



Denne guide er oprindeligt udgivet på Eksperten.dk

Visual Basic for begyndere

I denne artikel vil jeg gennemgå Visual Basic (IKKE .NET / Visual Basic 2005 Express Edition) på et begynder-niveau.

fed kursiv understreget [link til billede](#)

Skrevet den **03. Feb 2009** af **jih** I kategorien **Programmering / Visual Basic** | ★★★★★

Først af alt:

Visual Basic 2005 Express Edition = Visual Basic.NET og!

Eksperten.dk har skiftet måden man skriver artikler på, og jeg har ikke tid til at rette i den. Jeg har fjernet alle tags for at rydde op i den, men den er stadigvæk lidt uoverskuelig - så bær over med mig. :-)

On November 7 2005 Visual Studio 2005 was released, which includes Visual Basic .NET 2005 along with the .NET Framework 2.0. Microsoft also introduced Visual Basic 2005 Express Edition, a cut-down free edition designed to introduce people to the Visual Basic .NET environment

Så hvis du har VB 2005 Express Edition, så vil denne artikel ikke hjælpe dig!

Jeg er ny til artikler, så bær over med mig :)

Giv endelig kommentarer! Modtager med glæde enhver kommentar, er den dårlig eller god. Ros er jo altid dejligt at få, og dårlige kommentarer .. så længe det ikke er sådan nogle som bare kritiserer for at kritisere, men faktisk er nogle der peger ud hvad der er galt med min artikel, er jeg glad for at høre det. Man lærer jo så længe man lever ;)

Edit:

- Jeg er kommet i tanker om, at jeg mere eller mindre kun henviser til koder her. Siden det skulle handle om det generelle af Visual Basic bliver jeg nødt til at lave den lidt om, så jeg tilføjer et lille ekstra afsnit om hvordan man placerer objekter og sådan noget ind øverst. :)

- Forklaringer om hvor koden skal stå er blevet tilføjet

- Svar til antikat's kommentar: Alle de koder jeg har skrevet her, virker fint i min version af Visual Basic..

Jeg bruger Visual Basic 6. Du bruger en version som er bygget på Visual Basic.NET..

Hvis du ellers har de problemer, vil jeg gå ud fra at du har skrevet koden forkert. Prøv at kopiere koden, dobbeltklikke på formen, Fjern alt der står der (hvis der altså kun står, fx, Private Sub Form_Load() og End Sub), og sætte koden ind. Den skulle virke fint. Noget af den kode jeg bruger, kræver nogle objekter. Fx hvis der står "Command1.Caption", så kræver det et objekt der hedder Command1. Som så i min artikel er en command button.

Vi går bare direkte i gang:

Når du skal lave et program, starter du med at åbne Visual Basic, og vælger New Project. Der plejer at poppe en boks frem som spørger hvad det skal være. Der trykker du bare OK (Hvis Standard EXE er valgt), ellers vælger du Standard EXE. Så skulle der komme en "form" frem, som er hvordan programmet kommer til at se ud, når vi starter det. Du kan til enhver tid teste programmet, og se hvordan det virker ved at trykke F5.

Når du dobbeltklikker på denne form, kommer der et andet vindue frem - et vindue hvor du kan skrive koderne. Her skal alle koderne til den bestemte form skrives.

Du kan tilføje objekter til dit program ved at trykke på den menu, som jeg mener er på venstre side som standard. Hvis du dobbelt klikker på en af de ting der er på den toolbar, vil det komme frem midt på formen. Hvis du siden dobbeltklikker på det objekt du lige har placeret, kommer kode-vinduet frem - men denne gang med indgang og udgang til koden for det objekt du klikkede på.

Vi starter med Option Explicit, Dim og MsgBox(). Option Explicit bør altid stå øverst i din kode, siden den tvinger dig til at deklarerer dine variabler før den går i gang med at compile din kode. Så længe du har den stående i toppen af din kode, vil du altid være (næsten) sikker på at du ikke laver stavefejl i dine variabler. Eksempel: kode 1.1

Dim skal skrives før du bruger en variabel. Brug: Dim (variabelnavn) As (Typenavn). Eksempel: kode 1.1
MsgBox() er en funktion, som kan bruges til at få en boks til at poppe frem, med noget tekst som du selv vælger. Brug: MsgBox (besked) [(type af boks) (titel på boks)]. Eksempel: kode 1.1

kode 1.1

```
---  
Option Explicit  
  
Private Sub Form_Load()  
    Dim strHello As String  
    strHello = "Hej!"  
    MsgBox strHello  
End Sub  
---
```

Dette lille program gør det, at når du åbner din form (i dette tilfælde - program), popper der en boks frem, der siger "Hej!".

Første linie sørger for at compileren tjekker alle variabler igennem koden, for at se om de er definerede. Hvis ikke, giver den en fejlbesked ("Variable not defined") og vil ikke køre programmet.

Tredje linie (Private Sub...) og sidste linie (End Sub) er hoved og hale på den kode, som du vil have programmet til at køre igennem, når formen bliver loadet. De to linier kommer automatisk frem ved at dobbeltklikke på formen.

Fjerde linie definerer variabelen strHello som en String, som er en variabel der kan indeholde alle tegn. Femte linie fortæller compileren at du vil have strHello variabelen til at indeholde en værdi der hedder "Hej!" og tildeler strHello den værdi.

Sjette linie er MsgBox() funktionen, som får en boks til at poppe frem med beskeden "Hej!" (siden den er tildelt variabelen strHello, og strHello har værdien "Hej!").

Her er et lille eksempel på Option Explicit:

kode 1.2

```
---  
Private Sub Form_Load()  
    Dim strHejMedDig As String  
    strHeMedDig = "Hej Med Dig!"  
    MsgBox strHejMedDig  
End Sub  
---
```

Dette program giver en blank boks frem, når formen bliver loadet. Hvorfor? Fordi at strHejMedDig har ikke fået tildelt en værdi. Stavefejlen på linie 3 bliver ikke tjekket, pga. at der ikke er nogen Option Explicit.

Hvis vi havde Option Explicit øverst, ville fejlen komme ("Variable not defined"). Så kan man regne ud at hvis man har en lang kode med masser af variabler, ville det være smart med en Option Explicit, så man kan tjekke variablerne, og ikke bare stå og ikke vide hvad fejlen er. Det kan gøres lidt nemmere ved at gøre det sådan at Option Explicit bliver skrevet i toppen af din kode automatisk hver gang du åbner Visual Basic. Du kan gøre dette ved at gå ind i Tools -> Options... og sætte flueben ved "Require Variable

Declaration" og trykke OK. nu kommer det automatisk frem hver gang.

Data typer:

Der er 11 forskellige data typer i Visual Basic 6. Det er Boolean, Byte, Currency, Date, Double, Integer, Long, Object, Single, String, og Variant. De bliver alle brugt til forskellige ting, og du kan få dine programmer til at arbejde langt hurtigere, hvis du bruger dem rigtigt. Jeg vil forklare de mest brugte. Dine programmer vil køre meget hurtigere, hvis du har angivet en data type når du definerer variabler, end hvis du ikke angiver en data type. Hvis du ikke angiver en data type, vil den automatisk blive til Variant, som er den type der fylder mest hukommelse og derfor arbejder langsomt.

Typerne:

Byte:

Byte data typen er en 8-bit variabel, som kan indeholde værdier fra 0 til 255. Dette kan være meget nyttigt til at holde binært data og til at fx sende/modtage byte værdier.

Double:

Double data typen er et 64-bit "floating point" nummer (er ikke ligefrem nogen ørn til dansk, men tror vist det hedder decimaltal) , som man bruger hvis man vil være helt nøjagtig. Den rækker fra -1.79769313486232e308 til -4.94065645841247e-324 for negative værdier og fra 4.94065645841247e-324 til 1.79769313486232e308 for positive værdier.

Integer:

Integer data typen er et 16-bit nummer, som rækker fra -32768 til 32767. Integer betyder "helt tal" på engelsk, og der er derfor ingen kommatal.

Long:

Long data typen er et 32-bit nummer, som rækker fra -2,147,483,648 til 2,147,483,647. Long variabler kan kun indeholde hele tal. De fleste Win32 funktioner bruger Long data typen for at programmet skal køre bedre. Jeg bruger selv Integer for det meste, siden jeg næsten aldrig kommer til at bruge tal der er højere end 32767. Hvis du har bare den mindste mistanke om at tallet kan blive højere end 32767, så brug hellere Long istedet for Integer.

Single:

Single data typen er et 32-bit nummer, som rækker fra -3.402823e38 til -1.401298e-45 for negative værdier og fra 1.401298e-45 til 3.402823e38 for positive værdier. Når du skal bruge kommatal indenfor denne rækkevidde, er det smartest at bruge denne data type.

String:

String data typen har plads til ca. 2 milliarder tegn. Hvert tegn har en værdi, som rækker fra 0 til 255, baseret på ASCII. String data typen bliver brugt når der skal være tekst i en variabel.

Nu vil jeg prøve at give nogle eksempel-koder, som jeg vil forklare lidt om. Men der er lige et par småting der skal laves i din form først:

- Opret en Label, to Command Button og en TextBox, og placér dem hvor som helst på din form.

Dobbeltklik så på formen (ikke på nogen af de ting du lige har placeret, men på baggrunden). Så kommer vi til koden:

kode 1.3:

Option Explicit

Private Sub Form_Load()

```

Label1.Caption = "Jeg kan programmere!"
Command1.Caption = "Klik på mig!"
Command2.Caption = "Skriv og klik!"
Text1.Text = ""
End Sub

```

```

Private Sub Command1_Click()
Label1.Caption = "uuh tryk igen ;)"
End Sub

```

```

Private Sub Command2_Click()
MsgBox Text1.Text, vbOkOnly, "Jeg kan programmere!"
End Sub

```

Denne kode vil lave et program, som har forskellige små funktioner.

Private Sub Form_Load() fortæller compileren at nu skriver vi den kode der skal køre lige så snart vi åbner programmet. I denne, kan du se Label1.Caption, Command1.Caption, Command2.Caption og Text1.Text. Caption kan også gives til fx Form1.Caption, som giver titelen på formen. Her er det stort set det samme, hvor Label1.Caption angiver teksten på Label1, og Command1.Caption angiver teksten på Command1, osv..

Private Sub Command1_Click() fortæller compileren at nu skriver vi den kode der skal køre når vi klikker på Command1. I dette tilfælde skifter det Label1's tekst til "uuh tryk igen ;)".

Private Sub Command2_Click() er det samme som ovennævnte, men nu er det bare Command2 istedet for Command1. Ved klik af denne knap sender vi en dialogboks frem, hvor teksten er det samme som du har skrevet i tekstboksen, Text1.

I den MsgBox vi kaldte denne gang, har nogle flere funktioner end vi har set før. Nu har vi vbOkOnly og "Jeg kan programmere!" med.

MsgBox'ens parametre ser sådan ud:

```

MsgBox(prompt[, buttons] [, title] [, helpfile, context])

```

Alt som står i [] er ikke nødvendigt, men kan skrives. Jeg har personligt aldrig rigtig brugt de to sidste. prompt er hvad der skal stå i den, buttons er hvad for en type det skal være (eksempel længere nede) og title er titelen på boksen.

Alle End Sub's du kan se, er for at slutte Sub'en af. For et eller andet sted skal koden jo stoppe.. ;-)

Det er altid godt at kunne kommentere hvad du laver, så andre (og dig selv) ved hvad det er du har lavet. Det kan godt ske at du sidder og programmerer noget som ser helt ok ud, og meget nemt. Men siden det er en lang og (lidt) indviklet kode, så nogle måneder efter når du ser på programmet igen, vil du ikke ane hvad det var du engang lavede. Du kan lave kommentarer ved at bruge "" tegnet. Eksempel: Vi skriver nu det forrige program om - med kommentarer.

kode 1.3.1

Option Explicit ' Har jeg nogen fejl i mine variabler?

```

Private Sub Form_Load() ' Når programmet åbnes
Label1.Caption = "Jeg kan programmere!" ' Ændr Label1's tekst til "Jeg kan programmere!"
Command1.Caption = "Klik på mig!" ' Ændr Command1's tekst til "Klik på mig!"
Command2.Caption = "Skriv og klik!" ' Ændr Command2's tekst til "Skriv og klik!"
Text1.Text = "" ' Sæt Text1 til at være tom
End Sub

```

```
Private Sub Command1_Click() ' Når Command1-knappen bliver trykket
    Label1.Caption = "uuh tryk igen ;)" ' Ændr Label1's tekst til "uuh tryk igen ;)"
End Sub
```

```
Private Sub Command2_Click() ' Når Command1-knappen bliver trykket
    MsgBox Text1.Text, vbOkOnly, "Jeg kan programmere!" ' Åbn en dialogboks, hvor der står den tekst du
    har skrevet i Text1,
                                     ' med kun en OK knap og titelen "Jeg kan programmere!"
End Sub
```

Der er ingen særlig regel for hvordan du skal sætte dine kommentarer. Du må sætte dem hvor som helst, og hvornår som helst .. så længe det er det sidste der står på linien.

Du kan også sætte flere variabler, strings, tal eller hvad det nu skal være, sammen, vha. ampersand "&".

```
Dim strS As String ' strS - brugt til at indeholde hele teksten
Dim strName As String ' strName - brugt til at indeholde navn
Dim intAge As Age ' intAge - brugt til at indeholde alder
```

```
intAge = 20 ' Sæt alder til 20
strName = John ' Sæt navn til John
strS = "Hej! Mit navn er " & strName & ", og jeg er " & intAge & " år gammel."
```

```
MsgBox strS
```

Det er altid godt at have orden i din kode, så andre (og - stadigvæk - dig selv) nemt kan læse den igen engang. Fx, ikke skrive for langt ud på en linie - hellere over flere linier. Det gør du ved brug af understrege ("_"). Det vil jeg vise i det næste eksempel, men først skal vi lære lidt om If.

```
If (condition) Then
' Din kode her
End If
```

Sådan skrives en if sætning. Hvor du i condition har noget der tjekker om et eller andet er true eller false.

kode 1.4:

```
Option Explicit
```

```
Private Sub Form_Load()
    Dim intl As Integer
    intl = 7

    If (intl = 7) Then ' Hvis variabelen intl er lig med 7
        MsgBox "intl er 7", vbOkOnly, "Jaa, 7" ' Skriv så "intl er 7" ud i en dialogboks
    End If ' Her slutter If sætningen.
End Sub
```

En if sætning tjekker om noget er sandt eller falsk, og handler efter det. Hvis vi havde sat intl til at være noget andet end 7, ville alt fra "If (intl = 7) Then" og til "End If" blive sprunget over.

Men som altid så må en if sætning have en else. Det har den også. Den tjekker om noget er rigtigt, og hvis det er, udfører den et stykke kode - eller et andet stykke kode. Sådan her:

kode 1.4.1:

```
---  
Option Explicit  
  
Private Sub Form_Load()  
    Dim intl As Integer  
    intl = 7  
  
    If (intl = 7) Then ' Hvis variabelen intl er lig med 7  
        MsgBox "intl er 7", vbOkOnly, "Jaa, 7" ' Skriv så "intl er 7" ud i en dialogboks  
    Else ' Ellers hvis variabelen intl ikke er lig med 7  
        MsgBox "intl er ikke 7", vbOkOnly, "Øv!" ' Skriv så "intl er ikke 7" ud i en dialogboks  
    End If ' her slutter If sætningen  
End Sub  
---
```

Du kan så også specificere det lidt med Elself. Sådan her:

kode 1.4.2:

```
---  
Option Explicit  
  
Private Sub Form_Load()  
    Dim intl As Integer  
    intl = 7  
  
    If (intl = 7) Then ' Hvis variabelen intl er lig med 7  
        MsgBox "intl er 7", vbOkOnly, "Jaa, 7" ' Skriv så "intl er 7" ud i en dialogboks  
    Elself (intl = 8) Then ' Ellers hvis variabelen intl er lig med 8  
        MsgBox "intl er ikke 7, men 8", vbOkOnly, "Oh well.." ' Skriv så denne tekst i en dialogboks  
    Else ' Hvis variabelen intl ikke er lig med 7 (eller 8)  
        MsgBox "intl er ikke 7 eller 8", vbOkOnly, "Øv!" ' Skriv så "intl er ikke 7" ud i en dialogboks  
    End If ' her slutter If sætningen  
End Sub  
---
```

Her har vi If, Elself, Else og så End If til sidst. Koden bliver gennemgået fra toppen ned i bunden, så den tjekker først om intl er lig med 7. Hvis den ikke er det, spring så ned til Elself og tjek om den er 8. Hvis den heller ikke er det, spring så ned til Else og skriv den kode der står mellem Else og End If.

Du kan også bruge And og/eller Or i dine if sætninger for at gøre det mere detaljeret.

kode 1.4.3:

```
---  
Option Explicit  
  
Private Sub Form_Load()  
    Dim intl As Integer  
    Dim intj As Integer  
    intl = 7  
    intj = 8
```

```

If (intl = 7) And (intj = 8) Then ' Hvis variabelen intl er lig med 7, og intj er lig med 8
  MsgBox "intl er 7 og intj er 8", vbOkOnly, "Jaa, 7-8" ' Skriv så "intl er 7 og intj er 8" ud i en dialogboks
Elseif (intl = 8) Or (intj = 1) Then ' Ellers hvis variabelen intl er lig med 8, ELLER intj er lig med 1
  MsgBox "intl er 8, eller er intj 1", vbOkOnly, "Oh well.." ' Skriv så denne tekst i en dialogboks
Else ' Hvis variabelen intl ikke er lig med 7 (eller 8) og intj ikke er 8 (eller 1)
  MsgBox "intl er ikke 7 eller 8, og intj er ikke 8 eller 1", vbOkOnly, "Øv!" ' Skriv så "intl er ikke 7" ud i en
dialogboks
End If ' her slutter If sætningen
End Sub
---
```

Du kan også bruge Not sådan her:

kode 1.4.4:

```

---
Option Explicit

Private Sub Form_Load()
  Dim intl As Integer
  Dim intj As Integer
  intl = 7
  intj = 8

  If (intl = 7) And Not (intj = 8) Then ' Hvis variabelen intl er lig med 7, og intj ikke er lig med 8
    MsgBox "intl er 7 og intj er ikke 8", vbOkOnly, "Jaa, 7" ' Skriv så denne tekst i en dialogboks
  Elseif (intl = 8) Then ' Ellers hvis variabelen intl er lig med 8
    MsgBox "intl er ikke 7, men 8", vbOkOnly, "Oh well.." ' Skriv så denne tekst i en dialogboks
  Else ' Hvis variabelen intl ikke er lig med 7 (eller 8) og intj er 8
    MsgBox "intl er ikke 7 eller 8, intj er 8", vbOkOnly, "Øv!" ' Skriv så denne tekst i en dialogboks
  End If ' her slutter If sætningen
End Sub
---
```

Nu vil jeg gerne vise hvordan man bruger underscore til at sætte en tekst over flere linier. Det kan være meget nyttigt hvis man bruger lange linier af koder.

```

---
Label1.Caption = "Dette er en tekst, " & _
  "som skrives over " & _
  "flere linier" ' Her kan den godt
```

Du behøver ikke at bruge mellemrum der. Det er bare noget jeg gør for at lettere kunne overskue koden. Du kan også skrive sådan:

```

---
Label1.Caption = "Dette er en tekst, " & _
"som skrives over flere linier"
```

Du kan bruge underscore alle steder, hvis bare brugt på den rette måde. Ampersand (&) skal ikke altid sættes ind. Kun hvis du skal sætte flere strings eller variabler sammen eller lignende, som kunne kræve en ampersand. Du kan fx gøre sådan:

```

---
If Not (intl = 0) And Not (intl = 1) _
And Not (intl = 2) And Not (intl = 3) _
And Not (intl = 4) And Not (intl = 5) _
And Not (intl = 6) And Not (intl = 7) _
```

```
And Not (intl = 8) And Not (intl = 9) Then
  MsgBox "Tallet er 10 eller højere"
End If
```

Som så kan gøres meget nemmere og mere overskueligt.

Du kan bruge forskellige "operators". Nu vil jeg forklare de forskellige "operators".

Logical operators:

```
Operator | Hvad den står for | Eksempel | True/False
= | Lig med | (1 = 1) | True
> | Mere End | (1 > 2) | False
< | Mindre End | (1 < 2) | True
>= | Mere End Eller Lig Med | (3 >= 3) | True
<= | Mindre End Eller Lig Med | (4 <= 5) | False
<> | Ikke Lig Med | (5 <> 8) | True
```

If operators:

```
Operator | Hvad den står for
And | Begge sider skal være "True"
Or | En af siderne skal være "True"
Xor | Kun en af siderne skal være "True"
Not | Får "True" til at være "False"
```

Så koden ovenover kan skrives meget nemmere sådan her:

```
---
If (intl >= 10) Then
  MsgBox "Tallet er 10 eller højere"
End If
```

Nu burde du kunne alt det basiske af Visual Basic. Håber du kunne lide min artikel/tutorial. :) God Jul!

Kommentar af antikat d. 20. Dec 2005 | 1

Nu har jeg været inde og hente Visual Basic 2005 Express Edition fra <http://msdn.microsoft.com/vstudio/express/support/install/> - og har sidet at leget lidt med det men kan nu ikke rigtig få det til at lave noget som helst andet end nogle fejl meldinger.. er det, det rigtige program at kaste de kode stumper ind i som du har lavet..og er det så bare mig som gør noget forkert.. for så vil jeg mene der mangler lidt mere info om hvad der skal bruges af programmer og fremgangs måden for en helt Nybegynder.. mvh Daniel

Kommentar af fqthjoe d. 27. Dec 2005 | 2

Fin artikel

Kommentar af shjorth d. 12. Oct 2007 | 3

Kommentar af krydset d. 24. Dec 2005 | 4

fin artikel, jeg har osse hentet vb expres og det her gav mig da lidt og starte med :-D giver den karakter god fordi den ikke går helt i dybden :-D

Kommentar af kasper_knaldperle d. 20. Dec 2005 | 5

Jeg syntes også det er fin artikel. Dog mangler der (når nu vi starter helt fra begynderniveau) lidt forklaring om hvor man skal skrive koden helt i starten. Her tænker jeg hvis man åbner VB for første gang.

Men jeg kan godt lide tonen og stilen i artiklen c",)

Kommentar af baxos d. 07. Jan 2006 | 6

Fed artikle

Kommentar af osgodenhubbe d. 28. Dec 2005 | 7

rigtig god artikel

Kommentar af iostream d. 24. Dec 2005 | 8

Det er jo fordi Visual Basic 2005 EE ikke har noget med VB at gøre. Det er VB.NET.
Jeg synes det er en udemærket artikel !

Kommentar af nillinx d. 20. Dec 2005 | 9

Meget fin artikel :D

Kommentar af ncoder d. 12. Jan 2006 | 10

jeg synes at du har gjort et godt job med denne artikel

Kommentar af buf d. 27. Jul 2007 | 11

Kommentar af hansoc d. 05. Mar 2008 | 12

Kommentar af Binabo d. 09. Aug 2012 | 13

hvis jeg har lavet en textbox og så en knap så jeg trykker på den så blinker min textbox, men før det har jeg sat et password på knappen og hvis jeg skriver koden forkert så skriver den wrong password, så tænkte jeg om der var en som kender koden til så hele min form1 blinker