

ASP.NET with vb.net 2003

Af Morten Rasmussen, maj 2004

ASP.NET with vb.net 2003.....	1
1. Start med ASP.NET (s. 9).....	2
Installation (s. 23)	2
Editor: Web Matrix	2
Forskel til almindelig ASP (s. 19).....	2
VB.NET	2
2. Opbygning af ASP.NET (s. 45)	3
Hvad er .NET (og hvad gør det)?.....	3
.NET basisklasser (s. 50).....	3
3 måder og placers ASP.NET kode i HTML	3
3. Server Controls and Variables (s. 69)	3
Web sider, HTML forms og Web Forms.....	3
Page_load()	4
Server Controls (web controls)	4
<asp:komponenter> label, dropdownlist, listbox, textbox, radiobutton, checkboxlist,	4
Variables s. 92.....	4
Simple data type s. 95	4
Numerisk.....	4
Text	4
Andre.....	5
Scope.....	5
Navngivning og konstanter	5
Typekonverteringsfunktioner:.....	5
Arrays s. 107	5
Collections	5
4 Control Structures and procedural programmering	5
Operators (assign, arithmetic, numeric, logical) (s. 122).....	5
Kontrol strukturer (branching, looping, jumping) (s. 132)	5
Branching.....	5
Looping and repeat.....	6
Jumping.....	6
Random rnd s. 140-141	6
5 Jumping Structures – Subrutines and Functions (s. 161).....	6
Sub rutine (s. 164).....	6
Functions (s. 174).....	6
Variabel overførsel 'ByVal' eller 'ByRef' (s. 182)	6
God dokumentation (s. 188)	6
6 Event-Driven programming and Postback.....	7
Der er 3 grupper af events.....	7
Hvad er en event?.....	7
- En hendelse (klik på knap).....	7
- ASP.NET registrere at en event er trigget.....	7
- Systemet finder eventen og udfører noget kode.....	7
- Systemet vender tilbage til den oprindelige state og venter på næste event	7
Event programmering	7
- Normalt (gamle dage) talte man om sekventiel programmering	7
- Dette ændre event programmering, sekventiel er ikke længere passende	7
- To forskelle mellem en event procedure og en almindelig procedure	7
o Anden navne syntaks (event procedure bruger *_ , ex. _Click)	7
o Hvordan den eksekveres (når man kalder proceduren eller trigger en event)	7
HTML events	7
- Køres af browseren (ikke part af ASP.NET) (s. 195)	7
Debugging	7

- <%@ Page Language="VB" Debug="true" trace="true" %> (s. 195-197).....	7
ASP.NET page-level events (s.197)	7
- Denne gruppe udføres automatisk når en side loades	7
- Page_Load() Sker når hele siden er synlig.....	7
ASP.NET Web controls events (s. 199).....	7
Forskeld på HTML og ASP.NET Web controls	7
Postback (s. 201)	7
8 Reading From Databases (s. 249)	8
SQL og Stored Procedures	8
Web Matrix Data Explorer (s. 254)	8
Data sider med Web Matrix	8
Data sider vha. Data Explorer (s. 256).....	8
Data sider vha. Web Matrix Template Pages (s. 259)	8
Data sider vha. Code Wizards (s. 265).....	9
ADO.NET (s. 274).....	9
En stor SQL kontra 'filtering queries' (282)	9
9 Advanced Data Handling	10

1. Start med ASP.NET (s. 9)

- Bruges til at lave **dynamiske hjemmesider** ('samling' af MS's ASP og .NET Freamwork)
- Gemmer informationer til databaser og XML **hurtigere** og **nemmere**
- Hvad man kan lave afgrænses kun af fantasien
- **Værdifuld fleksibel 'server-side' teknologi til at lave dynamiske websider**
- Er en teknokogi ikke et sprog
- Filerne hedder .aspx (ved view source ses kun html + <input type .. __VIEWSTATE .. xXxX (VIEWSTATE er en 'encoded' repræsentation af formens tilstand som ASP.NET bruger til at holde styr på alle kontrollers tilstand! Hvis ikke blev en form nulstillet hver gang den blev genereret af serveren (s. 82))
- 3 vigtige pointer vedr. __VIEWSTATE
 - o Det er kritisk for at kunne huske kontrollers tilstand
 - o Man skal ikke programmere noget (integreret/automatisk del af ASP.NET)
 - o Bliver givet til browseren som standard HTML

Installation (s. 23)

- Kræver MDAC (Microsoft Data Access Components)
- Kræver også .NET freamwork

Editor: Web Matrix

- Integreret editor og webserver (IIS, Windows 2003 server og andre)
- Gratis light udgave af *Visual Studio* (ældre: Visual Interdev)

Forskel til almindelig ASP (s. 19)

ASP.NET	ASP
Stor kasse af ' fulde ' programmeringssprog	Kun <i>scripting language</i> , JavaScript, VBScript (light udgaver af alm. Sprog)
	Kode sammenblanding og bbegrænset funktionalitet
Indeholder en stor mængde kontroller	Indeholder 6: ex. request og response
Adskil html og asp.net kode	-
Implementation af en robust event model	

VB.NET

- Man kan lave .NET med mange sprog men VB.NET følger gratis med ASP.NET

2. Opbygning af ASP.NET (s. 45)

Hvad er .NET (og hvad gør det)?

- ASP.NET er kun en ud af mange komponenter i freameworket
- Ordner det hårde arbejde ved at skabe store pålidelige applikationer
- Tilbyder to slags programmer: lokale og kører over nettet
- Nemt at få imponerende resultat
- Kan få forskellige programmeringssprog til at arbejde sammen
- .Net er et objekt orienteret udviklingsmiljø (s. 48)

Sammensat af: MSIL + CLR + Class Libraries + .NET Language + ASP.NET + Web Services

Fra kode til maskinkode, i mange år har der været to metoder:

- Pre-kompileret: Hurtig 'execution' (svært med rettelser)
- Interpreted code: Kompileres ved første udførsel ('rettelser hurtig ude' men langsom)

.NET løser dette problem med to trins kompilering

- .NET kompilere til et specielt format MSIL (ingen specifik maskin optimering, sker når en bruger spørger efter en ASP.NET side), derefer (MSIL sprog er ens for alle 'sprog!')
- Sørger CLR for en JIT (Just In Time) kompilering til optimeret maskinkode

.NET basisklasser (s. 50)

- Ex. System.Drawing (*namespace : gruppe af klasser med en bestemt funktionalitet*)
- Et klassebibliotek kan findes på nettet (Freamwork SDK)
- For at bruge basisklasserne skal de importeres, ex.
 - o `<%@ Namespace=System.Drawing %>`

.NET Freamwork: MSIL + CLR + .NET Language Base classes (s. 52)

- Når freameworket er installers kan bl.a. .aspx filer læses

3 måder og placers ASP.NET kode i HTML

- *Inline*: `<% %>` ... classic ASP ... ikke en foretrukket teknik I ASP.NET
Også kendt som 'render code block' – samme sammenblanding som alm. ASP!
- `<script>`: `<%@ Page language="VB">` og `<script>`
Web server scanner filen for '`<script runat="server">`' og udføre først Page_Load()
- *Server controls*: `<asp:label ... fromfor response.write ...` opdeling af og renere kode

Binding til database eks s. 61

Binding til xml fil eks s. 63

3. Server Controls and Variables (s. 69)

- VB.NET er et 'strongly typed' sprog – hver variable har en type tilknyttet

Web sider, HTML forms og Web Forms

- Web form er et nyt koncept, meget nemmere og hurtigere at programmere end HTML forms
- HTML forms kan kun bruge standard HTML
- ASP.NET har sine egne Web Form komponenter (kører på serveren)
- Web form er sider der kominere ASP.NET kode og HTML
- ASP.NET form er sider der indeholder ASP.NET server controls inside a web form

HTTP er en stateless protokol som er med til at gøre internettet hurtigere men man kan ikke skelne mellem brugere

Med ASP.NET oprettes HTML dynamisk

ASP.NET form komponenter er en udvidelse af HTML form komponenter, forskellen er at ASP.NET komponenter dannes dynamisk når de skal bruges
(andre komponenter er kalender og grid)

Alle ASP.NET forms bruger POST metoden - sender information som del af request (mod GET metoden der tilføjer information til URL'en)

Page_load()

- bliver udført hverenten htmsiden vises eller opdateres (F5)
- 'if Page.IsPostBack then' ... siden vises for anden gang eller mere

Server Controls (web controls)

- En web control har to nødvendige attributer: runat="server" og id="xx"
- <asp: indikere at der er en indbygget ASP.NET kontrol
- runat="server" – fortæller serveren at den skal køre kontrollen og returnere html kode til klienten men andre ord at formen eller komponenten bliver udført på serveren

<asp:komponenter> label, dropdownlist, listbox, textbox, radiobutton, checkboxlist, ...

<asp:label> s. 75
<asp:label id="labNavn" runat="server" text="hej med dig" forecolor="red">
<asp:dropdownlist> s. 79
msg = msg + List1.SelectedItem.Value
<asp:listbox> s. 82
if listbox1.Items(0).Selected then msg = msg & listbox1.Items(0).Text
Multibel selection: <asp:listbox id="list1" runat="server" selectionmode="multiple"
AutoPostBack="true" ... husker valg ved 'PostBack'
<asp:textbox> s. 84
<asp:textbox textmode="multiple" rows="3" columns="50">
textmode="password
<asp:radiobutton> s. 87
<asp:radiobuttonlist> s. 87
radio1.SelectedItem.Value
<asp:checkbox> s. 89
<asp:checkboxlist> s. 89
<asp:checkbox id="chkXxx" runat="server" />
chkXxx.Items(0).Selected
<asp:button>

Variables s. 92

- Plads I memory gemp med et navn og en given datatype
- VB.NET er *strongly typed* language (hver variabel har en datatype)
- Man kan tjekke om en variabel er tom vha. **IsNothing(myVar)**

Simple data type s. 95

- Der er 12 indbyggede typer i VB.NET

Numerisk

Integer (-/+ 2.147.483.648)	intX	s. 99
byte (0..255)	bytX	
short (-32.768 - +32.767)	shoX	
long (-/+ 9.223.372.036.854.808)	lngX	
single (decimaltal med med 28 'digits' og derover)	sngX	
double (.. endnu mere)	dblX	
decimal (decimaltal med færre end 28 'digits')	decX	

Text

String	"Xxx"	strX	s. 97
--------	-------	------	-------

Brug '&' (ampersand operator) (undgå +!!)

Concatination kan være langsom brug class `StringBuilder` hvis der er mange

Char		chrX
Andre		
Date	#1/1/2004#	datX
Boolean		blnX

Scope

- Mindste scope er en blok (asgrænses af: end If, Else, Loop eller Next)
- Fordel ved lille scope: Spare resourcer
- Næste scope er '*procedure-level*'
- '*global variables*' (står over `Sub Page_Load()`)

Navngivning og konstanter

- Dim xx As double
- Const XXX As integer = 1975 s. 106

Typekonverteringsfunktioner:

- CInt(), CDate() og CStr() s. 107

Arrays s. 107

- Dim Xxx(8) As String
- Dim Xxx(8,8) As String s. 110
- Fordele: Nemme, hurtig at tilgå, datatypen er given
- Ulemper: Fast størrelse, svært at 'indsætte' elementer (collections løser dette)

Collections

- Ikke fast antal, behøver ikke at være samme datatype
- ArrayList (men ... langsommere og bruger mere memory) s. 112
`MyArrayList.Add("Xxx") myArrayList.Insert(2, "Xxx")`
- HashTable (brug key – nr, tekst el. Dato - fremfor indeks) , meget hurtig s. 114
Langsommere at tilføje og fjerne elementer, key skal være unik, kan ikke sorteres
- SortedList
(indeholder alle fordele fra ArrayList og HashTable, sotes ud fra key s. 118)

4 Control Structures and procedural programming

- *branching and looping*

Operators (assign, arithmetic, numeric, logical) (s. 122)

- assign: =, += s. 122
- Arithmetic: +, -, *, /, ^, MOD s. 122
- numeric: =, <, <=, <>, >, >= s. 127
- logical: NOT, AND, OR s. 128 (brug paranteser!)

Kontrol strukturer (branching, looping, jumping) (s. 132)

- Branching
- Looping and repeat
- Jumping

Branching

- If ... then ... else (s. 136)
- Select case (s. 144) **ToUpper()**
Select Case Xxx
Case "yes", "YES", "y", "Y"
X

Case else
End Select

Looping and repeat

- For ... next Kender antallet af gange (bruger variabel) s. 148
 - Do while ... Loop Kender ikke antallet af gentagelser (test) s. 152
 - Do ... Loop Until --/-- (mindst en gang) s. 156
 - For Each ... Next Har en samling men ved ikke hvor mange, hvert enkelt item (array eller collection) s. 157
- Ikke egentlig et loop men muliggøre at gøre flere ting på samme objekt
With ... End With s. 157

Jumping

- Subroutines (eller procedures)
- Functions

Random rnd s. 140-141

- RandomNumber = int((1 + upperbound – lowerbound) * rnd) + lowerbound

5 Jumping Structures – Subroutines and Functions (s. 161)

- For at modularisere koder
- Fordele: nemmere at skrive, læse, overskue, teste og man spare kode

Sub rutine (s. 164)

- Eksempel: Page_Load() (dog en speciel da den 'dannes' af VB.NET)
- Sub MySubroutine() (procedure)
 .. code ..
End Sub
- Det er ikke muligt med sub rutiner inde i andre (nastede)
- Kald en sub rutine med: **Call** MySubroutine() (call ikke nødvendig men viser at ...)
- Parametre medtages: Sub MySubroutine(MyParameter as DataType) – Call MyS(“xx”)
- Hvis et subrutinekald skal deles over flere linier brug _ (s. 168)
- **Web Controls as Parameters:** Syb MySub(Target as textbox) s. 171

Functions (s. 174)

- Kan returnere værdi til den kaldende kode
- Function MyFunction() As Integer
 .. code ..
 Return MyResult (eller MyFunction = MyResult)
End Function
- God modulopbygning kræver god dokumentation

Variabel overførsel 'ByVal' eller 'ByRef' (s. 182)

- Default overføres variable *ByVal* i funktioner og subrutiner
- *ByVal*: Man arbejder med en kopi af variabelen
- *MyRef*: Man arbejder med den oprindelige variabel
- Hvis variabelen fylder meget i memory sparer *ByRef* ressourcer
- (andre måder: code-behind ??? og objekter)

God dokumentation (s. 188)

- Formål, oprettelses dato, versionsnr, forfatter og variabel beskrivelse

6 Event-Driven programming and Postback

Der er 3 grupper af events

- HTML events (exe i browseren)
- ASP.NET page-level events (bl.a. køre kode når siden loades, ex. *postback*), alle sker inden brugeren ser siden
- ASP.NET server control events, den største gruppe trigges af brugeren

Hvad er en event?

- En hændelse (klik på knap)
- ASP.NET registrere at en event er trigget
- Systemet finder eventen og udfører noget kode
- Systemet vender tilbage til den oprindelige state og venter på næste event

Event programmering

- Normalt (gamle dage) talte man om sekventiel programmering
- Dette ændre event programmering, sekventiel er ikke længere passende
- To forskelle mellem en **event procedure** og en **almindelig procedure**
 - o Anden navne syntaks (event procedure bruger `_*`, ex. `_Click`)
 - o Hvordan den eksekveres (når man kalder proceduren eller trigger en event)

HTML events

- Køres af browseren (ikke part af ASP.NET) (s. 195)

Debugging

- `<%@ Page Language="VB" Debug="true" trace="true" %>` (s. 195-197)
- `Trace.write()`

ASP.NET page-level events (s.197)

- Denne gruppe udføres automatisk når en side loades
- Hver gang en side kaldes, sker initialisering, processing og disposing af information
- **Page_Init()** Sker når sider er blevet initialiseret
- **Page_Render()** Sker for at supportere avancerede emner
- **Page_Load()** Sker når hele siden er synlig
- **Page_Unload()** Sker når siden forsvinder fra IIS memory (ikke når siden forlades)

ASP.NET Web controls events (s. 199)

- Sker ved komponenter som `<asp:textbox>` og `<asp:button>`
- 3 grunde til at disse er mere brugbare end HTML events
 - o Man kan eksekvert kode i event procedure på serveren
 - o Man skal ikke stole på klientens browser
 - o Koden kan skrives i ethvert .NET sprog
- 2 ting for at oprette en event
 - o Tilføj en attribut til komponenten og sæt den lig med en event procedure
`<asp:button id="bXxx" onclick="Button1ClickEventHandler"`
 - o Opret denne event procedure
Sub **Button1ClickEventHandler(Sender As Object, E As EventArgs)**
 - Sender er en reference til det object der triggede eventen
 - E er en *event class* til information om tilstand af eventen

Forskeld på HTML og ASP.NET Web controls

- Måden eventsne udføres på
- **Nogle** (ex. *MouseOver* håndtere browseren da serveren ikke kan)
- Andre sendes vha. *Postback message* til serveren der håndtere eventen

Postback (s. 201)

- Postback er den proces browseren bruger til at sende information tilbage til serveren

- Sker kun hvis **runat="server"**
- Kun ASP.NET Web controls (almindelig ASP havde dette svagt implementeret)
- `__VIEWSTATE` benyttes (se øverst)
- Man kan teste på dette vha. **If Page.IsPostBack then**
- Lommeregner eksempel s. 205

8 Reading From Databases (s. 249)

- Til dette bruges **ADO.NET** som er en data tilgangs teknologi og en del af .NET framework
- Objekt ADO.NET kan skabe forbindelse til databaser
- Database, tabeller, normalisering

SQL og Stored Procedures

- *Queries* er den måde vi kommunikerer med databaser (udtræk eller manipulation)
- Det sker vha. *SQL* eller *stored procedures*
- Stored procedures er mere end SQL!: En samling af SQL statements der udføres vha. Kald af et navn (sammenlign med procedure)
- Hvorfor brug stored procedures frem for sql!?
 - o Sikkerhed, beskyttet for visse muligheder for hacking
 - o Hastighed, SP bliver optimeret første gang de kaldes
 - o Man adskiller SQL fra form kode

Web Matrix Data Explorer (s. 254)

- Access eller SQL DB
- Web Matrix kan ikke benytte *stored procedures* med Access!

Data sider med Web Matrix

- Disse kan laves på flere forskellige måder

Data sider vha. Data Explorer (s. 256)

- Benytter en **MxDataGrid** og en **AccessDataSourceControl**
- Har nogle simple attributter (s. 258) bl.a. **AllowSorting**

Data sider vha. Web Matrix Template Pages (s. 259)

- Er en simple måde at få lidt mere funktionalitet i grid'den
- De mulige templates er
 - o *Simple Data Report* simpel uden paging og sorting
 - o *Filtered Data Report* men et filter så man kan udvælge bestemte rækker
 - o *Data Report med paging*
 - o *Data Report med paging og sorting*
 - o *Master-detail grid* med to grids en master tabel og en child
 - o *Editable Grid* med opdateringsmulighed
 - o *Simple Stored Procedure* bruger en grid og en stored procedure
- De bruges med en SQL Server database,- kræver mmodificeringer hvis Access

- Eksempel (s. 260)

```
<% import Namespace="System.Data.OleDb" %>
Dim ConnectionString As String = ...
Dim CommandText As String = "Select * from table"
Dim myConnection As New OleDbConnetion(ConnectionString)
Dim myCommand As New OleDbDataAdapter(CommandText, my Connetion)
Dim ds As New DataSet()
myCommand.Fill(ds)
DataGrid1.DataSource = ds
DataGrid1.DataBind()
```



```
<asp:datagrid id="dgXxx" runat="server" AllowPaging="true" PageSize="6"
    OnPageIndexChanged="procedure"
    AllowSorting="true" OnSortCommand="procedure2">
```

```
Kald med
Sub Page_Load(...)
    If not Page.IsPostBack then
        BindGrid()
    End if
End Sub
```

Data sider vha. Code Wizards (s. 265)

- Tillader at tilføje rutiner
- Brug af Web Matrix 'SELECT Data Code Wizard'

ADO.NET (s. 274)

- Data access mulighederne ovenover benytter alle ADO.NET
- *Connection*: Sti til database
 - o OleDbConnection (s. 276)
 - o .ConnectionString
- *Command*: SQL
 - o Connection property – Hvilken db?
 - o CommandText – SQL eller stored procedure kommando
 - o OleDbCommand (s. 276)
 - o Metoder:
 - ExecuteNonQuery
 - ExecuteReader
 - ExecuteScalar (en enkelt værdi)
 - o Kan indeholde flere parametre (s. 277)
- *DataAdapter*: Håndtere udtræk og opdatering af data
 - o SelectCommand
 - o UpdateCommand
 - o InsertCommans
 - o DeleteCommand
 - o OleDbDataAdapter, udfører ex. automatisk selectCommand når **Fill()** kaldes, de andre kaldes når **Update()** bliver kaldt (s. 283)
- *DataSet*: Gider en beholder (store) til dataene
- *DataReader*: En hurtig mulighed for read-only access til data, - håndtere DB kommunikation (s. 283)
 - o Skal gerne ske så hurtigt som muligt grundet brug af ressourcer
 - o Være connected så lidt tid som muligt, - muliggør hurtigt andre request
 - o Hvis den åbnes manuelt skal den også lukkes manuelt!
- *System.data*: set af objekter bl.a. DataSet
- *System.data.OleDb*: set af objekter til kommunikation med OLEDB databaser
- *System.data.SqlClient*: set af objekter til kommunikation med SQL Server (så der er bl.a. to forskellige Connections, muligår optimering til given db)

En stor SQL kontra 'filtering queries' (282)

- Normalt "Select * from tabel WHERE (id = " & myId & ")"
- Fylder mindre kode
- Dette er en dårlig måde fordi det muliggør hacking (SQL Injection Attacks)
- Hvis man i stedet bruger *parameters* er dette ikke muligt

9 Advanced Data Handling

Eksterne funktioner

Hour(Now)

Minute(Now)

Second(Now)