# Canvas module versie 2 Liberty BASIC

**Documentatie**

Met deze module kun je wat eenvoudiger tekenen dan met de standaard stringcommando’s van Liberty BASIC.

Alle subroutines en functies zijn de methoden die met het woord Canvas beginnen.

Lees eerst de instructies in commentaar bovenaan de module.

De module is Engelstalig. Elke methode heeft een commentaarregel over wat de methode doet.

|  |  |
| --- | --- |
| **Methode** | **Omschrijving** |
| Init handle$, hWnd | Roep eerst deze methode aan om de Canvas module te initialiseren voordat je de methoden gaat gebruiken.  Geef de windowhandle als een string mee en de pointer naar de window voor de API methoden. |
| GetHandle$() | Deze methode geeft de opgeslagen windowhandle terug. De functie is private en het is de bedoeling dat alleen de methoden deze aanroepen. |
| GetHWnd() | Deze methode geeft het adres terug waar de pointer voor de windowhandle naar wijst. Deze functie is private en alleen de API methoden gebruiken deze pointer. |
| GetCoord Q$, byref nx, byref ny | Deze methode haalt de coördinaat uit de gegeven queue string. De queue werkt als een First In First Out. |
| PopCoord$(Q$) | Deze methode verwijdert de eerste coördinaat en geeft de rest van de queue string terug. |
| WindowPos x, y | Stel hiermee de positie in van het venster. Deze methode gebruikt de UpperLeftX en UpperLeftY variabelen. |
| WindowSize w, h | Stel hiermee de breedte en hoogte in van het venster. Deze methode gebruikt de WindowWidth en WindowHeight variabelen. |
| MoveParent x, y, w, h | Met deze API methode kun je je venster verplaatsen en van grootte wijzigen. |
| Cls | Roept het stringcommando “cls” aan. |
| SetPen size, color$ | Deze methode maakt een nieuwe pen met een meegegeven dikte en kleur. De pen wordt niet neergezet. |
| GetPen byref size, byref color$ | Deze methode geeft de dikte en kleur van de ingestelde pen terug. |
| SetRGB r, g, b | Bewaar met deze methode je eigen RGB door de R, G en B waarden mee te geven. |
| GetRGB$() | Deze methode geeft de opgeslagen RGB terug. |
| Down | Roept het stringcommando “down” aan om de pen neer te zetten. |
| Up | Roept het stringcommando “up” aan om de pen uit te zetten. |
| SetPenDown x, y | Deze methode zet de pen neer op de gegeven positie. |
| Rule ruleConst | Roept het stringcommando “rule” aan. Gebruik de welbekende rule constanten, zie meer daarover in de Help van Liberty BASIC en in mijn boek. |
| Center | Roept het stringcommando “home” aan om de pen in het midden te zetten. Houd er rekening mee dat het alleen geldt voor de clientview, niet voor de WindowWidth en WindowHeight. |
| Home | Deze methode verplaatst de pen in de linkerbovenhoek. |
| Goto x, y | Deze methode verplaatst de pen naar een andere positie. Er wordt geen lijn getekend. |
| North | Roept het stringcommando “north” aan om de penrichting naar het noorden te laten wijzen. |
| South | Deze methode roept eerst “north” aan en draait de pen naar 180 graden. |
| West | Deze methode roept eerst “north” aan en draait de pen naar 270 graden. |
| East | Deze methode roept eerst “north” aan en draait de pen naar 90 graden. |
| Turn direction | Deze methode draait de pen naar de gegeven directie. |
| Go distance | Verplaatst de pen of tekent een lijn met gegeven afstand. De methode roept het stringcommando “go” aan. |
| Turtle component$ | Deze methode kan een heel component van stringcommando’s in één keer uitvoeren. Dit werkt niet met de Canvas methoden. |
| Print size | Met het stringcommando “print” wordt er bepaald met welke papiergrootte er afgedrukt kan worden. De meegegeven grootte kan een Paper struct enumeratie zijn.   * Canvas.Paper.normal.struct * Canvas.Paper.vga.struct * Canvas.Paper.svga.struct * Canvas.Paper.xga.struct |
| Size size | Stelt de dikte van de pen in met het stringcommando “size”. |
| Place x, y | Plaatst de pen op gegeven positie. De methode roept het stringcommando “place” aan. |
| TextWidth(string$) | Deze methode berekent de lengte van de meegegeven tekst in pixels en geeft dat terug. De methode wordt beïnvloedt door de gegeven font. |
| SetText string$ | Roept het stringcommando “|” aan om tekst te renderen. Alleen het pipe symbool wordt gebruikt. |
| BackColor color$ | Deze methode stelt de meegegeven achtergrondkleur in. Deze methode is ook belangrijk voor de API methoden. Bovendien wordt de achtergrondkleur bewaard in de Pen struct. |
| BackColor$() | Deze methode geeft de achtergrondkleur terug die in de Pen struct is bewaard. Indien er geen achtergrondkleur is ingesteld, krijgt de Pen struct de kleur “black”. |
| Color color$ | Deze methode stelt de meegegeven penkleur in. De kleur wordt ook bewaard in de Pen struct. |
| Color$() | Deze methode geeft de penkleur terug die in de Pen struct is bewaard. Indien er geen penkleur is ingesteld, krijgt de Pen struct de kleur “black”. |
| SetPColor pColor  GetPColor() | Deze methoden werken als eigenschap om alleen de meegegeven kleurcode te bewaren en te retourneren in de RGB struct. Kan voor eigen code worden gebruikt. De Canvas API methoden gebruiken deze niet. |
| SetFillColor fillColor  GetFillColor() | Idem als de SetPColor en GetPColor. Gebruik hiermee je eigen RGB kleurcode voor andere code dan de Canvas API methoden die deze niet gebruiken. |
| Point x, y | Roept het stringcommando “set” aan om een pixel te tekenen. Gebruik de SetPen methode om een andere pen in te stellen. Een alternatief voor pixels tekenen is de methode SetPixel, zie aldaar. |
| LineTo x2, y2 | Deze methode tekent een lijn vanaf de huidige pen positie. Geef de eindpositie mee voor de lijn. |
| Line x1, y1, x2, y2 | Roept het stringcommando “line” aan om een lijn te tekenen. |
| Box x1, y1, x2, y2, filled | Roept één van de stringcommando’s “box” of “boxfilled” aan om een box te tekenen. Geef aan met een boolean of je de box gevuld wilt hebben of niet. |
| Rectangle x, y, w, h, filled | Deze methode doet hetzelfde als de Box methode, maar nu geef je de breedte en hoogte mee in plaats van de eindpositie. |
| Circle x, y, r, filled | Roept één van de stringcommando’s “circle” of “circlefilled” aan om een circle te tekenen. Geef aan met een boolean of je de circle gevuld wilt hebben of niet. |
| Ellipse x, y, w, h, filled | Deze methode doet hetzelfde als de Circle methode, maar tekent nu een Ellips. |
| HScrollBar visible, min, max, withMinMax  VScrollBar visible, min, max, withMinMax | Deze methoden creëren de schuifbalken in de clientview. Geef aan of de schuifbalken aan of uit moeten staan en of de meegegeven min en max waarden ingesteld moeten worden. Zie meer over de schuifbalken in de Help van Liberty BASIC. |
| GetPos byref x, byref y | Roept het stringcommando “posxy” aan. De methode geeft de x en y waarden terug als referentieparameters. Meer over zulke parameters kun je vinden in mijn boek. |
| GetDC() | Deze methode retourneert de Device Context zodat de API tekenmethoden kunnen tekenen op de juiste clientview. |
| GetPixel(hDC, x, y) | Deze methode geeft de kleur terug van een geplaatste pixel op de gegeven positie. De methode is een API methode en heeft de Device Context nodig, zie de methode GetDC. |
| SetPixel hDC, x, y, pColor | Deze methode plaatst een pixel met gegeven kleur op de gegeven positie. De methode is een API methode en heeft de Device Context nodig, zie de methode GetDC. |
| ReleaseDC hDC | Roep deze methode aan als je zeker bent geen API tekenmethoden aan te willen roepen. Na het aanroepen van deze methode, kun je niet meer tekenen met de API methoden totdat je weer GetDC aanroept. |
| Pie x, y, w, h, angle1, length, filled | Roept één van de stringcommando’s “pie” of “piefilled” aan om een pie te tekenen. Geef aan met een boolean of je de pie gevuld wilt hebben of niet. |
| PieAngle size, x, y, width, height, start | Deze methode tekent ook een pie, maar op een andere manier. De laatste parameter bepaald de juiste hoek zowel negatief als positief. Een instelling als start = 45 en ending = -45 geeft een leuk resultaat. Deze methode tekent zonder afrondingsproblemen. |
| Arc size, x, y, width, height, start, ending | Deze methode tekent een kromme vanaf een starthoek naar de eindhoek. Net als bij PieAngle, kun je hier ook een negatieve eindhoek geven. |
| FloodFill hDC, x, y, target, replacement | Deze methode vult een shape met de meegegeven replacement kleur met een recursieve aanroep. Het vullen is goed te volgen. Omdat hier een recursieve stack wordt gebruikt, zit er af en toe een kleine pauze in als de vulling ergens klaar is. Ook als de hele shape gevuld is, duurt het eventjes voordat de methode wordt verlaten.  Tijdens het recursief vullen kan het venster helaas niet gebruikt worden. Het is het beste om alleen kleine shapes hiermee te vullen. |
| FloodFillQueue hDC, x, y, target, replacement | Deze methode vult een shape via een Queue string. Het vullen gaat door totdat de Queue leeg is. Alle richtingen worden meteen gevuld, maar daardoor werkt deze methode veel trager. Wel kan tijdens het vullen het venster worden gebruikt, zoals het sluiten van het venster. Bepaal in welke shapes je deze wilt gebruiken. Zijn de vlakken te groot? Je kunt de Queue afbreken door het venster te sluiten. |
| FloodFillAPI hDC, x, y, fillcolor, filltype | Deze methode maakt gebruik van de ExtFloodFill API functie. Deze methode is het beste om te gebruiken. Stel met de BackColor methode de vulkleur in.  Geef één van de Windows constanten \_FLOODFILLBORDER of \_FLOODFILLSURFACE mee. Deze constanten bepalen hoe er gevuld moet worden.   * \_FLOODFILLBORDER Vult met de opgegeven achtergrondkleur zolang het niet de opgegeven kleur vindt in de aanroep van FloodFillAPI. * \_FLOODFILLSURFACE Vult met de opgegeven achtergrondkleur zolang het de opgegeven kleur vindt in de aanroep van FloodFillAPI. |
| PolyShape size, x, y, Q$, color$ | Doormiddel van een Queue string kan een polyshape getekend worden. De shape zal altijd gesloten zijn. |
| ContinueLine size, Q$, color$ | Deze methode tekent ook een polyshape, maar sluit niet automatisch de shape. |
| DrawBmp bmpName$, x, y | Roept het stringcommando “drawbmp” aan. |
| SetBmp bmpName$, x, y, w, h | Roept het stringcommando “getbmp” aan en bewaard de image in bmpName$. |

Voorbeeldprogramma.

call Canvas.WindowSize 1800, 1000

call Canvas.WindowPos 100, 100

open "Canvas demo" for graphics as #w

print "64"

#w "trapclose Quit"

call Canvas.Init "#w", hwnd(#w)

call Canvas.Down

call Canvas.SetPen 3, "black"

call Canvas.BackColor "blue" 'vulkleur

call Canvas.PieAngle 2, 400, 400, 60, 60, 45, -45

call Canvas.PolyShape 4, 400, 200, "250,600;550,600;400,200;", "black"

hDC = Canvas.GetDC()

'color = Canvas.GetPixel(hDC, 315, 400)

call Canvas.FloodFillAPI hDC, 399, 270, hexdec("&H000000"), \_FLOODFILLBORDER 'target

'call Canvas.Circle 399, 270, 10, False

'call Canvas.FloodFillAPI hDC, 399, 210, hexdec("&HFFFFFF"), \_FLOODFILLSURFACE 'replacement

'call Canvas.BackColor "red"

'call Canvas.FloodFillAPI hDC, 399, 410, 0, \_FLOODFILLBORDER

call Canvas.BackColor "red"

call Canvas.FloodFillAPI hDC, 399, 410, hexdec("&HFFFFFF"), \_FLOODFILLSURFACE

call Canvas.Color "yellow"

call Canvas.Arc 2, 400, 400, 45, -45, 50

#w "flush"

wait

sub Quit handle$

call Canvas.ReleaseDC hDC

#handle$ "discard"

close #handle$

end

end sub