



Afrund til helt tal

I denne artikel vil du ved hjælp af arrays kunne afrunde et decimaltal til et helt tal.

Skrevet den **03. Feb 2009** af **tjomsen** I kategorien **Programmering / Visual Basic .NET**

Før vi går i gang

For at forstå artiklen ordentlig, er det vigtigt, at du ved lidt om, hvad et array er og kan, og at du ved, hvad et if/then og select case statement er, da jeg ikke vil gå så meget i dybden med det.

Første skridt

Lad os starte med at lave en funktion. Skriv følgende:

```
Function Afrund(ByVal Tal As Decimal) "Tal" er det tal som skal afrundes  
End Function
```

Funktionen er lavet og første skridt er taget. Nu mangler vi kun at smide noget indhold i den.

Lad os springe videre til det sjove

Jeg fylder lige lidt indhold i den og kommenterer bagefter.

```
Function Afrund(ByVal Tal As Decimal)  
  
    Dim Facit As Integer  
    Dim arr As Array = Split(Tal, ",")  
  
End Function
```

Først har jeg erklæret er variabel, som skal indeholde det endelige resultat. Derefter har jeg lavet et array, som deler tallet ved kommaet.

"arr(0)" kommer til at indeholde det tal, som står til venstre for kommaet, altså det hele tal, og "arr(1)" kommer til at indeholde tallet til højre for kommaet, altså decimal-tallet.

Nu skal vi have fundet ud af, hvor mange decimaler der er bag kommaet. Til det bruger vi et 'select case' statement.

```
Function Afrund(ByVal Tal As Decimal)  
  
    Dim Facit As Integer  
    Dim arr As Array = Split(Tal, ",")
```

```
'arr(0) er det hele tal
'arr(1) er decimal-tallet, altså tallet bag kommaet
```

```
Select Case arr(1).Length
```

```
Case 1
```

```
Case 2
```

```
Case 3
```

```
Case 4
```

```
Case 5
```

```
Case Else
```

```
End Case
```

```
End Function
```

Som sagt skal vi have fundet ud af, hvor mange decimaler der er bag kommaet. Det gør vi ved at finde længden af "arr(1)", som jo indeholder decimalerne bag kommaet. Under "Case 1" skal der stå det, der skal ske, hvis der er ét decimal. Under "Case 2" skal der stå det, der skal ske, hvis der er to decimaler bag kommaet. Osv. Osv.

Nu skal vi finde ud af hvor stort tallet er. Lad os sige, at der er ét decimal bag kommaet. Så skal vi funde ud af, om tallet er større eller lig med fem. Er tallet under fem skal tallet rundes ned, eller hvis tallet er større eller lig med fem, skal det selvfølgelig rundes op.

Men hvad nu hvis der er to decimaler bag kommaet? Så er det jo pludselig ikke fem vi skal sammenligne det med. Nej, så er det 50, og hvis der er tre decimaler, så skal vi sammenligne med 500. Det skal hele tiden stige med ti gange så meget for hver decimal der er. (Hvis du kan finde ud af matematik, så er det faktisk logisk nok :o))

Men nok om det, lad os komme videre med koden.

```
Function Afrund(ByVal Tal As Decimal)
```

```
Dim Facit As Integer
```

```
Dim arr As Array = Split(Tal, ",")
```

```
'arr(0) er det hele tal
```

```
'arr(1) er decimal-tallet, altså tallet bag kommaet
```

```
Select Case arr(1).Length
```

```
Case 1
```

```
    If arr(1) >= 5 Then
```

```
        Facit = arr(0) + 1
```

```
    Else
```

```
        Facit = arr(0)
```

```
    End If
```

```
Case 2
```

```
    If arr(1) >= 50 Then
```

```

        Facit = arr(0) + 1
    Else
        Facit = arr(0)
    End If

Case 3
    If arr(1) >= 500 Then
        Facit = arr(0) + 1
    Else
        Facit = arr(0)
    End If

Case 4
    If arr(1) >= 5000 Then
        Facit = arr(0) + 1
    Else
        Facit = arr(0)
    End If

Case 5
    If arr(1) >= 50000 Then
        Facit = arr(0) + 1
    Else
        Facit = arr(0)
    End If

    'osv. osv. osv. osv....

Case Else
    MessageBox.Show("Der er for mange decimaler")
End Case
Afrund = Facit
End Function

```

Som du kan se, så har jeg lavet et if/then statement. Det tjekker hvor stort decimal-tallet er i forhold til hvor mange decimaler der er. Hvis tallet er større eller lig med det angivne, så skal det hele tal (arr(0)) rundes op og Facit (variablen vi lavede øverst) bliver det hele tal + 1. Hvis decimal-tallet er under det angivne, så skal det hele tal bare rundes ned, og Facit bliver det hele tal. Hvis antallet af decimaler er andet end 1-5, så håndterer "Case Else" det. Jeg har blot lavet en beskedbox med teksten "Der er for mange decimaler".

Vi slutter af med at tildele funktionen samme værdi som Facit.

Ja, så er funktionen faktisk færdig. Du kan altid selv udvide funktionen, hvis du vil have, at den skal kunne håndtere mere end fem decimaler.

Du kan afprøve funktionen ved f.eks. at skrive følgende i Button1_Click:

```

Label1.Text = Afrund("5,365")
eller
Label1.Text = Afrund(TextBox1.Text)

```

Nedenfor er hele koden, hvori hver enkel linje er kommenteret.

```

Function Afrund(ByVal Tal As Decimal) 'Her starter funktionen
  Dim Facit As Integer 'Vi laver lige en variabel som skal indeholde facit
  Dim arr As Array = Split(Tal, ",") 'Vi erklærer et array som splitter
  'komma-tallet i to

  'arr(0) er det hele tal
  'arr(1) er decimal-tallet, altså tallet bag kommaet

  Select Case arr(1).Length 'Vi skal tjekke hvor mange decimaler der er bag kommaet

    Case 1 'Hvis der er ét decimal bag kommaet...
      If arr(1) >= 5 Then 'Hvis decimal-tallet er større eller lig med fem...
        Facit = arr(0) + 1 '...så skal tallet rundes op og facit bliver
        'det hele tal + 1
      Else 'Eller hvis decimal-tallet er mindre end fem...
        Facit = arr(0) '...så skal tallet rundes ned og facit bliver det hele tal
      End If

    Case 2 'Hvis der er to decimaler bag kommaet...
      If arr(1) >= 50 Then 'Hvis decimal-tallet er større eller lig med 50...
        Facit = arr(0) + 1 '...så skal tallet rundes op og facit bliver
        'det hele tal + 1
      Else 'Eller hvis decimal-tallet er mindre end 50...
        Facit = arr(0) '...så skal tallet rundes ned og facit bliver det hele tal
      End If

    Case 3 'Hvis der er tre decimaler bag kommaet...
      If arr(1) >= 500 Then 'Hvis decimal-tallet er større eller lig med 500...
        Facit = arr(0) + 1 '...så skal tallet rundes op og facit bliver
        'det hele tal + 1
      Else 'Eller hvis decimal-tallet er mindre end 500...
        Facit = arr(0) '...så skal tallet rundes ned og facit bliver det hele tal
      End If

    Case 4 'Hvis der er fire decimaler bag kommaet...
      If arr(1) >= 5000 Then 'Hvis decimal-tallet er større eller lig med 5.000...
        Facit = arr(0) + 1 '...så skal tallet rundes op og facit bliver
        'det hele tal + 1
      Else 'Eller hvis decimal-tallet er mindre end 5.000...
        Facit = arr(0) '...så skal tallet rundes ned og facit bliver det hele tal
      End If

    Case 5 'Hvis der er fem decimaler bag kommaet...
      If arr(1) >= 50000 Then 'Hvis decimal-tallet er større eller lig med 50.000...
        Facit = arr(0) + 1 '...så skal tallet rundes op og facit bliver
        'det hele tal + 1
      Else 'Eller hvis decimal-tallet er mindre end 50.000...
        Facit = arr(0) '...så skal tallet rundes ned og facit bliver det hele tal
      End If

    'osv. osv. osv. osv....

    Case Else 'Hvis der er over fem decimaler...
      MessageBox.Show("Der er for mange decimaler", "Ups") '...så får du lige besked
  End Select

```

```
End Select
Afrund = Facit 'Vi slutter af med at tildele funktionen samme værdi som Facit
End Function 'Her slutter funktionen
```

Kommentar af mysitesolution d. 28. Nov 2007 | 1

Jeg sætter pris på at du forsøger dig på en relativ lang artikel. Men din metode er ekstrem ineffektiv. Og så er der jo allerede en i .Net lib. For det første så konvertere du et "decimal" til en tekst, som du så splitter. Allerede der er den 10* så langsom end den i .Net (estimat). Og så virker den ikke hvis du har mere end 5 cifre :/

Kommentar af arne_v d. 25. Nov 2007 | 2

Math.Round(tal, 0, MidpointRounding.AwayFromZero) var meget nemmere.

Kommentar af cwboy d. 27. Nov 2007 | 3

Selv hvis det skulle være for eksemplets skyld - og man ikke måtte bruge Math.Round, ville jeg mene, at der var mere effektive måder at gøre det på.

Kommentar af qua d. 27. Sep 2008 | 4

Dårlig struktur af kode og algoritme.

Kommentar af happycow d. 04. Jan 2008 | 5

Undskyld, men denne artikel hører til på thedailywtf.com -- specielt siden du tager betaling for den....