

Basic Bulletin

22^{ste} jaargang december 2015

Nummer 4





Inhoud

Onderwerp

blz.

BBC BASIC for Windows – De library's (7).	4
XNA Game Studio 4.0, de Game1 methoden.	6
Liberty BASIC API Reference.	9
Tekenen in GW-BASIC – Een tekenprogramma.	14
Excel – De functies.	16



Contacten

Functie	Naam	Telefoonnr.	E-mail
Voorzitter	Jan van der Linden	071-3413679	voorz@basic-gg.hcc.nl
Secretaris	Gordon Rahman Tobias Asserstraat 6 2037 JA Haarlem	023-5334881	secre@basic-gg.hcc.nl
Penningmeester	Piet Boere	0348-473115	penm@basic-gg.hcc.nl
Bestuurslid	Titus Krijgsman	075-6145458	t.krijgsman8@upcmail.nl
Redacteur	M.A. Kurvers Schaapsveld 46 3773 ZJ Barneveld	06-30896598	m.a.kurvers@live.nl
Ledenadministratie	Fred Luchsinger	0318-571187	f.luchsinger@kader.hcc.nl
Webmaster	Jan van der Linden	071-3413679	j.vd.linden@kader.hcc.nl

<http://www.basic.hcc.nl>



Redactioneel

In het artikel over XNA Game Studio wordt verder gegaan met de bal. Een bal alleen linksboven weergegeven is natuurlijk niet genoeg. In dit onderwerp laat ik zien hoe we een bal kunnen laten bewegen en laten botsen tegen de randen van het scherm.

Alleen maar een sprite gebruiken is niet altijd handig. Door er een object van te maken, kunnen we meerdere instanties gebruiken zodat we meer dan één bal kunnen laten bewegen.

In het GWBASIC artikel gaat het over het DRAW statement.

Met het DRAW statement kunnen we veel doen. In dit onderwerp kunt u een voorbeeldprogramma intypen en uitvoeren. Hiermee kunt u de indruk krijgen wat we (toen) in GW-BASIC allemaal konden doen en jammer is het zeker wel dat we deze mogelijkheden in de nieuwe Basic versies moeten missen.

Marco Kurvers

BBC BASIC for Windows – De library's (7).

Multiple Document Interface

De **MDILIB** bibliotheek bevat een set van procedures en functies voor het gebruik van de Windows™ Interface MDI (Multiple Document). Dit kan een programma openen van meerdere vensters van de uitvoer, meestal zodat de gebruiker met meer dan één document tegelijk of met meer dan één type weergave werken kan. Bijvoorbeeld, een spreadsheetprogramma kan één venster voor een tabelweergave van de gegevens en een ander venster gebruiken voor een grafische weergave.

De bibliotheek moet worden geladen vanuit uw programma met behulp van de opdracht:

```
INSTALL @lib$+"MDILIB"
```

De procedures en functies zijn:

- PROC_initmdi
- PROC_exitmdi
- FN_createmdichild
- FN_hdc
- PROC_closemdichild
- PROC_tile
- PROC_cascade

PROC_initmdi(menu%)

Deze procedure maakt de gegevens op van de Multiple Document Interface. Het vereist één parameter: de handle van het **venster** top-level menu. Elk MDI programma moet een venstermenu hebben, omdat de inhoud van dit menu automatisch door het besturingssysteem gewijzigd wordt als *zoonvensters* worden toegevoegd of worden verwijderd. Daarom moet uw programma met een minimale code er in worden opgenomen:

```
SYS "CreatePopupMenu" TO hwindow%
SYS "CreateMenu" TO hmenu%
SYS "AppendMenu", hmenu%, 16, hwindow%, "&Window"
SYS "SetMenu", @hwnd%, hmenu%
SYS "DrawMenuBar", @hwnd%
PROC_initmdi(hwindow%)
```

PROC_initmdi moet worden aangeroepen slechts eenmaal aan het begin van het programma; zodra het is aangeroepen is geen uitvoer naar het scherm mogelijk totdat u een of meer onderliggende vensters hebt gemaakt (indien nodig, kunt u nog steeds gegevens met behulp van een berichtvenster weergeven).

PROC_exitmdi

Deze procedure sluit de Multiple Document Interface af. Het moet worden aangeroepen voordat uw programma afgesloten wordt en voordat een CLEAR, CHAIN of een RUN statement uitgevoerd wordt; het is dan zeer waarschijnlijk dat *BBC BASIC for Windows* kan crashen. Het is aan te bevelen ON CLOSE en ON ERROR statements in uw programma in te voegen zodat u zeker kunt weten dat **PROC_exitmdi** aangeroepen wordt wanneer het nodig is.

FN_createmdichild(name\$)

Deze functie maakt een zoonvenster (of ook wel genoemd een *onderliggend* venster) aan en geeft het de opgegeven naam; de naam zal worden weergegeven in de titelbalk van het zoonvenster en in het **Window** menu, zie codeblok van **PROC_initmdi**. Als het zoonvenster gemaximaliseerd wordt zal het niet meer een titelbalk hebben, maar de naam is dan automatisch toegevoegd aan de titel van het hoofdprogramma.

Het zoonvenster is gemaakt met een standaard grootte en positie, maar uw programma kan dat volgens wijzigen indien het nodig is. Het kan geminimaliseerd, gemaximaliseerd, vergroot en verkleind en verplaatst worden door de gebruiker op de gebruikelijke manier, behalve dat het venster beperkt blijft binnen de grenzen van het hoofdvenster van uw programma. U kunt zoveel zoonvensters maken als u zelf wilt (binnen redent!).

FN_createmdichild retourneert het *venster handle* van het zoonvenster dat gemaakt is:

```
hwnd1% = FN_createmdichild("Hello world")
hwnd2% = FN_createmdichild("Rectangles")
hwnd3% = FN_createmdichild("Circles")
```

Zodra u een gemaakt zoonvenster heeft, kunt u uitvoer erheen sturen op dezelfde manier als u het naar een normaal BASIC venster stuurt, maar u moet dan wel eerst het juiste venster selecteren door de waarden van **@hwnd%** en **@memhdc%** te veranderen. De gemakkelijkste manier om dat te doen (en om zeker te weten dat zij daarna het systeem terugzetten naar hun oorspronkelijke waarden) is om het door te geven als parameters van een procedure. Dus als voorbeeld, een procedure voor het schrijven van tekst naar een zoonvenster zou er misschien uit kunnen zien als:

```
DEF PROCwritetext(A$, @hwnd%, @memhdc%)
PRINT A$
ENDPROC
```

Om een tekst string naar een soortgelijk venster te schrijven, kunt u de procedure op deze manier aanroepen:

```
PROCwritetext("Some text", hwnd1%, FN_hdc(hwnd1%))
```

(zie onderaan voor een uitleg over **FN_hdc**)

Zoals het er nu ligt zal dit prima werken bij het schrijven van tekst naar slechts één venster, maar als u tekst aan twee of meer vensters gelijktijdig schrijft, zult u zien dat alle vensters dezelfde tekst uitvoerpositie delen, dus een 'newline' verzenden naar één venster zal invloed hebben op de latere positie van tekst naar een ander venster. Een manier om elk zoonvenster te voorzien van een eigen 'privé' tekstpositie is voor het doorgeven van de X- en Y-coördinaten aan de procedure:

```
DEF PROCwritetext(A$, @hwnd%, @memhdc%, RETURN @vdu%!48, RETURN @vdu%!52)
PRINT A$
ENDPROC
```

die u op deze manier kunt aanroepen:

```
PROCwritetext("Text for window 1", hwnd1%, FN_hdc(hwnd1%), xpos1%, ypos1%)
PROCwritetext("Text for window 2", hwnd2%, FN_hdc(hwnd2%), xpos2%, ypos2%)
```

Aangezien ze doorgegeven worden bij referentie, worden de variabelen met de coördinaten van de tekst voor elk venster automatisch bijgewerkt.

Onthoud dat in *BBC BASIC for Windows* de MDI zoonvensters niet de tekstcursor (caret-teken) weer-geven. Wanneer gebruikersinvoer nodig is, moet dat gedaan worden doormiddel van een dialogoven-ster.

FN_hdc(hwnd%)

Deze functie retourneert het *geheugen hdc* behorend aan het opgegeven venster handle. Dit is nodig wanneer een soortgelijk zoonvenster geselecteerd wordt voor uitvoer (het moet gekopieerd worden in **@memhdc%**).

PROC_closemdichild(hwnd%)

Deze procedure sluit het opgegeven zoonvenster. Het heeft hetzelfde effect als wanneer de gebruiker op de sluitknop klikt of op de sneltoets **Ctrl+F4** drukt.

U kunt niet voorkomen dat de gebruiker een onderliggend venster sluit (er is geen direct equivalent op ON CLOSE), maar u kunt wel detecteren dat de gebruiker het gesloten heeft met behulp van de **IsWindow** API-aanroep:

XNA Game Studio 4.0, de Game1 methoden.

De methoden die we tijdens het maken van de sjabloon erin gezet hebben, kunnen we niet zelf noe-men. Deze methoden zijn beschermt. Deze techniek zullen we vaker tegenkomen als we met zulke li-brary's werken. Om de klassen zo veilig mogelijk te houden, worden ze in namespaces gezet zodat ze ingepakt in resources weer in andere projecten hergebruikt kunnen worden. Om schade aan de me-thoden te voorkomen, worden ze daarom beschermd. Vandaar dat de methoden met het sleutelwoord *Protected* moeten beginnen en ook moet u ze overschrijven met het sleutelwoord *Overrides*.

Dit is alleen noodzakelijk voor klassen die A: een basisklasse hebben en B: hergebruikt moeten wor-den. Klassen die in namespaces staan kunnen niet direct worden gebruikt zonder ze eerst te *importe-ren* met het sleutelwoord *Imports*.

Wilt u zelf namespaces maken? Dat kan. U moet dan de klassen in de nieuwe namespace plaatsen en de methoden beschermen. Bovendien is elke namespace een apart project. Projecten die u als klassenlibrary's in uw solution toevoegt, moeten namespaces hebben. De klassen kunt u anders niet in uw hoofdproject hergebruiken.

Verder met VBGame

De vorige keer heb ik laten zien hoe u een speelscherm opzet en hoe u een sprite kunt renderen. Het weergeven van achtergronden en sprites gebeurt altijd in de methode *Draw()*.

Een stilstaande bal is natuurlijk geen bal waard! Daarom gaan we de bal laten rollen over het scherm. Met het Vector2 object kunnen we de voortgang en de positie van de bal bepalen. Als de bal gaat rol-len, kan deze het scherm verlaten. Dat klinkt gek, maar het gebeurt wel. De bal is dan alsnog niet ver-dwenen, maar is in een andere *viewport*. Een viewport is een deel van het hele speeloppervlak waar wij naar kijken. We kunnen het vergelijken met een camera dat een deel van het oppervlak projecteert, net zoals wij met onze ogen naar een richting kijken.

Open het VBGame project met de blauwe bal. Om het gemakkelijk te maken laat ik u nog eens de code zien om de bal in te laden en een positie en de snelheid te bepalen. De variabelen voor de snelheid had echter de vorige keer nog geen zin, omdat de bal op één plaats bleef staan. We hebben een

methode nodig om tijdens de game proces (de uitvoerloop) de bal een andere positie te geven, maar dan wel zo alsof de bal echt rolt. In feite houden we de gebruiker voor de gek. Kijk nog eens naar onderstaande code, die onder de spriteBatch declaratie hoort te staan:

```
'Texture die we willen renderen
Private texture As Texture2D
'Bepaal waar de sprite getekend wordt
Private spritePos As Vector2 = Vector2.Zero
'X en Y snelheid van de sprite
Private XSpeed As Single = 80
Private YSpeed As Single = 120
'Vector2 wordt gebruikt voor de snelheid van de sprite
Private spriteSpeed As New Vector2(XSpeed, YSpeed)
```

Weet u nog het probleem van de Content Pipeline? We kunnen niet de Content gebruiken, zoals we dat wel in Visual C# kunnen doen. Gelukkig is er een oplossing om textures in te kunnen laden. XNA heeft een **FromStream()** methode in de Texture2D klasse, een methode die afkomstig is van het .NET Framework.

```
'Laad de texture.
'Maak gebruik van een Stream om de inhoud in te kunnen laden.
Dim textureStream As IO.Stream = New _
    IO.StreamReader(Application.StartupPath & _
        "\Textures\Bal.png").BaseStream
'Laad de texture in.
texture = Texture2D.FromStream(GraphicsDevice, textureStream)
```

Zonder de methode FromStream zouden we niets kunnen doen met de inhoud.

Merk op dat **Application.StartupPath** wordt gebruikt. De map **Textures** bestaat niet. Maak de map zelf aan en zorg ervoor dat al de inhoud in die map staat. De StartupPath verwijst naar de map waar het uitvoerbare programma in staat. Dat is in de Debug map. Zorg er dus voor dat daar ook de Textures map in staat. De path zal dan zijn: “\VBGame\bin\Debug\Textures”

In de Draw() methode renderen we de sprite op het scherm.

```
'Teken de sprite
spriteBatch.Begin(SpriteSortMode.BackToFront, BlendState.AlphaBlend)
spriteBatch.Draw(texture, spritePos, Color.White)
spriteBatch.End()
```

Eerder zagen we een blauw scherm met de bal linksboven. Om te laten zien dat er veel gebeurt tijdens de uitvoer, gaan we de bal laten bewegen. Daarbij gaan we gelijk controleren of de bal de rand van het scherm raakt. We moeten dan de bal terug laten kaatsen.

Dit alles moet worden gedaan in de **Update()** methode. Hier wordt telkens de controle van het spel uitgevoerd; niet de praktijk zoals het tekenwerk, want dat doet de **Draw()** methode.

Bekijk onderstaande code. Neem het over in de Update() methode waar de TODO commentaarregel staat:

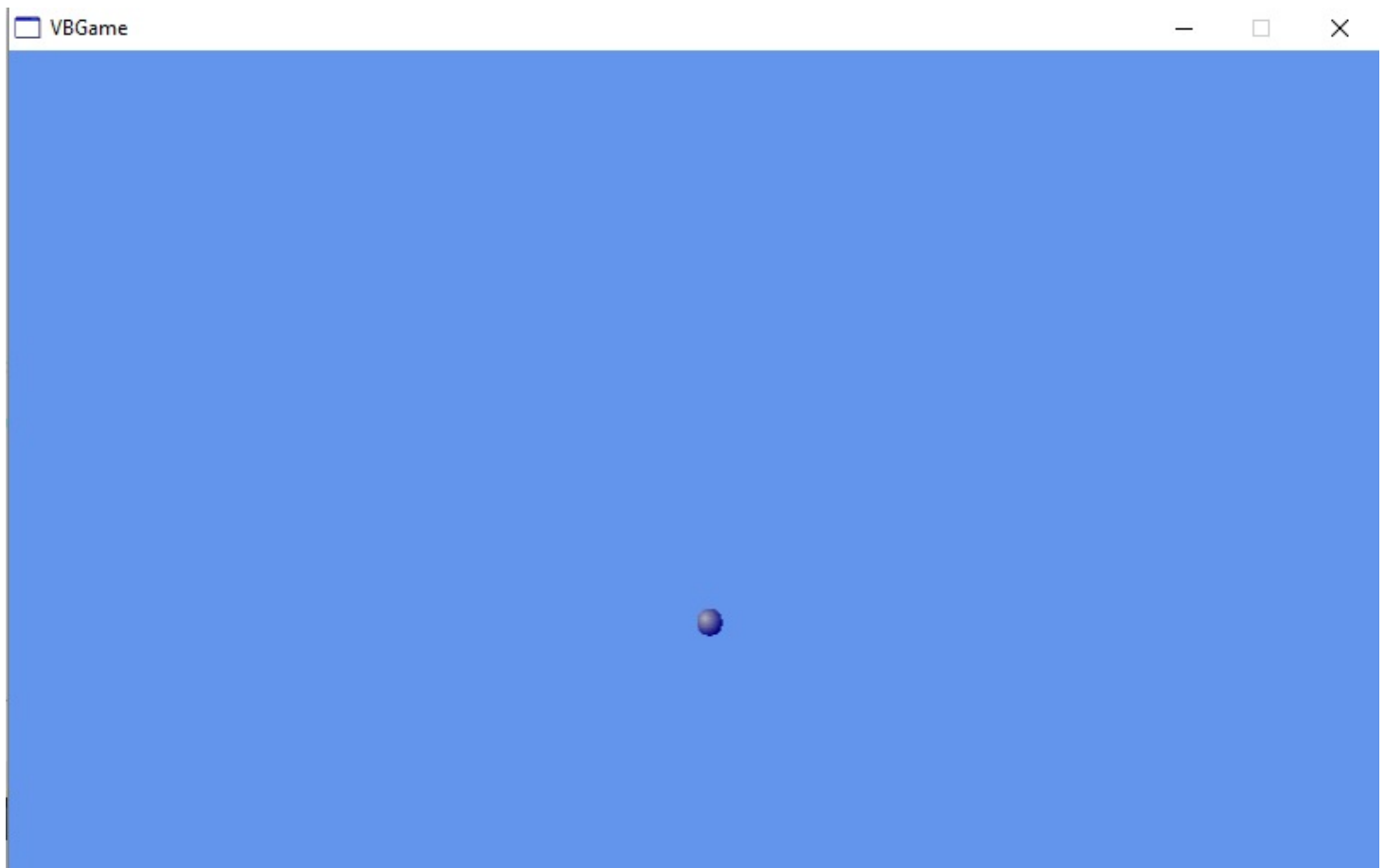
```
spritePos += spriteSpeed * 0.01F
Dim MaxWidth As Integer = graphics.GraphicsDevice.Viewport.Width
Dim MaxHeight As Integer = graphics.GraphicsDevice.Viewport.Height
```

```

If (spritePos.X < 0) Then
    spritePos.X = 0
    spriteSpeed.X *= -1
End If
If (spritePos.X + texture.Width > MaxWidth) Then
    spritePos.X = MaxWidth - texture.Width
    spriteSpeed.X *= -1
End If
If (spritePos.Y < 0) Then
    spritePos.Y = 0
    spriteSpeed.Y *= -1
End If
If (spritePos.Y + texture.Height > MaxHeight) Then
    spritePos.Y = MaxHeight - texture.Height
    spriteSpeed.Y *= -1
End If

```

Hebt u alles goed overgenomen, dan kunt u het programma starten. U ziet een scherm verschijnen met een bal die een bepaalde richting op gaat. Laat de bal even doorgaan totdat de rand onderaan wordt geraakt. De bal kaatst terug omhoog, maar met dezelfde horizontale richting, evenzo voor de verticale richting als de bal de linkerkant en rechterkant raakt. Een voorbeeld ziet u zoals hieronder:



Experimenteer eens met de waarden. Verander de snelheid van de bal door onderstaande regel:

```
spritePos += spriteSpeed * 0.01F
```

te wijzigen in:

```
spritePos += spriteSpeed * 0.1F
```


Als u het programma start, raast de bal met een flinke snelheid over het scherm, maar dankzij de Update() methode blijft de bal in de huidige Viewport.

Wijzig ook eens de richting. Geef bijvoorbeeld de variabele **YSpeed** de waarde 20 en start dan het programma opnieuw. U zult zien dat de bal minder verticaal beweegt en meer horizontaal. Deze Vector waarden zijn dus belangrijk om sprites bepaalde richtingen te geven. Alleen maar een snelheid is niet voldoende om te kunnen bewegen.

Volgend jaar laat ik u zien hoe we meer sprites kunnen renderen en hoe we het als objectinstanties kunnen definiëren. Een leuke arkanoid game komt er dan als voorbeeld en natuurlijk komt er uitleg bij hoe de code in elkaar zit.

Liberty BASIC API Reference.

Systeminformatie ophalen

U kunt informatie over het systeem ophalen met API aanroepen. Dit onderwerp laat zien hoe u de systeemtijd, systeem statistieken en de power status kunt ophalen.

GetSystemTime

U mag de GetSystemTime functie gebruiken om de dag en datum informatie op te halen. De functie heeft een SYSTEMTIME struct nodig.

```
Struct SYSTEMTIME, _
    wYear As short, _
    wMonth As short, _           'Januari=1, Februari=2, enz.
    wDayOfWeek As short, _      'zondag=0, maandag=1, enz.
    wDay As short, _
    wHour As short, _
    wMinute As short, _
    wSecond As short, _
    wMilliseconds As short
```

```
CallDLL #kernel32, "GetSystemTime", SYSTEMTIME As Struct, ret As Long
```

```
print "Jaar is "; SYSTEMTIME.wYear.struct
print "Maand is "; SYSTEMTIME.wMonth.struct
print "Dag van de week is "; SYSTEMTIME.wDayOfWeek.struct
print "Dag is "; SYSTEMTIME.wDay.struct
print "Uur is "; SYSTEMTIME.wHour.struct
print "Minuut is "; SYSTEMTIME.wMinute.struct
```

```
Wait
```

GetSystemMetrics

U mag de GetSystemMetrics API functie gebruiken om informatie over het gebruikerssysteem te vinden.

```
calldll #user32, "GetSystemMetrics", _
    nIndex as long, _ 'vlag voor info gewenst
```

result as long 'retourneert opgevraagde informatie

In onderstaand voorbeeld wordt de functie gebruikt om de hoogte van een titelbalk van een venster op te halen.

```
'ken de hoogte van de titelbalk toe  
call dll #user32, "GetSystemMetrics", _  
    _SM_CYCAPTION as long, _          'ken de titelbalkhoogte toe  
    CaptionHeight as long            'retourneert hoogte in pixels van de  
                                     'titelbalk  
                                     'retourneert 0 als niet succesvol
```

Hier zijn de mogelijke waarden voor het nIndex argument.

<code>_SM_ARRANGE</code>	Vlaggen die bepalen hoe het systeem de geminimaliseerde vensters rangschikt.
<code>_SM_CLEANBOOT</code>	Waarde dat bepaald hoe het systeem opgestart is: 0: Normale start 1: Mislukt-veilige start 2: Mislukt-veilige start met netwerk Mislukt-veilige start (ook aangeropen als SafeBoot) geeft de gebruikers opstartbestanden door.
<code>_SM_CMOUSEBUTTONS</code>	Aantal knoppen van de muis of nul als er geen muis geïnstalleerd is.
<code>_SM_CXBORDER</code> <code>_SM_CYBORDER</code>	De breedte en hoogte in pixels van een vensterborder. Dit is hetzelfde als de <code>_SM_CXEDGE</code> waarde voor de vensters met een 3D zicht.
<code>_SM_CXCURSOR</code> <code>_SM_CYCURSOR</code>	De breedte en hoogte in pixels van een cursor. Dit zijn de cursor dimensies ondersteund bij de huidige weergave driver. Het systeem kan geen cursors maken van andere grootte.
<code>_SM_CXDLGFRAME</code> <code>_SM_CYDLGFRAME</code>	Zelfde als <code>_SM_CXFIXEDFRAME</code> en <code>_SM_CYFIXEDFRAME</code> .
<code>_SM_CXDOUBLECLK</code> <code>_SM_CYDOUBLECLK</code>	De breedte en hoogte in pixels van de rechthoek om de locatie heen van een eerste klik in een dubbelklik volgorde. De tweede klik moet zich voordoen binnen deze rechthoek voor het systeem om te overwegen dat de twee kliks een dubbelklik is. (De twee klikken moeten ook plaatsvinden binnen een bepaalde tijd.)
<code>_SM_CXDRAG</code> <code>_SM_CYDRAG</code>	De breedte en hoogte in pixels van een rechthoek gecentreerd op een sleppunt om toe te staan voor een beperkte beweging van de muis voordat een sleep operatie begint. Hierdoor kan de gebruiker klikken en de muisknop loslaten en gemakkelijk onbedoeld een bewerking slepen.
<code>_SM_CXEDGE</code> <code>_SM_CYEDGE</code>	De dimensies in pixels van een 3D border. Deze zijn de collega's van <code>_SM_CXBORDER</code> en <code>_SM_CYBORDER</code> .
<code>_SM_CXFIXEDFRAME</code>	De dikte in pixels van het frame rond de omtrek van een venster

<code>_SM_CYFIXEDFRAME</code>	dat een bijschrift heeft maar niet omvangrijk is. <code>_SM_CXFIXEDFRAME</code> is de breedte van de horizontale border en <code>_SM_CYFIXEDFRAME</code> is de hoogte van de verticale border. Zelfde als <code>_SM_CXDLGFRAME</code> en <code>_SM_CYDLGFRAME</code> .
<code>_SM_CXFRAME</code> <code>_SM_CYFRAME</code>	Zelfde als <code>_SM_CXSIZEFRAME</code> en <code>_SM_CYSIZEFRAME</code> .
<code>_SM_CXFULLSCREEN</code> <code>_SM_CYFULLSCREEN</code>	De breedte en hoogte van het clientgebied voor een volledig venster. Als u de coördinaten van het gedeelte van het scherm niet verduisterd door de lade, roept u de functie <code>SystemParametersInfo</code> aan met de <code>_SPI_GETWORKAREA</code> -waarde.
<code>_SM_CXHSCROLL</code> <code>_SM_CYHSCROLL</code>	De breedte in pixels van de pijl bitmap op een horizontale schuifbalk en de hoogte in pixels van een horizontale schuifbalk.
<code>_SM_CXHTHUMB</code>	De breedte in pixels van het duim vak in een horizontale schuifbalk.
<code>_SM_CXICON</code> <code>_SM_CYICON</code>	De standaard breedte en hoogte in pixels van een icoon. Deze waarden zijn typisch 32x32, maar kan variëren afhankelijk van de geïnstalleerde weergave hardware. De <code>LoadIcon</code> functie kan alleen iconen laden van deze dimensies.
<code>_SM_CXICONSPACING</code> <code>_SM_CYICONSPACING</code>	De dimensies in pixels van een raster cel voor items in grote pictogramweergave. Elk item past in een rechthoek van deze grootte als het geordend is. Deze waarden zijn altijd groter dan of gelijk aan <code>_SM_CXICON</code> en <code>_SM_CYICON</code> .
<code>_SM_CXMAXIMIZED</code> <code>_SM_CYMAXIMIZED</code>	De standaard dimensies in pixels van een gemaximaliseerd venster op het hoogste niveau.
<code>_SM_CXMAXTRACK</code> <code>_SM_CYMAXTRACK</code>	De standaard maximale dimensies in pixels van een venster dat een beschrijving en formaatgrepen heeft. De gebruiker kan niet het venster frame slepen dat groter is dan deze afmetingen. Een venster kan deze waarden overschrijven door het <code>WM_GETMINMAXINFO</code> bericht te laten verwerken.
<code>_SM_CXMENUCHECK</code> <code>_SM_CYMENUCHECK</code>	De dimensies in pixels van het standaard menu vinkje bitmap.
<code>_SM_CXMENUSIZE</code> <code>_SM_CYMENUSIZE</code>	De dimensies in pixels van de menubalk knoppen, zoals een sluitknop voor het multiple document (MDI) zoonvenster.
<code>_SM_CXMIN</code> <code>_SM_CYMIN</code>	De minimale breedte en hoogte in pixels van een venster.
<code>_SM_CXMINIMIZED</code> <code>_SM_CYMINIMIZED</code>	De dimensies in pixels van een normale geminimaliseerd venster.
<code>_SM_CXMINSPPACING</code> <code>_SM_CYMINSPPACING</code>	De dimensies in pixels van een raster cel voor geminimaliseerde vensters. Elke geminimaliseerd venster past in een rechthoek van deze grootte, wanneer gerangschikt. Deze waarden zijn altijd groter dan of gelijk aan <code>_SM_CXMINIMIZED</code> en <code>_SM_CYMINIMIZED</code> .

_SM_CXMINTRACK _SM_CYMINTRACK	Het minimum bijhouden van de breedte en hoogte in pixels van een venster. De gebruiker kan niet het venster frame slepen tot een grootte die kleiner is dan deze afmetingen. Een venster kan deze waarden overschrijven door het WM_GETMINMAXINFO bericht te laten verwerken.
_SM_CXSCREEN _SM_CYSCREEN	De breedte en hoogte in pixels van het scherm.
_SM_CXSIZE _SM_CYSIZE	De breedte en hoogte in pixels van een knop in een beschrijving van een venster of titelbalk.
_SM_CXSIZEFRAME _SM_CYSIZEFRAME	De dikte in pixels van het frame rond de omtrek van een venster dat aangepast kan worden. _SM_CXSIZEFRAME is de breedte en _SM_CYSIZEFRAME is de hoogte. Zelfde als _SM_CXFRAME en _SM_CYFRAME.
_SM_CXSMICON _SM_CYSMICON	Aanbevolen dimensies in pixels van een klein pictogram (icoon). Kleine pictogrammen verschijnen typisch in vensterbeschrijvingen en in kleine pictogrambeelden
_SM_CXSMSIZE _SM_CYSMSIZE	De dimensies in pixels van kleine beschrijvingsknoppen.
_SM_CXVSCROLL _SM_CYVSCROLL	De breedte in pixels van een verticale schuifbalk en de hoogte in pixels van de pijl bitmap op een verticale schuifbalk.
_SM_CYCAPTION	De hoogte in pixels van een normaal beschrijvingsgebied.
_SM_CYKANJIWINDOW	Voor dubbel-byte karakterset versies van Windows wordt de hoogte in pixels van het Kanji venster aan de onderkant van het scherm bepaald.
_SM_CYMENU	De hoogte in pixels van een enkele-lijn menubalk.
_SM_CYSMCAPTION	De hoogte in pixels van een kleine beschrijving.
_SM_CYVTHUMB	De hoogte in pixels van een duim vak in een verticale schuifbalk.
_SM_DBCSENABLED	Is WAAR of niet-nul als de dubbel-byte karakterset (DBCS) versie van USER.EXE geïnstalleerd is, anders ONWAAR of nul.
_SM_DEBUG	Is WAAR of niet-nul als de foutopsporing versie van USER.EXE geïnstalleerd is, anders ONWAAR of nul.
_SM_MENUDROPALIGNMENT	Is WAAR of niet-nul als de drop-down menu's rechts uitgelijnd zijn, relatief aan het gecorrespondeerde menubalk-item, anders ONWAAR of nul als ze links uitgelijnd zijn.
_SM_MIDEASTENABLED	Is WAAR als het systeem ingeschakeld is voor Hebreeuwse/Arabishe talen
_SM_MOUSEPRESENT	Is WAAR of niet-nul als een muis geïnstalleerd is, anders ONWAAR of nul.

<code>_SM_MOUSEWHEELPRESENT</code>	Alleen Windows NT: Is WAAR of niet-nul als een muis met een wiel geïnstalleerd is, anders ONWAAR of nul.
<code>_SM_NETWORK</code>	De minste significante bit is ingesteld als een netwerk aanwezig is; anders, het is uitgeschakeld. De andere bits zijn gereserveerd voor toekomstig gebruik.
<code>_SM_PENWINDOWS</code>	Is WAAR of niet-nul als de Microsoft Windows for Pen computing extensies geïnstalleerd zijn, anders ONWAAR of nul.
<code>_SM_SECURE</code>	Is WAAR als bescherming aanwezig is, anders ONWAAR.
<code>_SM_SHOWSOUNDS</code>	Is WAAR of niet-nul als de gebruiker een applicatie vereist op de huidige informatie visueel in situaties, waar het anders de informatie uitsluitend in hoorbare vorm zou presenteren, anders ONWAAR of nul.
<code>_SM_SLOWMACHINE</code>	Is WAAR als de computer een low-end (langzame) processor heeft, anders ONWAAR.
<code>_SM_SWAPBUTTON</code>	Is WAAR of niet-nul als de betekenissen van de linker en rechter muisknoppen verwisseld zijn, anders ONWAAR of nul.

Tip! Veel documentaties in boeken en internetpagina's laten vaak de onderstreping vooraan weg.

GetSystemPowerStatus

De `GetSystemPowerStatus` functie is handig voor computers die op batterijen werken, zoals laptop computers. Het vereist een `SYSTEMPOWERSTATUS` struct.

```
struct SYSTEMPOWERSTATUS, _
    ACLineStatusBatteryFlag as word, BatteryLifePercent as word, _
    BatteryLifeTime as ulong, BatteryFullLifeTime as ulong
```

Zodra de functie resulteert, ontvangt de struct informatie over de systeempower als volgt:

<code>ACLineStatus</code>	0=offline 1=online 255=onbekend
<code>BatteryFlag</code>	1 hoog 2 laag 4 kritisch 8 opladen 128 geen systeembatterij 255 onbekende status
<code>BatteryLifePercent</code>	0-100% of 255 als de status onbekend is
<code>BatteryLifeTime</code>	Aantal resterende seconden van de batterij, of hexdec("FFFFFFFF") als de resterende seconden onbekend zijn.
<code>BatteryFullLifeTime</code>	Aantal seconden van de batterij resterend wanneer deze opgeladen is, of hexdec("FFFFFFFF") als de volle batterijtijd onbekend is.

De eerste twee members voor `ACLineStatus` en `BatteryFlag` zijn elk van het type `BYTE`, dus we geven het door als een single member van het type `WORD` om het in Liberty BASIC gemakkelijk te maken.

Om de eerste byte van het geroutineerde woord op te halen, gebruiken we de functie MOD en om de tweede byte op te halen, gebruiken we de integer deling, zoals dit:

```
firstByte = returnedWord MOD 256
secondByte = int(returnedWord / 256)
```

Hieronder is wat code dat de functie demonstreert:

```
struct SYSTEMPOWERSTATUS, _
    ACLineStatusBatteryFlag as word, _
    BatteryLifePercent as word, _
    BatteryLifeTime as ulong, _
    BatteryFullLifeTime as ulong

call dll #kernel32, "GetSystemPowerStatus", _
    SYSTEMPOWERSTATUS as struct, re as boolean

ACBF = SYSTEMPOWERSTATUS.ACLineStatusBatteryFlag.struct
print "ACLineStatus is "; ACBF MOD 256
print "BatteryFlag is "; int(ACBF / 256)
print "BatteryLifePercent is "; SYSTEMPOWERSTATUS.BatteryLifePercent.struct
print "BatteryLifeTime in seconds is ";
print SYSTEMPOWERSTATUS.BatteryLifeTime.struct
print "BatteryFullLifeTime in seconds is ";
print SYSTEMPOWERSTATUS.BatteryFullLifeTime.struct
```

In de volgende Bulletin komt het onderwerp over **Shell en Kernel API**. Na dit theoretisch onderwerp zal het volgende onderwerp wat meer praktijkgericht zijn, maar u moet dan wel zeker weten dat u begrijpt hoe de API aanroepingen gemaakt worden. Er zullen voorbeelden bij zijn die de API's demonstren.

Wordt vervolgd

Tekenen in GW-BASIC – Een tekenprogramma.

Om de mogelijkheden van het DRAW statement nog eens goed te demonstreren, kunt u onderstaand programma uitproberen. Dit programma maakt het de gebruiker mogelijk om met behulp van aantal toets indrukken een tekening op het scherm te laten verschijnen. Elke toets heeft zijn eigen van te voren gedefinieerde functie. Het programma is slechts een voorbeeld. Het kan op allerlei plaatsen naar eigen ideeën en wensen aangepast worden. De kern van het programma bestaat uit de regels 130 tot en met 170. In regel 160 gaat het programma naar de subroutine in regel 210, waar om een toets indruk wordt gevraagd; en afhankelijk daarvan wordt een opdracht uitgevoerd. De toetsen hebben de volgende functie:

Betekenis van de toetsen:

o	=	10 pixels omhoog	g	=	10 pixels linksonder
n	=	10 pixels naar beneden	h	=	10 pixels linksboven
l	=	10 pixels naar links	v	=	teken een vierkant
r	=	10 pixels naar rechts	d	=	teken een driehoek
e	=	10 pixels rechtsboven	k	=	verander de kleur
f	=	10 pixels rechtsonder	t	=	verander de penhoek
spatiebalk	=	stoppen			max. 360 graden

Hier komt het programma:

```
100 REM Tekenen door toetsen in te drukken.
110 SCREEN 1
120 CLS:KEY OFF
130 KLEUR = 1
140 VIER$="U30 L30 D30 R30"
150 DRIE$="H30 G30 R60"
160 GOSUB 210
170 GOTO 160
180 CLS:SCREEN 0:WIDTH 80:KEY ON
190 END
200 REM toets
210 LOCATE 23,1:PRINT"Druk op toets! o-n-l-r-e-f-g-h-k-v-d-t "
220 I$=INKEY$:IF LEN(I$)=0 THEN 220
250 REM Invoer commando's
260 IF I$ = "o" THEN DRAW "U10"
270 IF I$ = "n" THEN DRAW "D10"
280 IF I$ = "l" THEN DRAW "L10"
290 IF I$ = "r" THEN DRAW "R10"
300 IF I$ = "e" THEN DRAW "E10"
310 IF I$ = "f" THEN DRAW "F10"
320 IF I$ = "g" THEN DRAW "G10"
330 IF I$ = "h" THEN DRAW "H10"
340 IF I$ = "v" THEN DRAW VIER$
350 IF I$ = "d" THEN DRAW DRIE$
360 IF I$ = "k" THEN KLEUR = KLEUR + 1
370 IF KLEUR > 3 THEN KLEUR = 1
380 IF I$ = "k" THEN DRAW "C=KLEUR;"
390 IF I$ = "t" THEN GOSUB 420
400 IF I$ = CHR$(32) THEN 180
410 RETURN
420 REM Hoek veranderen
430 LOCATE 23:PRINT"
440 LOCATE 23,1:INPUT "Hoeveel graden roteren? ";GR
450 DRAW "TA=GR;"
460 RETURN
```

Toelichting:

- 130 Als kleur van het palet wordt 1 gekozen. Dat is van belang om te weten, als we straks de mogelijkheid zien om de pen van kleur te laten veranderen.
- 140 Hier staat de reeks commando's waarmee een vierkant kan worden getekend met het DRAW statement.
- 150 Met deze reeks commando's kan een driehoek ontstaan op vergelijkbare wijze.
- 160 Het programma wordt naar de subroutine gestuurd.
- 170 Na elke toets indruk wordt het programma weer naar regel 160 gestuurd, tenzij op de spatiebalk is gedrukt.
- 210 Op de onderste regel staat een aanduiding op welke toetsen gedrukt kan worden. Deze toetsen hebben de hierboven aangegeven functie.
- 250 tot en met 400: In de meeste gevallen wordt per toets indruk een DRAW statement gegeven. Alleen bij de *k*, de *t* en de *spatiebalk* gebeurt er iets anders.
- 360 Als op de *k* gedrukt wordt, wordt de penkleur met één verhoogd.

- 370 Elk palet bevat maar vier penkleuren. Van de kleuren 0 tot en met 3 is kleur 0 de achtergrondkleur. De penkleur moet dan ook altijd 1, 2 of 3 zijn, willen we iets op het scherm kunnen zien. Zodra de waarde van KLEUR hoger is dan 3 zet regel 370 de waarde weer op 1.
- 390 Als de *t* wordt ingedrukt, springt het programma naar de subroutine van regel 420.
- 430 Eerst wordt de onderste regel leeggemaakt door er een lege string te plaatsen, om ervoor te zorgen dat er geen resten van de vorige commandoregel te zien zijn.
- 440 Op dezelfde plaats moet de gebruiker nu het aantal graden intikken.
- 450 Het DRAW statement wordt nu uitgevoerd.
-

Excel – De functies.

Opmaak en VBA zijn niet het enige wat we in Excel kunnen doen. Excel bevat tal van functies geordend in categorieën. Door naast de invoerbalk (ook wel genoemd: formulebalk) op de f_x te klikken, verschijnt er een venster met de mogelijkheden die u kunt kiezen.

Op elke cel kunt u vanuit dit venster een functie invoegen. U kunt er naar zoeken door een beschrijving in te voeren en op de **Zoeken** knop te klikken of een functie uit de lijst te kiezen. Elke functie die u aanklikt zal onder de lijst verschijnen met de parameters en een beschrijving van het doel van de functie. De categorie staat op **Laatst gebruikt**. Dit kan soms handig zijn wanneer sommige functies vaak gebruikt moeten worden. Deze functies kunt u dan snel in de Laatst gebruikte lijst vinden.

Op de volgende pagina ziet u de categorielijst uitgeklaapt.

Alle categorieën zijn op de afbeelding te zien, behalve de categorie **Alles**, maar dat spreekt voor zich. Laten we eens kijken wat elke categorie voor functies heeft.

Financieel

Deze categorie bevat functies die gebruikt kunnen worden voor administratie, zoals boekhouding. De functies nemen veel werk over, dus u hoeft veel van deze administratieve functies niet zelf te schrijven.

Datum en tijd

Deze functies zijn zeer handig voor het bepalen van datum en tijd, zoals tussendatum, het bepalen wanneer iemand vakantie heeft, enz. Daarom passen deze functies goed in planning- en afspraakdocumenten. Ook geschikt voor het onderhouden van een kasboek.

Wiskunde en trigonometrie

In deze categorie zijn functies te vinden geschikt voor het bepalen voor het onderhouden van grafieken, maar ook voor het afronden van breuken.

Statistisch

Dit zijn functies die niet echt een doel in een categorie hebben. Hier kunt u bijvoorbeeld functies vinden voor het bepalen van de gemiddelden of bepalen of het aantal gegeven cellen wel of niet leeg zijn. Leuk om te weten is bijvoorbeeld de functie GROOTSTE. Deze functie kijkt in een gegeven bereik welke waarde op k na het grootste getal is. We kunnen deze categorie dus een algemene functie-categorie noemen.

Zoeken en verwijzen

Deze functies zijn hele grote functies. Zij besparen veel tijd voor u voor het bepalen van bereikverwijzingen en het zoeken naar bereiken of verwijzingen. Ook kunt u hyperlinks instellen en het is zelfs mogelijk een verticaal cellenbereik om te zetten in een horizontaal cellenbereik.

Database

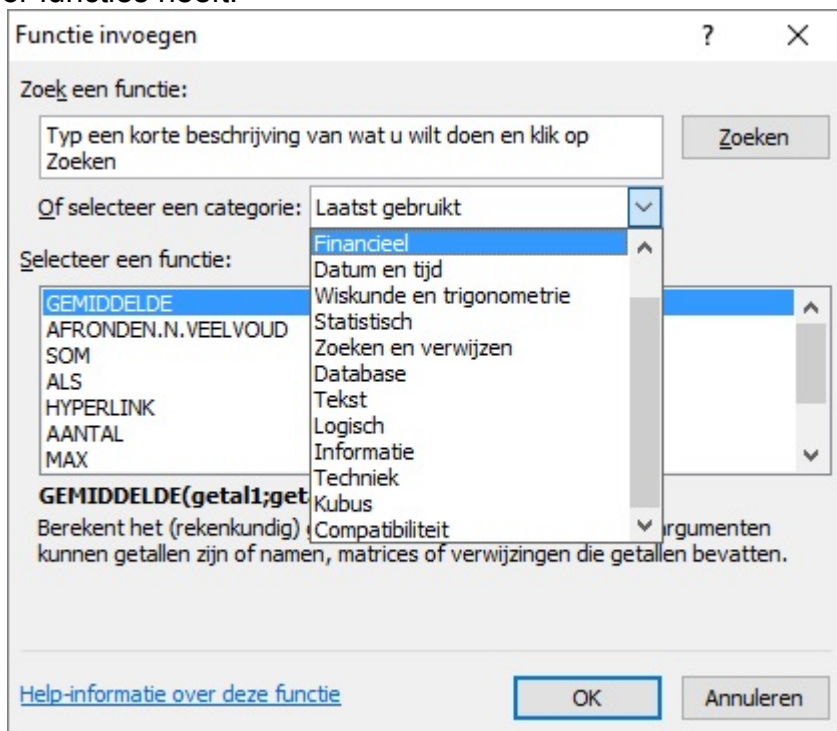
We slaan onze gegevens op en Excel bewaard ze in de werkbladen. Echter kan het een nadeel zijn als we een zoekfunctie met een criterium uitvoeren. Dit overschrijft dan het huidige werkblad. Door middel van een macro kunnen we het probleem omzeilen, bijvoorbeeld de gevonden gegevens tijdelijk op te slaan in een array of op een ander werkblad. Dit werkt echter niet zoals we willen. Deze categorie bevat database functies waarmee we een aparte database kunnen gebruiken en een criterium uit kunnen voeren, zonder er wat met de werkbladen gebeurt. Om hier wat meer over te weten te komen, ga ik in de volgende bulletins hier dieper op in. Na wat ik in de laatste zinnen zei over het apart in de database kunnen bewaren, is niet zeker. Daar moet ik ook zelf meer over weten.

Tekst

Deze categorie bevat functies die we in BASIC ook regelmatig tegenkomen, zoals het bepalen van een tekstdeel of het samenvoegen van tekstdelen.

Logisch

Deze categorie bevat logische functies. Hier vinden we de ALS functie, in BASIC genoemd als: IIF. Ook de bekende operatoren zijn hier te vinden.



Informatie

Deze categorie bevat informatieve functies, zoals het bepalen van de opmaak en de inhoud van een cel. Er zijn ook functies te vinden die alleen aan de opgegeven waarde een WAAR of ONWAAR resulteren.

Techniek

Hier vindt u technische functies voor het berekenen van machten, octale en hexadecimale waarden. De lijst van deze categorie is zeer lang. Daarom zullen de functies handig zijn voor technische- en bouwbedrijven.

Compatibiliteit

Deze categorie bevat functies van oudere Excel versies. Door deze functies te gebruiken, kunt u het document ook openen in een oudere Excel applicatie.

In de volgende bulletins zal ik wat functies laten zien met voorbeelden.

Cursussen

Liberty BASIC:

Cursus en naslagwerk, beide met voorbeelden op CD-ROM, € 6,00 voor leden. Niet leden € 10,00.

Qbasic:

Cursus, lesmateriaal en voorbeelden op CD-ROM, € 6,00 voor leden. Niet leden € 10,00.

QuickBasic:

Cursusboek en het lesvoorbeeld op diskette, € 11,00 voor leden. Niet leden € 13,50.

Visual Basic 6.0:

Cursus, lesmateriaal en voorbeelden op CD-ROM, € 6,00 voor leden. Niet leden € 10,00.

Basiccursus voor senioren, Windows 95/98,

Word 97 en internet voor senioren, (geen diskette). € 11,00 voor leden. Niet leden € 13,50.

Computercursus voor iedereen: tekstverwerking met Office en eventueel met VBA, Internet en programmeertalen, waaronder ook Basic, die u zou willen leren.

Elke dinsdag, woensdag en vrijdag in buurthuis Bronveld in Barneveld van 19:00 uur tot 21:00 uur op de dinsdag en van 9:00 uur tot 11:00 uur op de woensdag en vrijdag. Kosten € 5,00 per week.

Meer informatie? Kijk op '<http://www.i-t-s.nl/rdkcomputerservice/index.php>' of neem contact op met mij.

Computerworkshop voor iedereen; heeft u vragen over tekstverwerking of BASIC, dan kunt u elke 2^{de} en 4^{de} week per maand terecht in hetzelfde buurthuis Bronveld in Barneveld van 19:15 uur tot 21:15 uur. Kosten € 5,00.

Meer informatie? Kijk op '<http://www.buurthuisbronveld.nl>' of neem contact op met mij. Voor overige informatie: <http://www.tronicasoftware.nl>

	Software
Catalogusdiskette,	€ 1,40 voor leden. Niet leden € 2,50.
Overige diskettes,	€ 3,40 voor leden. Niet leden € 4,50.
CD-ROM's,	€ 9,50 voor leden. Niet leden € 12,50.

Hoe te bestellen

De cursussen, diskettes of CD-ROM kunnen worden besteld door het sturen van een e-mail naar penm@basic-gg.hcc.nl en storting van het verschuldigde bedrag op:

ABN-AMRO nummer 49.57.40.314

**HCC BASIC ig
Haarlem**

Onder vermelding van het gewenste artikel. Vermeld in elk geval in uw e-mail ook uw adres aangezien dit bij elektronisch bankieren niet wordt meegezonden. Houd rekening met een leveringstijd van ca. 2 weken.

Teksten en broncodes van de nieuwsbrieven zijn te downloaden vanaf onze website (<http://www.basic.hccnet.nl>). De diskettes worden bij tijd en wijlen aangevuld met bruikbare hulp- en voorbeeldprogramma's.

Op de catalogusdiskette staat een korte maar duidelijke beschrijving van elk programma.

Alle prijzen zijn inclusief verzendkosten voor Nederland en België.


Vraagbaken


De volgende personen zijn op de aangegeven tijden beschikbaar voor vragen over programmeerproblemen. Respecteer hun privé-leven en bel alstublieft alleen op de aangegeven tijden.

Waarover	Wie	Wanneer	Tijd	Telefoon	Email
Liberty BASIC	Gordon Rahman	ma. t/m zo.	19-23	(023) 5334881	grahman@planet.nl
MSX-Basic	Erwin Nicolai	vr. t/m zo.	18-22	(0516) 541680	basic@lordthanatos.com
PowerBasic CC	Fred Luchsinger	ma. t/m vr.	19-21		f.luchsinger@kader.hcc.nl
QBasic, QuickBasic	Jan v.d. Linden				j.vd.linden@kader.hcc.nl
Visual Basic voor Windows	Jeroen v. Hezik	ma. t/m zo.	19-21	(0346) 214131	j.a.van.hezik@kader.hcc.nl
Visual Basic .NET	Marco Kurvers	vr. t/m zo.	19-22	06 30896598	m.a.kurvers@live.nl
Basic algemeen, zoals VBA	Marco Kurvers	vr. t/m zo.	19-22	06 30896598	m.a.kurvers@live.nl
Office					
Web Design, met XHTML en CSS					

