

# Vad är ASP.NET Web Forms?

# Upphovsrätt för detta verk

Detta verk är framtaget i anslutning till kursen ASP.NET Web Forms vid Linnéuniversitetet.

## Du får använda detta verk så här:

Allt innehåll i detta verk av Mats Looch, förutom Linnéuniversitetets logotyp och symbol samt ikoner, bilder och fotografier, är licensierad under:



Creative Commons Erkännande-IckeKommersiell-DelaLika 2.5 Sverige licens.

<http://creativecommons.org/licenses/by-nc-sa/2.5/se/>

## Det betyder att du i icke-kommersiella syften får:

- kopiera hela eller delar av innehållet
- sprida hela eller delar av innehållet
- visa hela eller delar av innehållet offentligt och digitalt
- konvertera innehållet till annat format
- du får även göra om innehållet

Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp och symbol samt ikoner och fotografier i din nya version!

Vid all användning måste du ange källan: "Linnéuniversitetet – ASP.NET Web Forms" och en länk till <https://coursepress.lnu.se/kurs/aspnet-web-forms> och till Creative Common-licensen här ovan.

# ASP.NET – en del av .NET Framework 4.5

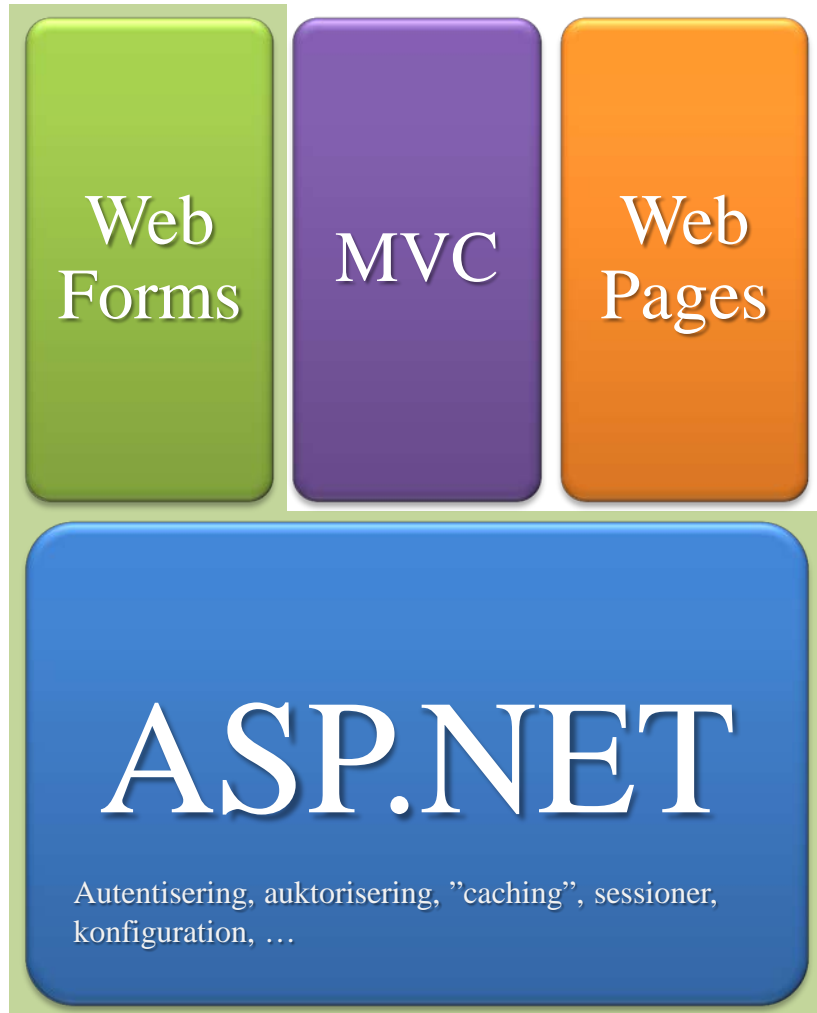
**Microsoft .NET Universe 2013**

- Services**
  - ASP.NET Web API
  - ASP.NET SignalR
  - WCF
  - WCF Data Services
  - LightSwitch OData Services
  - ODATA Lib
  - Service Bus for Windows Server Lib
  - Service Bus for Windows Azure Lib
- Cloud Apps**
  - Windows Azure .NET SDK
  - Windows Azure Storage Lib
  - Windows Azure Configuration Manager Lib
  - Windows Azure Media Services
  - Windows Azure Mobile Services
  - p/p Autoscaling App Block
  - p/p Transparent Fault Handling App Block
- Web Apps** (circled in red)
  - ASP.NET MVC
  - ASP.NET Web Forms
  - ASP.NET Web Pages
  - LightSwitch HTML5 Client
  - ASP.NET Razor
  - ASP.NET Web Optimization
- Core**
  - .NET Runtime
  - Base Class Library
  - Languages (C#, VB, F#)
  - Linq
  - XAML
- Desktop Apps**
  - Windows Presentation Foundation
  - Windows Forms
  - LightSwitch Desktop Apps
  - p/p Prism for WPF
- Windows Store Apps**
  - WinRT XAML/.NET/Windows Store SDK
  - Notification Extensions WinRT
  - p/p Prism for Windows Runtime
  - WinRecipe
  - p/p Unity for Windows Store Apps
- Windows Phone Apps**
  - .NET SDK for Windows Phone
  - Windows Phone Toolkit
  - p/p Prism for Windows Phone
- Partners Cross Platform Apps**
  - Xamarin
  - ITR-Mobility iFacer
  - Citrix Mobile SDK for Windows Apps
- Emerging Application Patterns**
- Established Application Patterns**
- Cross-Cutting Patterns**
- NuGet**
- Open Source**
- Support (MS Official)**
- Visual Studio**

**What's new in .NET Framework 4.5?**

- Windows Presentation Foundation**
  - Built-in Ribbon controls
  - Outlining improvements
  - Ability to add breakpoints to databindings
  - Data source change aware views (Live Shaping)
  - Validation improvements
  - Improved legacy UI integration
  - Dispatcher improvements
  - Speed-up of large data sets
- Windows 8 support**
  - Support for Windows Runtime (WinRT)
  - NET Profile for Metro-style apps
  - Improved support for sharing DLLs between .NET profiles
- ASP.NET**
  - ASP.NET Web API - a new framework for REST endpoints
  - Support for implementing Websocket endpoints
  - Built-in JavaScript + CSS combining and minification
  - Support for Request intelligent cookies
  - Asynchronous pipeline support (Response, Request, Input/Output)
  - Performance improvements: Multicore JIT, 35% faster startup, memory optimizations, assembly sharing between sites, pre-fetch support
- ASP.NET MVC 4**
  - Async controllers
  - Build-in mobile templates + jQuery Mobile support
  - Alternate views (e.g. print version, mobile site)
  - Support for Razor templating engine
- ASP.NET Web Pages 2**
  - New site templates
  - Visualizable validation
  - Support for OData, Entity Framework, and others
  - Build-in web server (IIS Express) and others
  - Did you know? ASP.NET Web Pages 2 can now be used to build a full-featured web application, not just a static website
- Web Forms** (circled in red)
  - Strongly typed data binding
  - MVC-like support for Models
  - HTML5 styled landing expressions
  - Control support for new semantic elements
  - Validator and Localized error messages
  - Markup errors improved (as HTML5 Express used by default)
  - Page Inspector
  - New browser
- Request validation improvements**
  - AntiXSS built-in, validation loadable per field
- Managed Extensibility Framework 2.0**
  - Support for generic parts
  - Outlining improvements
  - Support for explicit and convention-based bindings between objects
  - Support for binding POCO's: no more attribute requirements
- Windows Workflow Foundation**
  - Code-first activity design
  - Faster execution
  - Designer usability improvements
  - Workflow versioning
- Windows Communication Foundation**
  - New channels (UDP multicast, WebSockets, ...)
  - Simplified configuration, X509 cert validation
  - Multiple auth modes for HTTP endpoints
  - New, simple HttpClient class
- ADO.NET**
  - SQL Server 2012 ("Denali") Support
  - High Availability support on connection string level
  - Fast failover across multiple subnets
  - Support for new spatial data types (geography, geometry etc.)
- Entity Framework 4.5**
  - Enumeration support
  - Migrations for schema changes
  - Designer improvements
- Visual C++**
  - Multiple diagrams per model
  - Code-first support
  - Auto-compiled LINQ queries

# ASP.NET och dess programmeringsmodeller



## ✓ Web Forms

- Kontroll- och händelsestyrd programmeringsmodell
- Kontroller kapslar in HTML, JavaScript och CSS.

## ✓ MVC

- För traditionella webbutvecklare.
- Fullständig kontroll över HTML.
- Stöd för enhetstestning.
- Flexibelt.

## ✓ Web Pages

- Påminner om programmeringsmodellen PHP använder.

# Några av typerna i ASP.NET

## ASP.NET 3.5

### System.Web

*HttpApplication*  
*HttpCookie*  
*HttpRequest*  
*HttpResponse*  
*HttpRuntime*  
*HttpServerUtility*

### System.Web.ApplicationServices **NEW**

*AuthenticationService* **NEW**  
*ProfileService* **NEW**  
*RoleService* **NEW**

### System.Web.Caching

*Cache*

### System.Web.ClientServices **NEW**

*ClientFormsIdentity* **NEW**  
*ClientRolePrincipal* **NEW**  
*ConnectivityStatus* **NEW**

### System.Web.ClientServices.Providers **NEW**

*ClientFormsAuthenticationMembershipProvider* **NEW**  
*ClientRoleProvider* **NEW**

### System.Web.Compilation

*BuildProvider*

### System.Web.Configuration

*WebConfigurationManager* **NEW**

### System.Web.Hosting

*ApplicationManager*

### System.Web.Management

*WebBasedEvent*

### System.Web.Security

*FormsAuthentication*  
*FormsIdentity*  
*Membership*  
*Roles*

### System.Web.SessionState

*HttpSessionState*

### System.Web.UI

*Control*  
*MasterPage*  
*Page*  
*ScriptManager* **NEW**  
*System.Web.UI*  
*UpdatePanel* **NEW**  
*UpdateProgress* **NEW**  
*UserControl*

### System.Web.UI.HtmlControls

*HtmlButton*  
*HtmlControl*  
*HtmlForm*  
*HtmlInputControl*

### System.Web.UI.WebControls

*Content*  
*DetailsView*  
*FormView*  
*GridView*  
*LinqDataSource*  
*ListView*  
*LogIn*  
*Menu*  
*ObjectDataSources*  
*TreeView*  
*Wizard*

### System.Web.UI.WebControls.WebParts

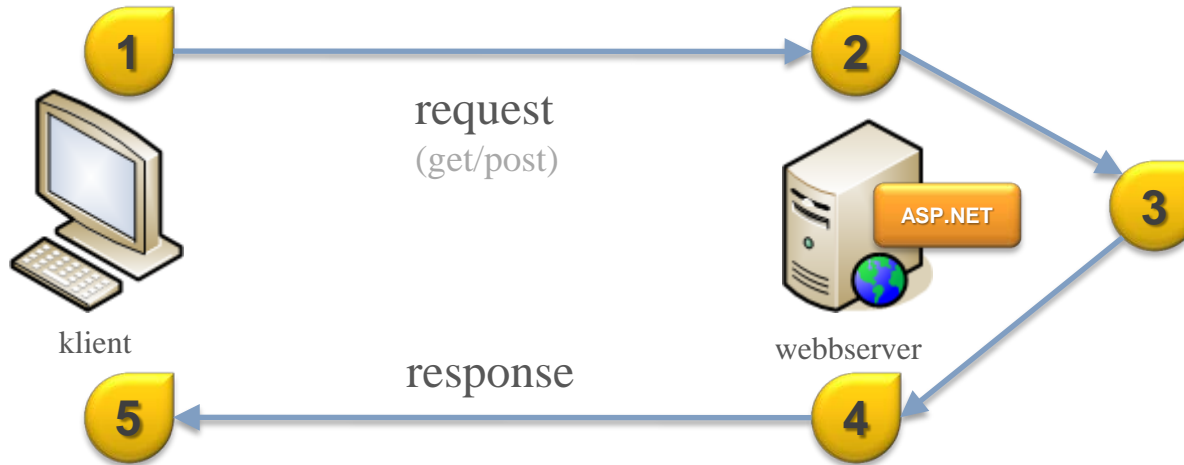
*WebPart*

Oh, vad många typer det är.  
Tror du att vi ska använda  
alla?

Som tur är ska vi inte det. Men  
alla typer vi kommer att använda  
finns å andra sidan inte med...



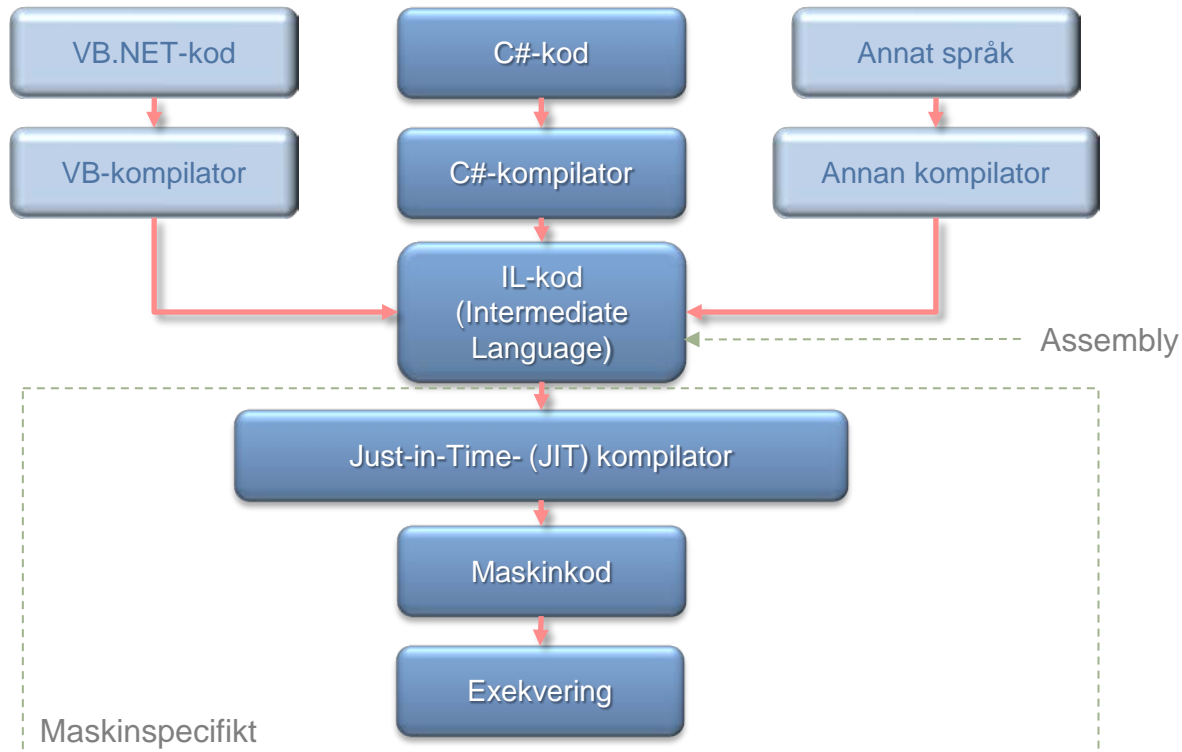
# ASP.NET körs på webbservern



1. Klienten efterfrågar en URL (en vanlig "HTTP request").
2. På webbservern skapar ASP.NET objekt som representerar sidan och kontrollerna sidan innehåller.
3. Kod körs på servern och sidan med kontroller renderas till HTML, som...
4. ...skickas till klienten (en vanlig "HTTP response").
5. Objekten som representerar sidan och kontroller "förstörs" och återanvänds inte.

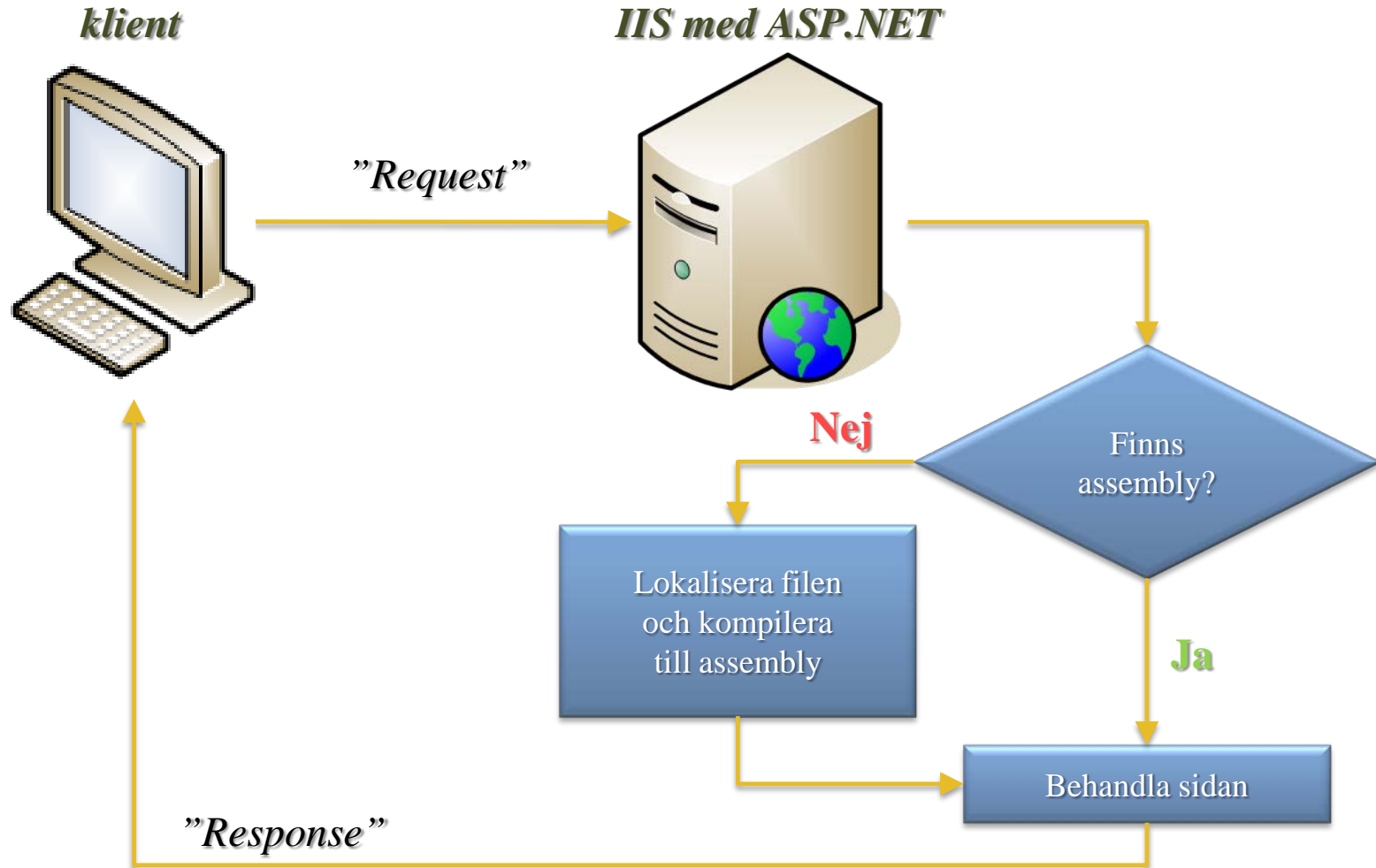


# ASP.NET-sidor kompileras



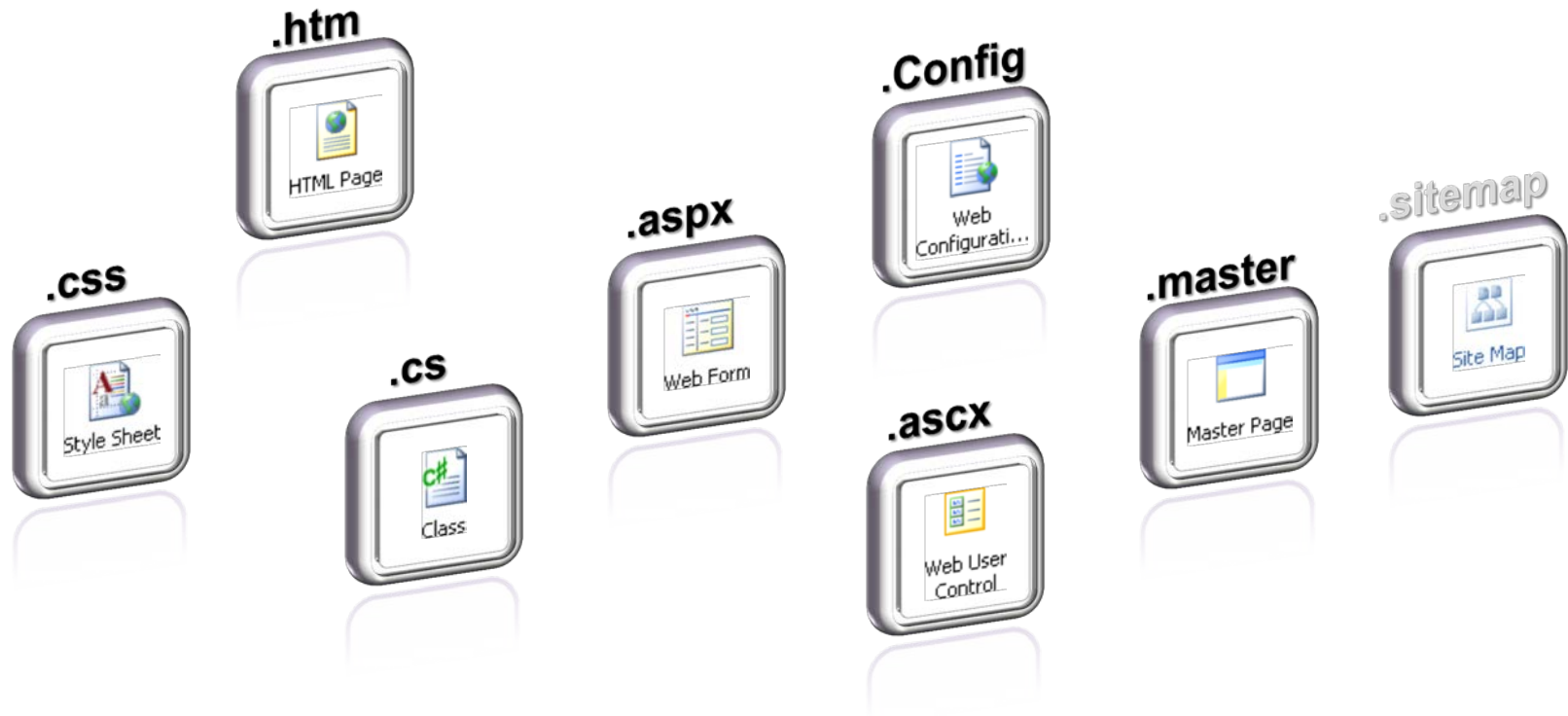
- ✓ ASP.NET-applikationer måste kompileras. Källkod kompileras till IL-kod som i sin tur kompileras till maskinkod.
- ✓ En kompilering görs bara då källkoden är ny eller har ändras.

# Så hämtas en sida



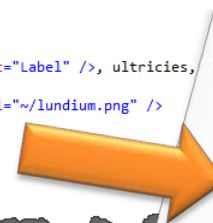
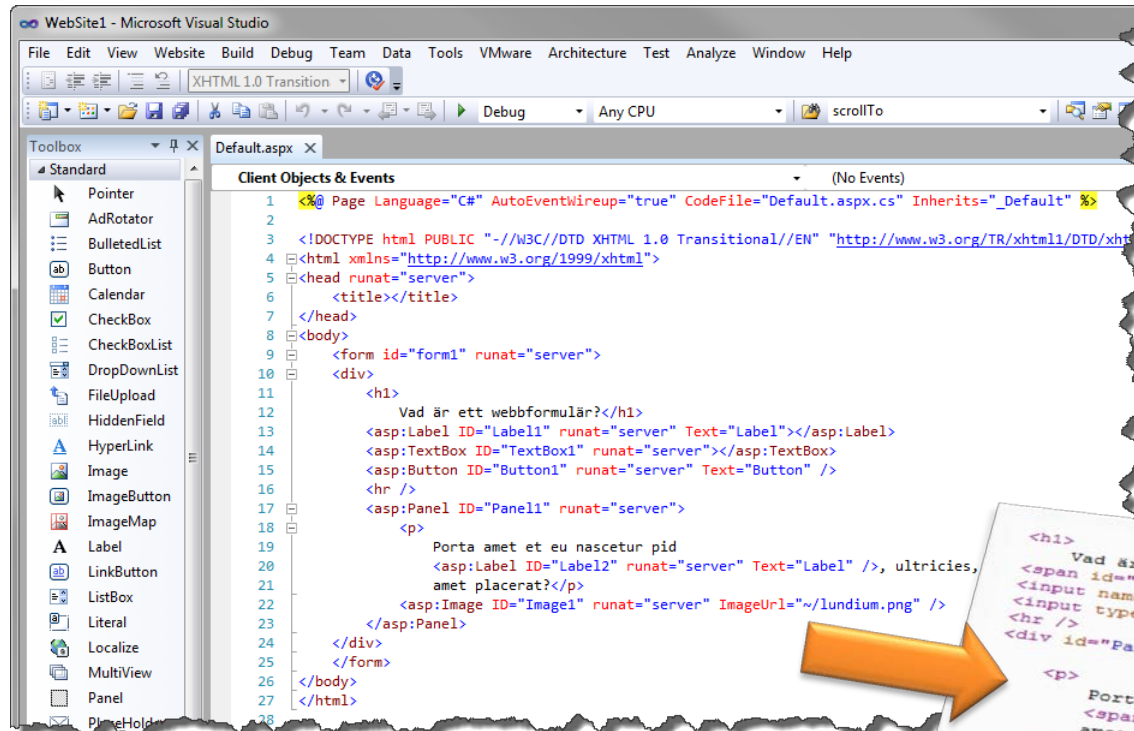


# Vanliga filtyper i en ASP.NET-applikation



- ✓ Webbformulär ("web forms") finns i filer med filändelsen .aspx.
- ✓ Andra filtyper du också kommer i kontakt med under denna kurs är .htm, .css, .cs och .Config.

# Vad är ett webbformulär?



- ✓ Serverkontroller placeras i webbformuläret, renderas och "vanlig" (X)HTML skickas till klienten.

# "Kodmodeller" i ASP.NET

## ✓ "Code-behind model"



```

1 <%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head runat="server">
7 <title></title>
8 </head>
9 <body>
10 <form id="form1" runat="server">
11 <div>
12 <h1>
13 "Inline code model"</h1>
14 <p>
15 Aktuell datum och tid:
16 <asp:Literal ID="DateLiteral" runat="server" /></p>
17 </div>
18 </form>

```

```

1 using System;
2
3 public partial class _Default : System.Web.UI.Page
4 {
5     protected void Page_Load(object sender, EventArgs e)
6     {
7         DateLiteral.Text = DateTime.Now.ToString();
8     }
9 }

```

Den här modellen ska du använda.

- + C#-koden är fullständigt separerad från XHTML-koden.
- + Flexibel och lättöverskådligt.

## ✓ "Inline code model"



```

1 <%@ Page Language="C#" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head runat="server">
7 <script runat="server">
8     protected void Page_Load(object sender, EventArgs e)
9     {
10         DateLiteral.Text = DateTime.Now.ToString();
11     }
12 </script>
13 </head>
14 <body>
15 <form id="form1" runat="server">
16 <div>
17 <h1>
18 "Inline code model"</h1>
19 <p>
20 Aktuell datum och tid:
21 <asp:Literal ID="DateLiteral" runat="server" /></p>
22 </div>
23 </form>
24 </body>
25 </html>

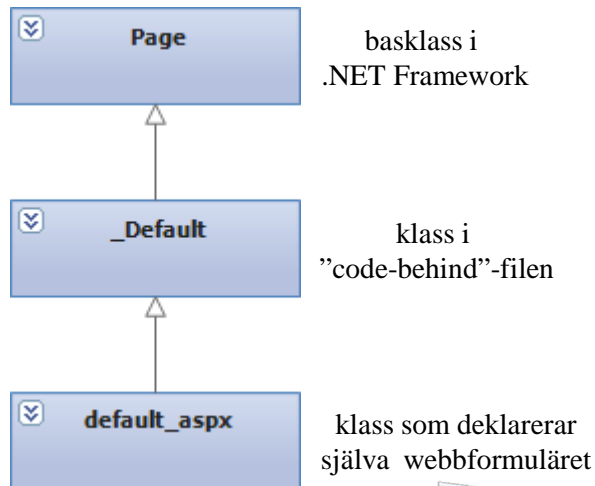
```

Du ska undvika "inline code"-modell under kursen.

- + Allt på samma ställe.
- Blir lätt rörigt.

# Ett webbformulär ärver från...

- ✓ ...en klass som deklarerats i "code-behind"-filen som i sin tur ärver från klassen Page.



# Page-direktivet

- ✓ Längst upp i varje .aspx-fil finns ett "Page"-direktiv, som definierar ett antal saker.

```

1  <%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
2
3  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head runat="server">

```

Vilket programspråk  
sidan använder.

Om händelser ska knytas  
automatiskt till metoder.

Om och i så fall vilken  
"code behind"-fil som används.

Vilken klass  
aspx-sidan ärver från.