



Constructeur de Robots
Pour les Loisirs Innovants et Pour l'Enseignement



Syntaxe POB-BASIC



Sommaire

1	INTRODUCTION	4
2	VARIABLES :	4
3	MOTS CLES	5
3.1	Global-Const-Static	5
3.2	Break-Continue	5
3.3	For-Next	6
3.4	If-Else-Endif	6
3.5	Repeat-Until	6
3.6	Select-EndSelect	6
3.7	While-Wend	7
3.8	Goto	7
3.9	End	8
4	STRUCTURE ET TABLEAU	9
5	PROCEDURES	10

Gestion du document

Nom de fichier	POB-BASIC Syntaxe
Date de création	21.03.06
Auteur	

Contacteur POB-Technology

Adresse	POB-TECHNOLOGY 4, rue Nicéphore Niepce 69 680 CHASSIEU, FRANCE
Adresse mail	contact@pob-technology.com
Téléphone	+33 (0)4 72 43 02 36
Fax	+33 (0)4 78 58 04 92

1 Introduction

L'ensemble de la syntaxe basic utilisé par l'outil POB-BASIC est issue de la syntaxe PureBasic, de même que la documentation : <http://www.purebasic.com/french/>

2 Syntaxes

Les commentaires sont signalés par '`;`'
; Un commentaire

Les fonctions doivent avoir un nom suivi de '('
uneFonction()

Les labels doivent être suivis par ':'
unLabel :

Les nombres hexadécimal doivent commencer par '0x'
a.l = 0x123ABC

3 Variables et types :

Le type des variables doit être spécifié à la première utilisation de la variable. Il existe 5 types de variables :

Nom	Extension	Mémoire	Plage
Byte	.b	1 octet	-128 à +127
Word	.w	2 octets	-32768 à +32767
Long	.l	4 octets	-2147483648 à +2147483647
Float	.f	4 octets	réel
String	.s	Longueur de la chaîne+1	Chaîne de caractère.

Exemples :

a.b ; Déclare une variable 'a' de type Byte

4 Mots clés

4.1 Global-Const-Static

Syntaxe

Global <variable>

Description

Global permet à des variables d'être utilisées globalement, pouvant ainsi être accessibles depuis l'intérieur de n'importe quelle procédure et depuis n'importe quel fichier basic.

Syntaxe

Const <variable>

Description

Const permet de déclarer des variables constantes.

*; Déclare une variable 'FASTGRAPHIC' constante et globale pour tous les fichiers basic du projet.
Global Const FASTGRAPHIC.I = 1*

Syntaxe

Static <variable>

Description

Static permet de créer des variables locales à un fichier basic. Ces variables sont utilisables dans toutes les procédures du fichier basic.

; Exemple : on est dans le fichier « test.bas » :

*; Déclare une variable 'FORMFOUND' accessible dans toutes les procédures du fichier « test.bas ».
Static FORMFOUND.I*

4.2 Break-Continue

Syntaxe

Break

Description

Break permet de quitter à n'importe quel moment une ou plusieurs des boucles suivantes: Repeat, For, While.

Syntaxe

Continue

Description

Continue permet de passer directement à la prochaine itération dans l'une des boucles suivantes: Repeat, For et While.

4.3 For-Next

Syntaxe

```
For <variable> = <expression1> To <expression2> [Step <constante>]  
    ...  
Next
```

Description

La fonction **For** : **Next** est utilisée pour produire une boucle dans le programme, avec les paramètres définis. A chaque cycle, <variable> est incrémentée de 1 (ou d'une valeur correspondant au pas indiqué dans **Step** <constante>).

La première valeur de <variable> est <expression1>. La boucle est interrompue dès que la valeur de <variable> atteint la valeur de <expression2>.

4.4 If-Else-Endif

Syntaxe

```
If <expression>  
    ...  
[Elseif <expression>]  
    ...  
[Else]  
    ...  
Endif
```

Description

La structure **If** est utilisée pour effectuer des tests et/ou changer le déroulement du programme selon le résultat (vrai ou faux) du test.

Elseif est utilisé pour produire un nombre quelconque de tests additionnels si le premier n'a pas eu un résultat vrai.

La commande optionnelle **Else** est utilisée pour exécuter une séquence de code si tous les tests précédents de la structure ont échoué. Les structures **If** peuvent être imbriquées sans limite de profondeur.

4.5 Repeat-Until

Syntaxe

```
Repeat  
    ...  
Until <expression>
```

Description

Cette fonction boucle jusqu'à ce que <expression> soit vrai.

4.6 Select-EndSelect

Syntaxe

```
Select <expression1>
    Case <expression2>
        ...
    [Case <expression3>]
        ...
    [Default]
        ...
EndSelect
```

Description

Select permet d'opérer des choix rapides. Le programme exécute <expression1> et retient la valeur en mémoire. Cette valeur est ensuite comparée à chacune des valeurs "Case <expression>" et s'il y a égalité, le code du bloc **Case** est exécuté pour quitter ensuite la structure **Select**. Si aucune des valeurs **Case** n'est vraie, alors le code du bloc **Default** code, (s'il est spécifié) est exécuté.

Bob.I = 2

```
Select Bob
    Case 1
        ...
    Case 2
        ...
    Default :
        ...
EndSelect
```

4.7 While-Wend

Syntaxe

```
While <expression>
    ...
Wend
```

Description

While produit une boucle jusqu'à ce que l'expression devienne fausse. Il est à noter qu'avec le test accompagnant **While**, si la première tentative échoue, le programme n'entrera pas dans la boucle et évitera donc cette section de code contrairement à une boucle Repeat qui est toujours exécutée au moins une fois (car le test est effectué en sortie de code conditionnel).

4.8 Goto

Syntaxe

```
Goto <label>
```

Description

Cette commande permet de transférer directement l'exécution du programme à l'emplacement d'un label. Soyez attentif en utilisant **Goto** car une mauvaise utilisation peut provoquer une fin anormale du programme...

4.9 End

Syntaxe **End**

Description

Termine et quitte la procédure courante de manière correcte à n'importe quel endroit du code source de la procédure.

5

6 Structure et Tableau

Syntaxe

Dim nom.<type>(<expression>, [<expression>], ...)

Description

Dim est utilisé pour créer un nouveau tableau. Un tableau peut être composé d'éléments de type quelconque connu sous PureBasic, incluant les structures et les types définis par l'utilisateur.

Attention : les éléments d'un tableau sont tous de même type.

Dim UnTableauEntier.l(9) ; Crée un tableau d'entier de 10 cases

UnTableauEntier(0) = 123 ; Dans la case 0, mettre l'entier '123'

UnTableauEntier(2) = 12345 ; Dans la case 2, mettre l'entier '12345'

Syntaxe

Structure <nom>

...

EndStructure

Description

Structure est utile pour définir un type utilisateur et accéder à des types complexes par exemple. Cela peut être plus efficace dans la mesure où vous pouvez regrouper dans un même objet des informations communes. On accède aux structures avec le caractère \.

; Déclare une structure 'TestStruct' de deux membres : 'UnEntier' et 'UnReel'

Structure TestStruct

UnEntier.l

UnReel.f

EndStructure

Test. TestStruct ; Créer une structure 'Test'

Test\UnEntier = 12345; Placé dans le membre 'UnEntier' de 'Test' l'entier 12345

Test\UnReel = 1.23456 ; Placé dans le membre 'UnReel' de 'Test' le réel 1.23456

7 Procédures

Syntaxe

```
Procedure[.<type>] nom(<variable1>[,<variable2>,...])  
    ...  
    [ProcedureReturn valeur]  
EndProcedure
```

Description

Une procédure est une partie du code indépendante du programme principal qui peut avoir des paramètres et des variables propres. En PureBasic, les procédures sont récursives et peuvent donc s'appeler elles-mêmes. Lors de chaque appel à la procédure, les variables locales sont automatiquement initialisées avec la valeur 0.

Une procédure peut avoir une valeur de retour si nécessaire. Il faut en définir le type après **Procedure** et utiliser le mot clef **ProcedureReturn** à tout moment opportun dans la procédure.

Remarque

Tout programme basic pour le POBEYE doit contenir une seule procédure de nom « **ProcedureMain** ». Cette procédure sera appelée au moment du lancement du programme. A noter que cette procédure peut ensuite appeler des procédures externes.

; La procédure principale du programme basic pour le POBEYE :

```
Procedure ProcedureMain()  
    ...  
    UneAutreProcedure() ; la procédure 'UneAutreProcedure' est appelée  
    ...  
EndProcedure
```

Syntaxe

```
Declare nom
```

Description

Une procédure peut appeler une autre procédure qui n'a pas été déclarée avant sa propre définition ou se situant dans un autre fichier. Ce cas peut se produire et provoquer une erreur de compilation 'Procedure not found'. **Declare** permet de traiter ce cas particulier en déclarant seulement l'en-tête de la procédure.

; Fichier Main.bas :

; Déclare la procédure UneAutreProcedure comme étant défini ailleurs :
Declare UneAutreProcedure

; La procédure principale du programme basic pour le POBEYE :

```
Procedure ProcedureMain()  
    ...  
    UneAutreProcedure() ; la procédure 'UneAutreProcedure' est appelée  
    ...  
EndProcedure
```

; Fichier Function.bas

; La procédure 'UneAutreProcedure' est défini dans un autre fichier appelé 'Function.bas'



Constructeur de Robots
Pour les Loisirs Innovants et Pour l'Enseignement
Documentation Syntaxe Pob-Basic

Procedure UneAutreProcedure()

...

EndProcedure



Constructeur de Robots
Pour les Loisirs Innovants et Pour l'Enseignement
Documentation Syntaxe Pob-Basic

Contacter POB-Technology

POB-TECHNOLOGY
4, rue Nicéphore Niepce
69 680 CHASSIEU
FRANCE

Adresse web : www.pob-technology.com

Adresse mail : contact@pob-technology.com

Téléphone : +33 (0)4 72 43 02 36
Fax : +33 (0)4 78 58 04 92