**NetBasic** © Jean-Marc DURO

NetBasic est un interpréteur de scripts au format Basic agrémenté de fonctions de contrôle de navigateurs (Google Chrome/Chromium à ce jour).

Le script ci-dessous est un exemple de programme NetBasic.

**gosub** sub

**for** i **=** 1 **to** 10 **do**

**print**

**if** i **==** 2 **then**

**print** i**^**4

**elsif** i **==** 4 **then**

**print** i**%**2

**elsif** i **>=** 6 **then**

**print** i**\***2

**else**

**print** i

**end if**

**end for**

**while** i **>** 1 **do**

**print** i

i **-=** 1

i **=** i **-** 1

**end while**

**print** **"end"**

**end**

**label** sub

**print** "subroutine"

**print** "subroutine" **&** " " **&** "sub"

**print** **"sub "** **\*** 3

**return**

**Instructions**

Le programme comprend un ensemble de mots-clés génériques listés ci-dessous.

**print** [ <parameters> ]

print affiche la liste d'expressions (optionnelle) transmises en paramètres et termine par un retour à la ligne.

Ces expressions peuvent être des chaînes de caractères, des nombres ou revêtir une forme plus complexe avec des opérateurs et des parenthèses (voir plus loin dans ce document).

Le formatage des nombres dépend de leur type, il n'est pas réglable.

**for** <identifier> = <from> **to** <upto> [ **by** <by> ] **do**

**end for**

Une boucle for classique avec une variable, un indice de départ, un indice de fin et en option un pas d'incrémentation.

**while** <condition> **do**

**end while**

Une boucle conditionnelle. La condition est une expression qui peut être complexe.

**if** <condition> **then**

**elsif** <condition> **then**

**else**

**end if**

Un test conditionnel. Les branches elsif et else sont optionnelles. Il peut y avoir plusieurs branches elsif.

**label** <etiquette>

**return**

**gosub** <label>

Les routines sont déclarées après l'instruction end, en fin de programme. Elles sont délimitées par une étiquette (label) et terminée par return. Elles sont appelées par l'instruction gosub.

Example:

print "main"

gosub sub

print "return to main"

end

label sub

print "subroutine"

return

**pause** <delay>

Temps d'attente en secondes

**end**

Fin du programme

**Commentaires**

Les commentaires simples sont préfixés par 2 tirets "--". Ce qui suit jusqu'à la fin de la ligne est ignoré.

Example:

print i + 8 -- addition

Les commentaires multilignes sont encadrés par "/\*" et "\*/".

Example:

/\*

Ceci est un commentaire muiltilignes.

Il sera ignoré.

\*/

**Fonctions Web**

**connect** <ip> <port> *(procédure)*

Se connecter au pilote du navigateur

Example pour Chrome/Chromium en local:

connect "127.0.0.1" 9515

**implicit\_wait** [<delay>] *(procédure)*

Spécifie le temps qu'attendra le driver pour trouver un élément avant de renoncer.

Example:

implicit\_wait 30

**navigate\_to** <url> *(procédure)*

Naviguer à la page spécifiée

Example:

navigate\_to "https://10.193.35.196/users/sign\_in?locale=fr"

**search** <element> [ **by** <selector>] *(procédure)*

Rechercher un élément de la page web avec un sélecteur de type optionnel. Si <selector> est omis, la recherche se fera sur l’attribut "id".

<selector> peut prendre les valeurs suivantes:

* "class name" Retourne un élément dont le nom de classe contient la valeur indiquée; les noms de classes composites ne sont pas permis.
* "css selector" Retourne un élément dont le sélecteur CSS correspond à la valeur indiquée.
* "id" Retourne un élément dont l'attribut ID correspond à la valeur indiquée.
* "name" Retourne un élément dont l'attribut NAME correspond à la valeur indiquée.
* "link text" Retourne un lien dont le texte visible correspond à la valeur indiquée.
* "partial link text" Retourne un lien dont une partie du texte visible correspond à la valeur indiquée.
* "tag name" Retourne un élément dont le tag correspond à la valeur indiquée.
* "xpath" Retourne un élément correspondant à une expression XPath.

Examples:

search "user\_uid"

search "user[password]" by "name"

**click** [<element>] *(procédure)*

Click sur un élément. Si <element> est renseigné, l'élément courant sera celui recherché (par l'attribut "id"), sinon le click sera effectif sur le dernier élément sélectionné.

Pour cliquer sur un élément identifié par un autre attribut que "id", il faut effectuer une recherche par attribut auparavant (search … by).

Examples:

click "user\_uid"

search "user[password]" by "name"

click

**clear** [<element>] *(procédure)*

Efface le contenu d'un élément. Si <element> est renseigné, l'élément courant sera celui recherché (par l'attribut "id"), sinon le contenu effacé sera celui du dernier élément sélectionné.

Pour effacer le contenu d'un élément identifié par un autre attribut que "id", il faut effectuer une recherche par attribut auparavant (search … by).

Examples:

clear "user\_uid"

search "user[password]" by "name"

clear

**get** [<element>] *(fonction)*

Récupère le contenu d'un élément. Si <element> est renseigné, l'élément courant sera celui recherché (par l'attribut "id"), sinon le contenu du dernier élément sélectionné sera récupéré.

Pour récupérer le contenu d'un élément identifié par un autre attribut que "id", il faut effectuer une recherche par attribut auparavant (search … by).

Examples:

login = get "user\_uid"

search "user[password]" by "name"

password = get

**set** <value> [**of** <element>] *(procédure)*

Affecte un contenu à un élément. Si <element> est renseigné, l'élément courant sera celui recherché (par l'attribut "id"), sinon le contenu du dernier élément sélectionné sera affecté.

Pour affecter un contenu à un élément identifié par un autre attribut que "id", il faut effectuer une recherche par attribut auparavant (search … by).

Examples:

set "user\_uid" "my\_login"

search "user[password]" by "name"

set "my\_password"

La liste des fonctions reste à compléter. Il faut savoir que pour:

* sélectionner une valeur dans une liste déroulante, il suffit généralement d'affecter la valeur voulue à ce champ, sans tenir compte de la liste  
    
  Example:

set "language" "French"

* cliquer sur une case à cocher, il faut créer une instruction spécifique à chaque situation. Il y aura une version de NetBasic adaptée à Ruby on Rails, une autre adaptée à Django, etc. Chaque framework à sa façon de faire.

**opérateurs de calcul**

+ addition  
ex: print 20 + 10 *(30)*

* soustraction   
  ex: print 20 - 10 *(10)*

\* multiplication *(nombre et chaines)*   
ex:  
print 20 \* 10 *(200)*  
print 3 \* "abc" *("abcabcabc")*

/ division   
ex: print 25 / 10 *(2.5)*

^ Elévation à la puissance   
ex: print 2 ^ 3 *(8)*

% (modulus) restant d'une division entière   
ex: print 25 % 10 (5)

& ET binaire (AND) si les opérandes sont des entiers,  
concaténation si les opérandes sont des chaînes   
ex:  
print 255 & 7 *(7)*  
print "abc" & "def" *("abcdef")*

| OU binaire (OR)   
ex: print 8 | 2 *(10)*

~ NON binaire (NOT)   
ex: print ~ 8 *(-1)*

# OU exclusif binaire (XOR)   
ex: print 15 # 6 *(9)*

== égalité   
ex: if (i == 10) then

!= différence   
ex: if (i != 10) then

< inférieur   
ex: if (i < 10) then

> supérieur   
ex: if (i > 10) then

<= inférieur ou égal

ex: if (i <= 10) then

>= supérieur ou égal   
ex: if (i >= 10) then

&& ET logique   
ex: if (i > 10) && (i < 20) then

|| OU logique

ex: if (i < 10) || (i > 20) then

! NON logique   
ex: if !(i < 10) then

div division entière   
ex: print 25 div 10 *(2)*

mod (modulus) restant d'une division entière   
ex: print 25 mod 10 *(5)*

and ET logique   
ex: if (i > 10) and (i < 20) then

or OU logique   
ex: if (i < 10) or (i > 20) then

not NON logique   
ex: if not (i < 10) then

xor OU exclusif logique   
ex: if (a xor b) then

**opérateurs d'affectation**

= affectation   
ex: i = 10

-= autosoustraction   
ex: i -= 10 *(équivalent à i = i - 10)*

+= autoaddition   
ex: i += 10 *(équivalent à i = i + 10)*

\*= automultiplication   
ex: i \*= 10 *(équivalent à i = i \* 10)*

/= autodivision   
ex: i /= 10 *(équivalent à i = i / 10)*

&= autoconcaténation   
ex: s &= "suffixe" *(équivalent à s = s & "suffixe")*

**Compléments**

Il y a des dizaines de fonctions utilisables dans la librairie de pilotage du navigateur (ex: take\_screenshot, mouve\_mouse), mais toutes ne seront pas implémentées car rarement utilisées (ex: fonctions spécifiques aux tablettes comme touch\_pinch).

Il y aura un jour une version de NetBasic pour Firefox, mais la librairie correspondante n'est pas encore écrite.

Il peut, si besoin, y avoir des versions Linux ou Mac OS X (plateforme matérielle à fournir pour Mac, je n'en ai pas).