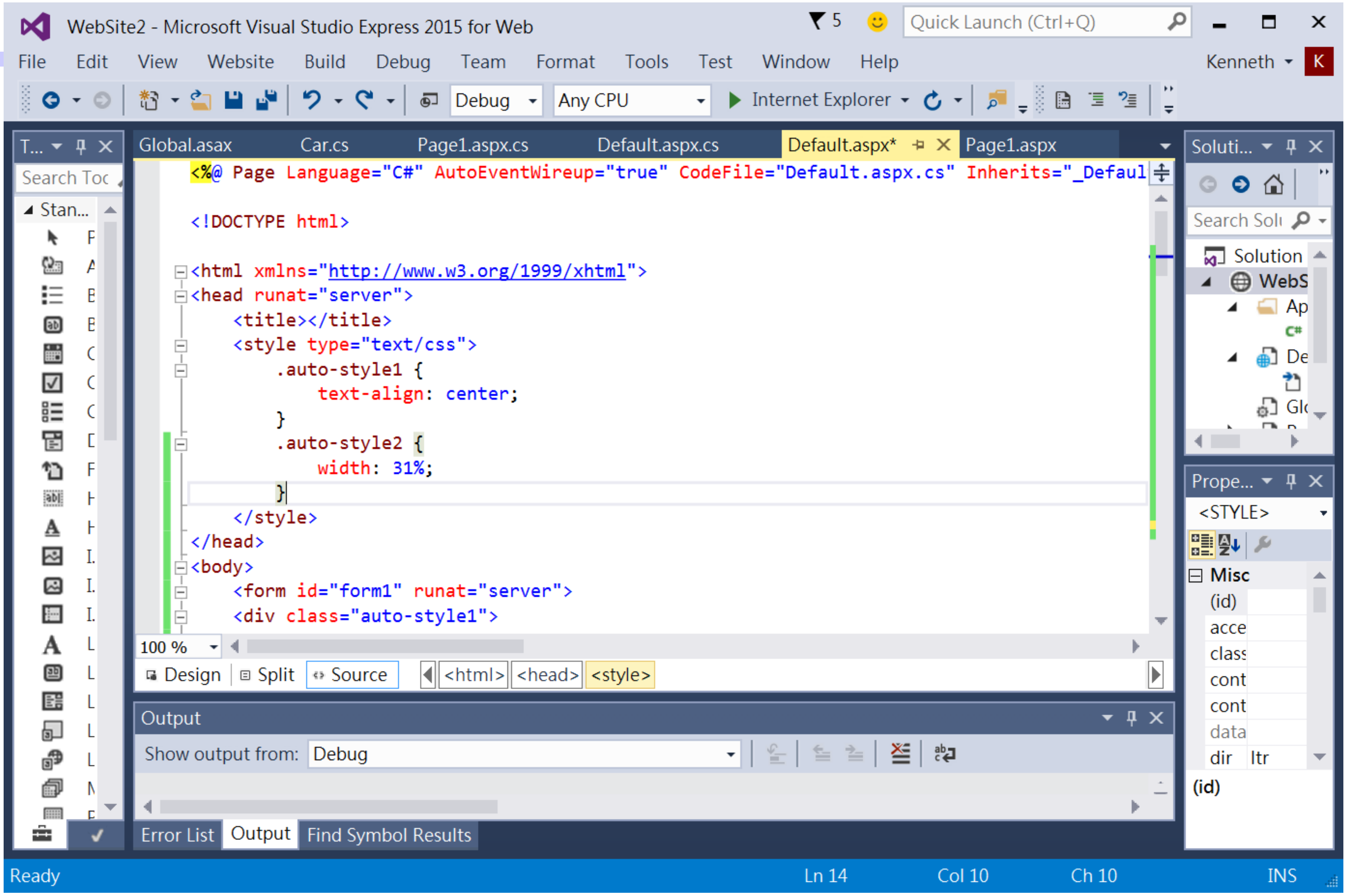
ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

- Design / Split / Source

Default Web Page Code (HTML)

- A Web Form contains a @Page directive
  - More later but it’s this directive that makes the page a Web form
  - The remaining HTML should look familiar to us
  - Note there is exactly one <form> tag
  - ASP.NET controls must appear in the <form>

Default Web Page Code (HTML)



WebSite2 - Microsoft Visual Studio Express 2015 for Web
File Edit View Website Build Debug Team Format Tools Test Window Help
Debug
Any CPU
Internet Explorer
Global.asax Car.cs Page1.aspx.cs Default.aspx.cs Default.aspx* Page1.aspx
Design Split Source
<html> <head> <style>
Output
Show output from: Debug
Error List Output Find Symbol Results
Ready Ln 14 Col 10 Ch 10 INS

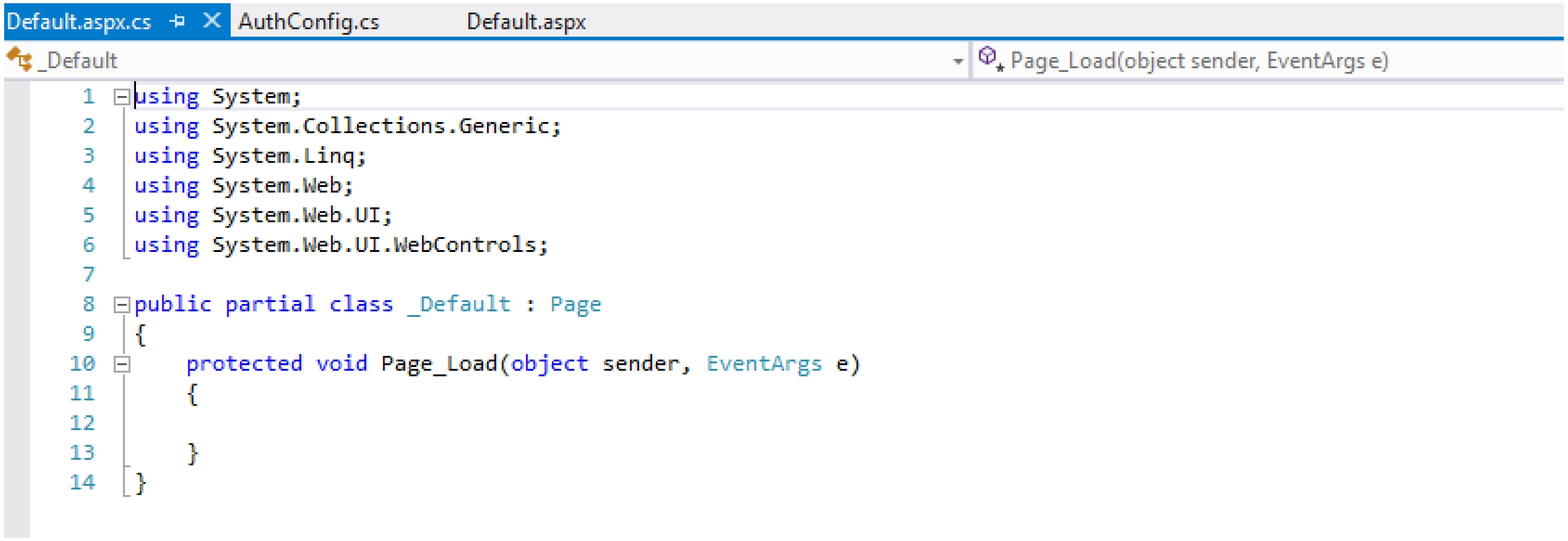
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Defaul

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            text-align: center;
        }
        .auto-style2 {
            width: 31%;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div class="auto-style1">
```

The Code Editor

- An ASP.NET Web page is made up of two files
  - The .aspx file contains the HTML
  - The .aspx.cs file contains the VB or C# code
- It works the same way as the Code Editor for desktop applications
  - The event names differ a bit
  - **`Page_Load`** instead of **`Form_Load`** for example

The Code Editor

- An ASP.NET Web page is made up of two files



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

  - The event names differ a bit
  - **`Page_Load`** instead of **`Form_Load`** for example

Toolbox Controls

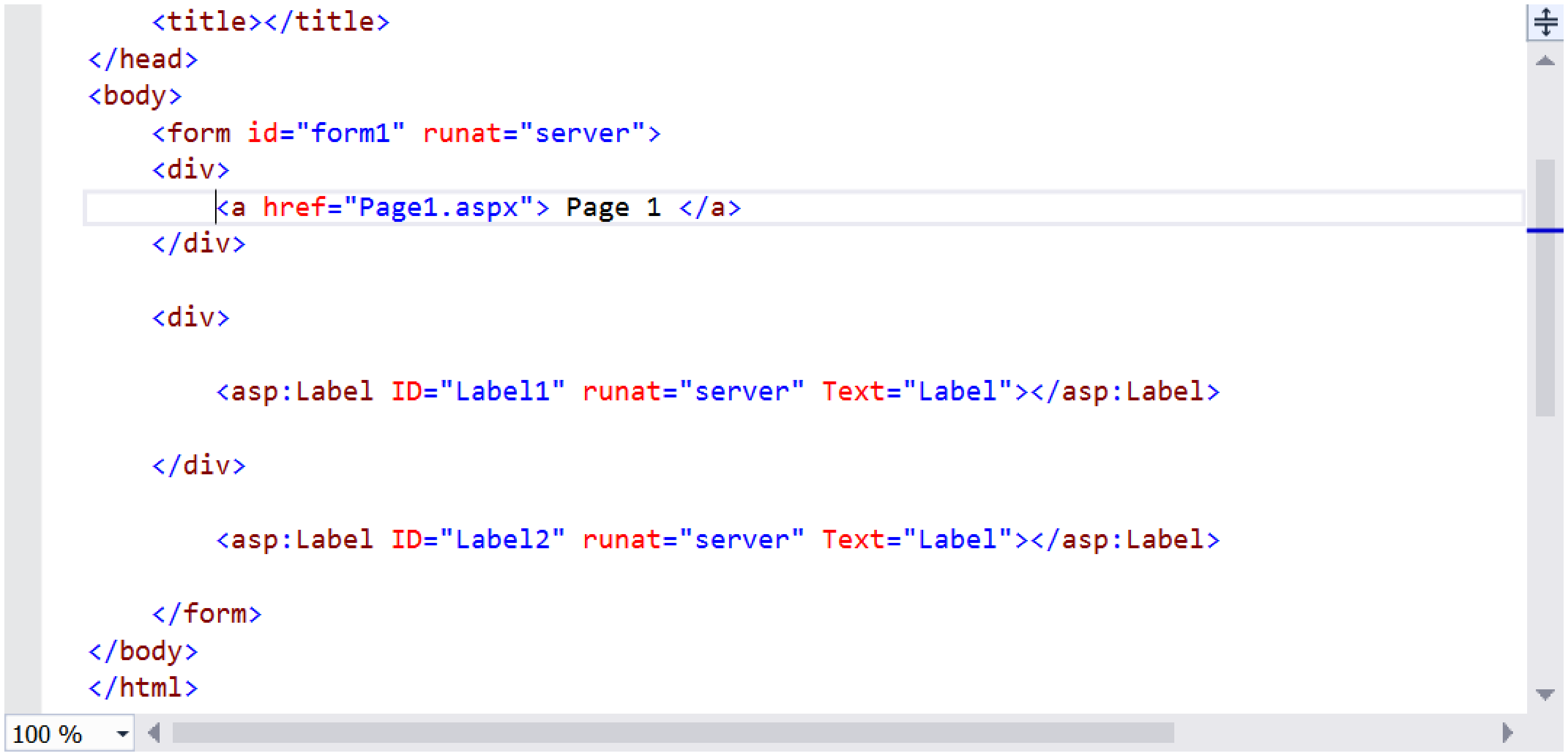
- They are different controls than those used for desktop applications
- HTML controls are standard HTML controls that will work with any browser
- Other controls are server side controls that only work with IIS and ASP

Using Controls

- Using the Toolbox, drag the desired control to the design surface
  - We can drag to the design surface or the HTML editor
- Set properties using the Properties window
- Create event handlers using the Properties window

ASP.NET Controls

- ASP controls have an <asp:> prefix. These are actually XML namespaces



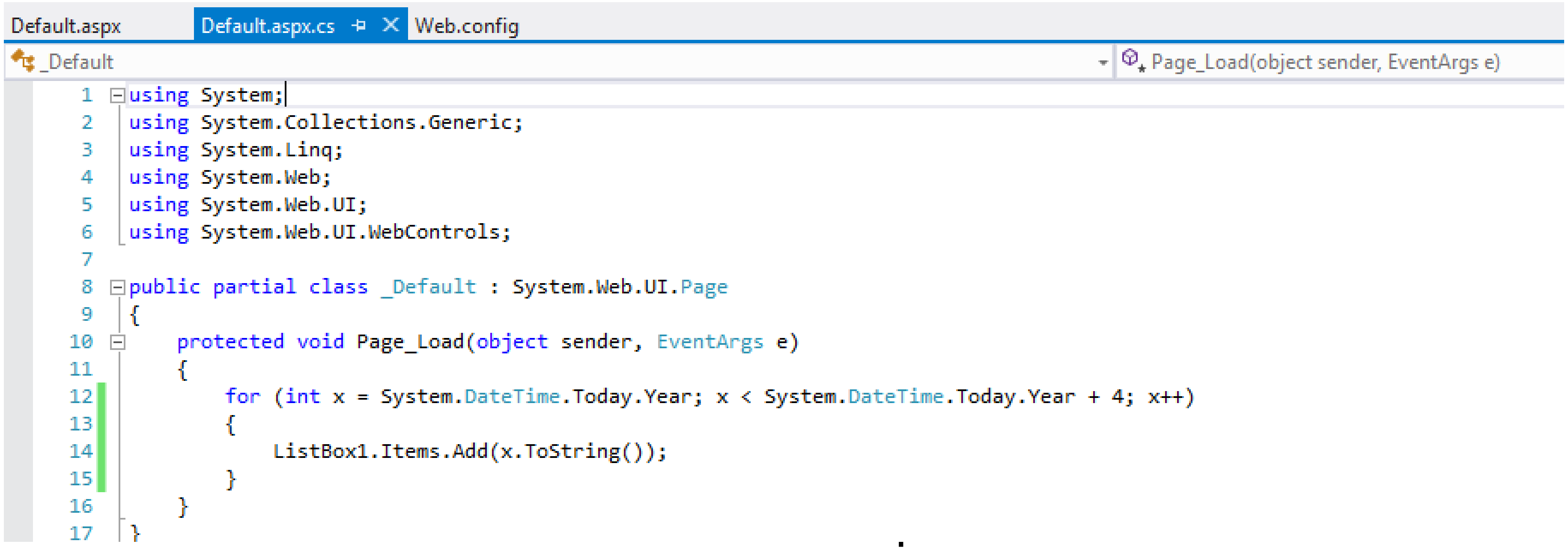
```
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
        <div>
            <a href="Page1.aspx"> Page 1 </a>
        </div>

        <div>

            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

        </div>

            <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>

        </form>
    </body>
    </html>
```

Creating Event Handlers

- ASP.NET control events vary significantly from desktop events
  - There are no server side focus events (Why?)
- There are a couple of ways to create event handlers
  - The easiest way is to select Design View
  - Then double-click the control to create the “default” event handler
  - Or use the event tab on the Properties window
  - Or create it by hand

Example: Event Handler

- ASP.NET control events vary significantly from desktop events



Default.aspx | Default.aspx.cs | Web.config

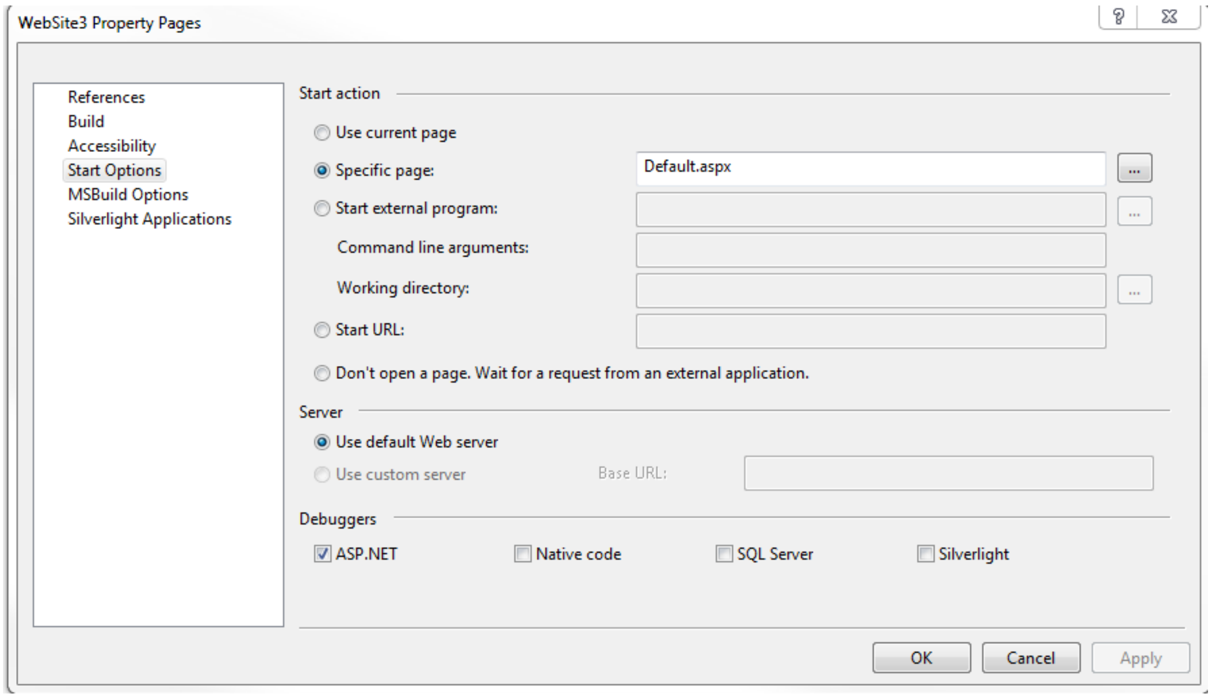
_Default | Page_Load(object sender, EventArgs e)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        for (int x = System.DateTime.Today.Year; x < System.DateTime.Today.Year + 4; x++)
        {
            ListBox1.Items.Add(x.ToString());
        }
    }
}
```

- Or create it by hand

Setting the Start Page

- Every Web site has exactly one page designated as the start (default) page
- To set the start page, right-click the page in the Solution Explorer - Click Set as Start Page



WebSite3 Property Pages

References
Build
Accessibility
Start Options
MSBuild Options
Silverlight Applications

Start action

- Use current page
- [✓ Specific page:] Default.aspx
- Start external program:
  - Command line arguments:
  - Working directory:
- Start URL:
- Don't open a page. Wait for a request from an external application.

Server

- [✓ Use default Web server]
- Use custom server — Base URL:

Debuggers

[✓ ASP.NET] Native code SQL Server Silverlight

OK Cancel Apply

Running the Application

- It’s possible to run an ASP.NET application from Visual Studio
  - There is a “test” Web server built in
- Just press F5 to run the application
  - It appears in the default Web browser

Debugging

- Debugging windows work the same way in desktop and ASP.NET applications
  - Set breakpoints on executable code lines
  - Print values to the Immediate window
  - Use Watch windows to interrogate the values of variables

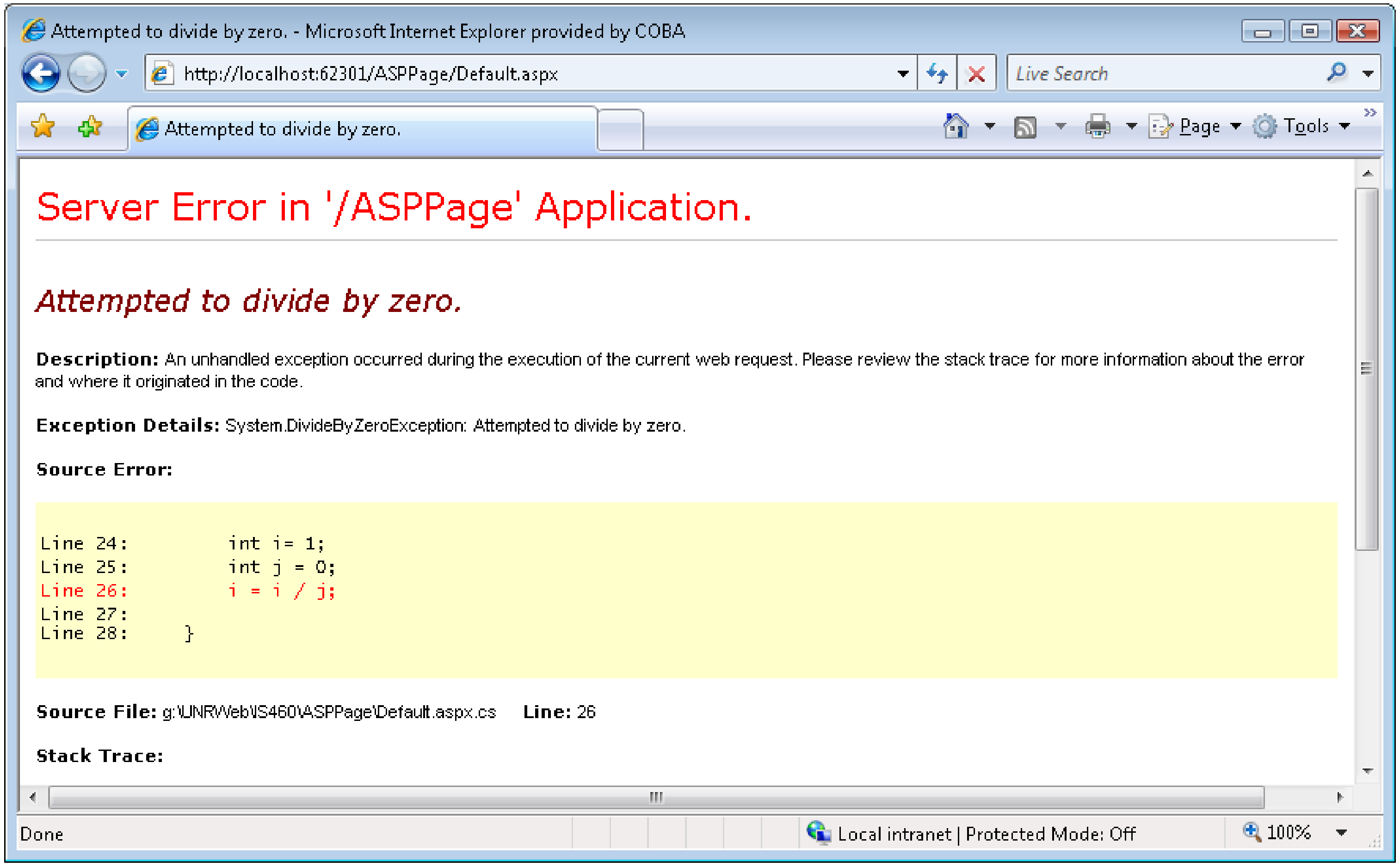
Introduction to Error Handling

- Just like desktop applications, exceptions are thrown in various cases
  - Divide by 0, IO, Database, etc.
- Unhandled, we call these the yellow page of death
- What happens depends on the following Web.config setting:

```
<compilation debug="false">
or
<compilation debug="true">
```

Introduction to Error Handling

```
<compilation debug="true">
```


Attempted to divide by zero. - Microsoft Internet Explorer provided by COBA
http://localhost:62301/ASPPage/Default.aspx
Live Search
Attempted to divide by zero.
Page
Tools
Done
Local intranet | Protected Mode: Off
100%

Server Error in '/ASPPage' Application.

Attempted to divide by zero.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.DivideByZeroException: Attempted to divide by zero.

Source Error:

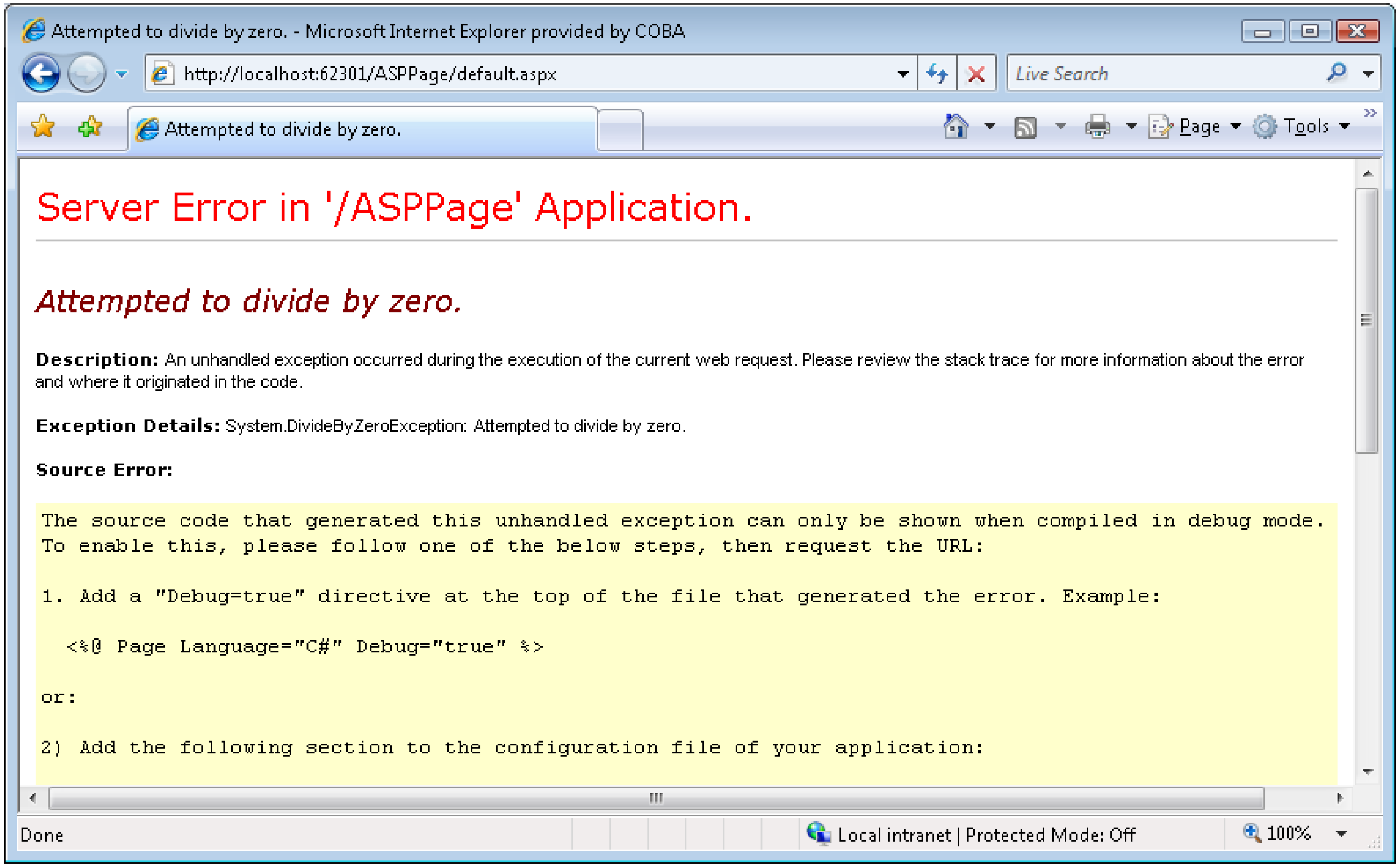
```
Line 24:         int i= 1;
Line 25:         int j = 0;
Line 26:         i = i / j;
Line 27:
Line 28:     }
```

Source File: g:\UNRWeb\IS460\ASPPage\Default.aspx.cs   **Line:** 26

Stack Trace:

Introduction to Error Handling

```
<compilation debug="false">
```


Attempted to divide by zero. - Microsoft Internet Explorer provided by COBA
http://localhost:62301/ASPPage/default.aspx
Live Search
Attempted to divide by zero.
Page
Tools
Done
Local intranet | Protected Mode: Off
100%

Server Error in '/ASPPage' Application.

Attempted to divide by zero.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.DivideByZeroException: Attempted to divide by zero.

Source Error:

```
The source code that generated this unhandled exception can only be shown when compiled in debug mode.
To enable this, please follow one of the below steps, then request the URL:

1. Add a "Debug=true" directive at the top of the file that generated the error. Example:

  <%@ Page Language="C#" Debug="true" %>

or:

2) Add the following section to the configuration file of your application:
```

Handling Errors (Introduction)

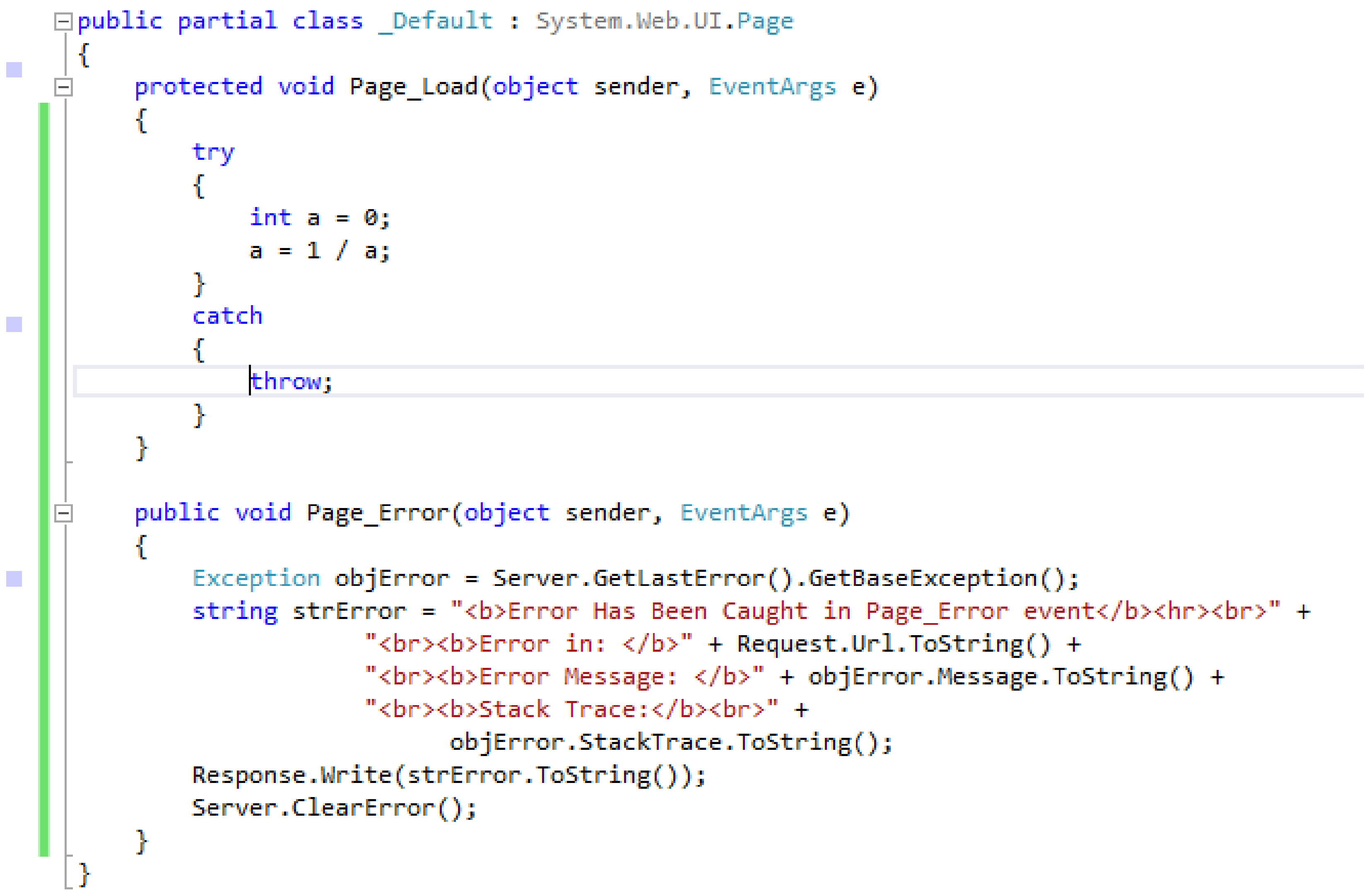
- Exception handlers are created (as usual) using **`try`**, **`catch`** blocks
- The **`Page_Error`** event handler gives you a global exception handler for the page
- The **`Application_Error`** event applies to the entire application itself

Handling Errors (Introduction)

- Use **`Server.GetLastError`**() to get information about

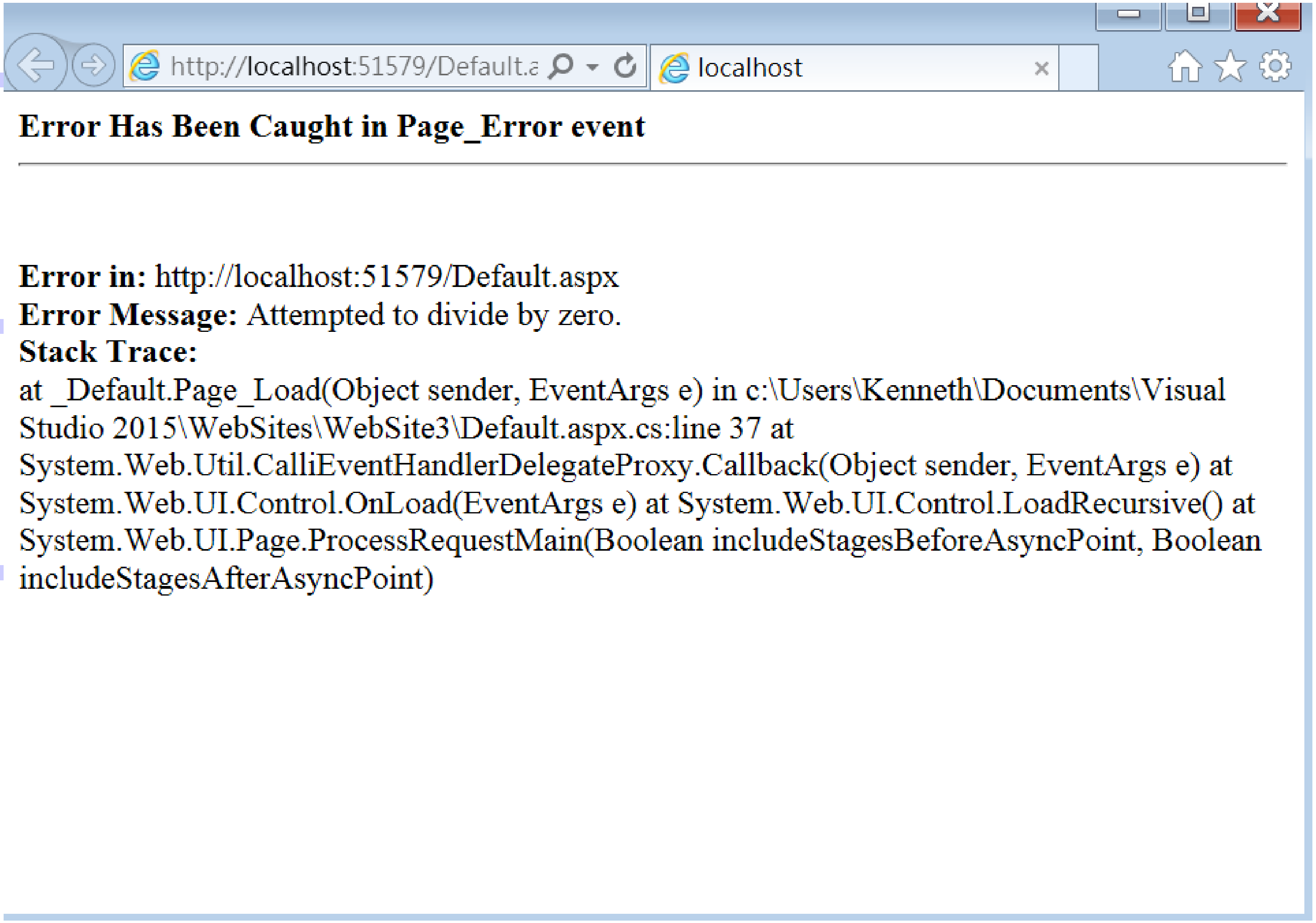
  the most recent error
- Use **`Server.ClearError`**() to clear the exception
- Create other exception handlers, as necessary

Handling Errors (Introduction)



```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            int a = 0;
            a = 1 / a;
        }
        catch
        {
            throw;
        }
    }

    public void Page_Error(object sender, EventArgs e)
    {
        Exception objError = Server.GetLastError().GetBaseException();
        string strError = "<b>Error Has Been Caught in Page_Error event</b><hr><br>" +
                    "<br><b>Error in: </b>" + Request.Url.ToString() +
                    "<br><b>Error Message: </b>" + objError.Message.ToString() +
                    "<br><b>Stack Trace:</b><br>" +
                          objError.StackTrace.ToString();
        Response.Write(strError.ToString());
        Server.ClearError();
    }
}
```

Handling Errors (Introduction)



Error Has Been Caught in Page_Error event

Error in: http://localhost:51579/Default.aspx
Error Message: Attempted to divide by zero.
Stack Trace:
at _Default.Page_Load(Object sender, EventArgs e) in c:\Users\Kenneth\Documents\Visual Studio 2015\WebSites\WebSite3\Default.aspx.cs:line 37 at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) at System.Web.UI.Control.OnLoad(EventArgs e) at System.Web.UI.Control.LoadRecursive() at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)

Tracing (Introduction)

- It’s possible to trace execution by enabling tracing in the Web.config file
- @Page directives can also be used
- Example:

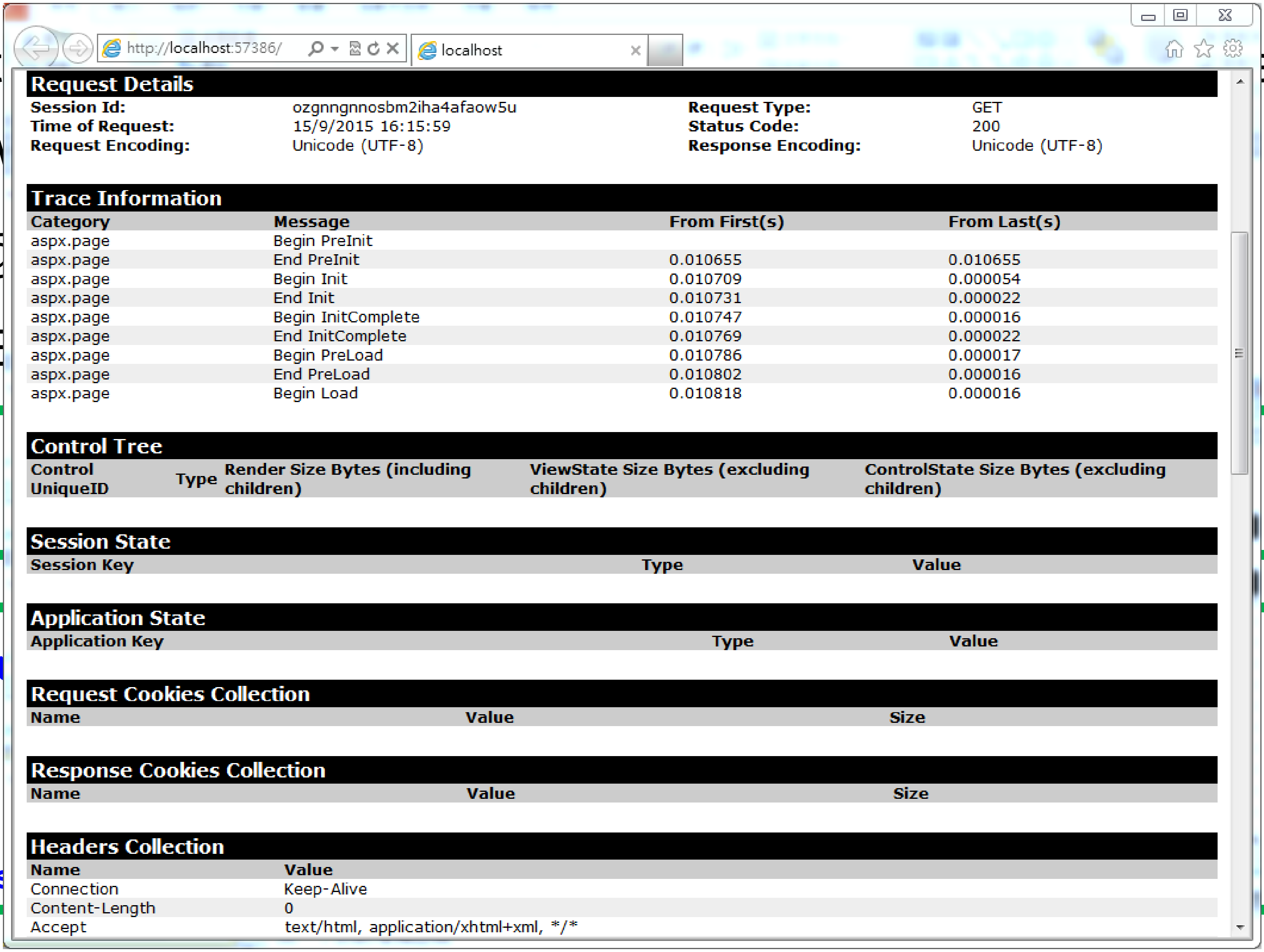
```
<trace enabled="true" pageOutput="true"/>
```

```
<system.web>
     <compilation debug="true" targetFramework="4.5.2" />
     <httpRuntime targetFramework="4.5.2" />
     <trace enabled="true" pageOutput="true"/>
</system.web>
```

Tracing (Introduction)

- It
- W
- @
- E

```
<syst
</sys
```



Request Details

Session Id:	ozgnngnnosbm2iha4afaow5u	**Request Type:**	GET
Time of Request:	15/9/2015 16:15:59	**Status Code:**	200
Request Encoding:	Unicode (UTF-8)	**Response Encoding:**	Unicode (UTF-8)

Trace Information

Category	Message	From First(s)	From Last(s)
aspx.page	Begin PreInit		
aspx.page	End PreInit	0.010655	0.010655
aspx.page	Begin Init	0.010709	0.000054
aspx.page	End Init	0.010731	0.000022
aspx.page	Begin InitComplete	0.010747	0.000016
aspx.page	End InitComplete	0.010769	0.000022
aspx.page	Begin PreLoad	0.010786	0.000017
aspx.page	End PreLoad	0.010802	0.000016
aspx.page	Begin Load	0.010818	0.000016

Control Tree

Control UniqueID	Type	Render Size Bytes (including children)	ViewState Size Bytes (excluding children)	ControlState Size Bytes (excluding children)

Session State

Session Key	Type	Value

Application State

Application Key	Type	Value

Request Cookies Collection

Name	Value	Size

Response Cookies Collection

Name	Value	Size

Headers Collection

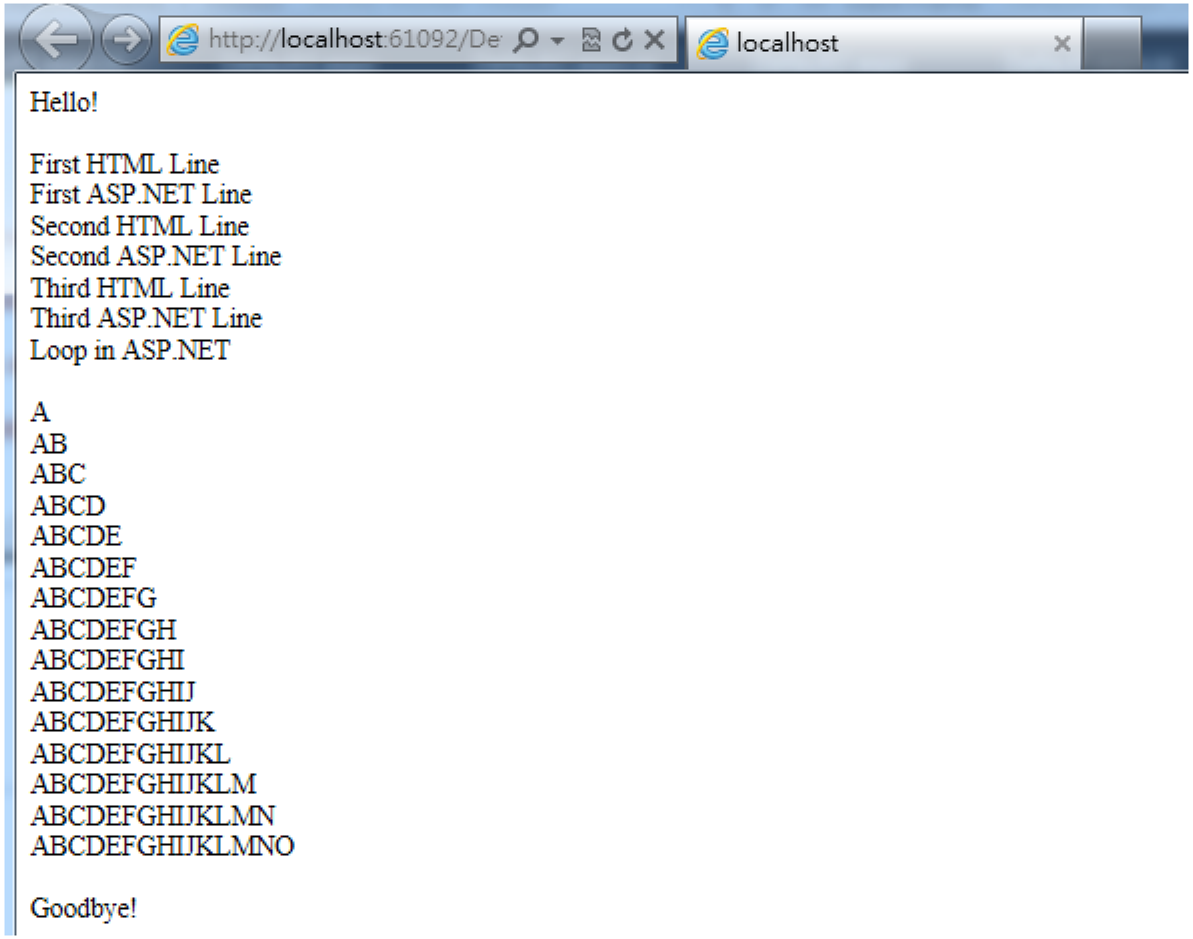
Name	Value
Connection	Keep-Alive
Content-Length	0
Accept	text/html, application/xhtml+xml, */*

Master and Content Pages

- When the pages in a site should share a common theme
  - Master pages contain the reusable theme
  - The file is typically named site.master
- Content pages appear in the master page

Exercises

1. Using Response.Write(…) to display the following page:



```
Hello!

First HTML Line
First ASP.NET Line
Second HTML Line
Second ASP.NET Line
Third HTML Line
Third ASP.NET Line
Loop in ASP.NET

A
AB
ABC
ABCD
ABCDE
ABCDEF
ABCDEFG
ABCDEFGH
ABCDEFGHI
ABCDEFGHIJ
ABCDEFGHIJK
ABCDEFGHIJKL
ABCDEFGHIJKLM
ABCDEFGHIJKLMN
ABCDEFGHIJKLMNO

Goodbye!
```