

CasioPython

Python sur calculatrices Casio monochromes

Manuel d'utilisation (v1.4)

Table des matières

I) Introduction.....	2
II) Installation de CasioPython.....	3
A) Calculatrices supportées.....	3
B) Rendre compatible la Graph 35+(E).....	3
C) Transfert de CasioPython sur la calculatrice.....	3
III) Ecriture et exécution d'un script python.....	4
A) Ecriture d'un script python.....	4
B) Exécution dans le « shell ».....	7
C) Erreurs et bugs.....	8
1) Erreurs liées à Python.....	8
2) Test d'égalité de flottants.....	8
3) Erreurs système.....	9
4) Boucles infinies.....	9
5) Récursion.....	9

I) Introduction

Les nouveaux programmes d'algorithmie préconisent l'usage d'un langage avec, entre autres, gestion des types de variables (entier/réel/ complexe/chaîne de caractères...) et des fonctions à paramètres. Toutefois, les langages natifs aux calculatrices TI et Casio ne satisfont pas ces besoins.

Le Ministère de l'Education Nationale préconise l'utilisation du langage « python » pour l'algorithmie. Il est en effet adapté à un public débutant du fait de sa syntaxe simple et concise.

Cependant, la plupart des lycées ont un accès restreint, voire inexistant, aux salles informatiques ; ainsi, exécuter du code Python sur ordinateur n'est pas une solution optimale. Il est préférable de l'exécuter sur la calculatrice, que chaque élève doit avoir.

Les calculatrices Numworks, ainsi que la Casio 90+E, possèdent la possibilité d'y exécuter du code python. Toutefois, Casio n'a pas inclus d'interpréteur python dans les calculatrices « bas de gamme » monochromes.

Ce programme vise à inclure un interpréteur python pour certaines calculatrices monochromes (voir partie II pour une liste de calculatrices sur lesquelles CasioPython fonctionne).

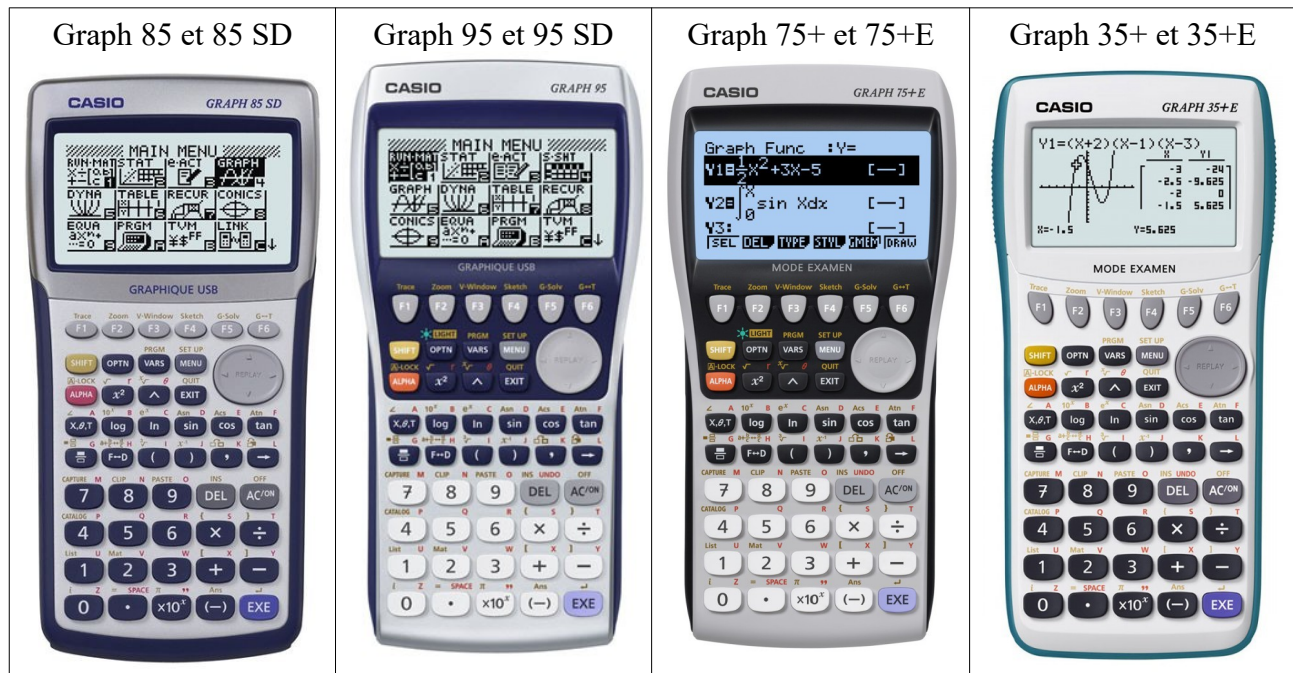
CasioPython a été créé par Zezombye, avec l'aide de Lephenixnoir et jickster. Il incorpore l'interpréteur python MicroPython (créateur : Damien George), ainsi que l'éditeur de texte Edit par Diego Marin.

ATTENTION ! CasioPython n'étant pas un programme officiel, il n'est pas disponible en mode examen. Si vous voulez avoir accès à l'interpréteur python lors d'un examen (pour vérifier les algorithmes, par exemple) il vous faudra les calculatrices Numworks ou Graph 90+E.

II) Installation de CasioPython

A) Calculatrices supportées

Les modèles de calculatrices sur lesquelles CasioPython peut être installé sont :



Concernant les Graph 35, vérifiez qu'elle ressemble à l'image : les Graph 35 « vertes » ne sont pas supportées.

Pour installer CasioPython sur les Graph 35+ et 35+E, il faut réaliser une manipulation (partie B).

B) Rendre compatible la Graph 35+(E)

Il est possible de rendre compatible les Graph 35+ et 35+E : il faut les « transformer » en Graph 75, en mettant le système d'exploitation des Graph 75 sur lequel il est possible d'installer CasioPython.

Cette manipulation nécessite toutefois des connaissances techniques. Il faut également faire attention à ne pas débrancher le câble d'alimentation durant le transfert du système d'exploitation.

Pour performer la manipulation, référez-vous à ce tutoriel :

<https://planet-casio.com/Fr/forums/topic13930-46-a.html>

C) Transfert de CasioPython sur la calculatrice

Téléchargez le programme (format .g1a) ici :

<https://planet-casio.com/Fr/programmes/programme3603-1-C.html>

Une fois le g1a obtenu, suivez le tutoriel de transfert :

https://planet-casio.com/Fr/logiciels/voir_un_logiciel_casio.php?showid=16

N'hésitez pas à poster un message dans les tutoriels si vous avez une question ou un problème.

III) Ecriture et exécution d'un script python

Nous prendrons ici l'exemple du script du « plus ou moins » (aussi appelé « nombre magique ») :

```
import random

n=random.randint(1,100)
i=0

while i!=n:
    i=int(input("entrez un nombre : "))

    if i < n :
        print("trop petit")
    elif i > n :
        print("trop grand")

print("vous gagnez")
```

A) Ecriture d'un script python

Dans le menu de la calculatrice, cliquez sur l'icône en bas du menu :



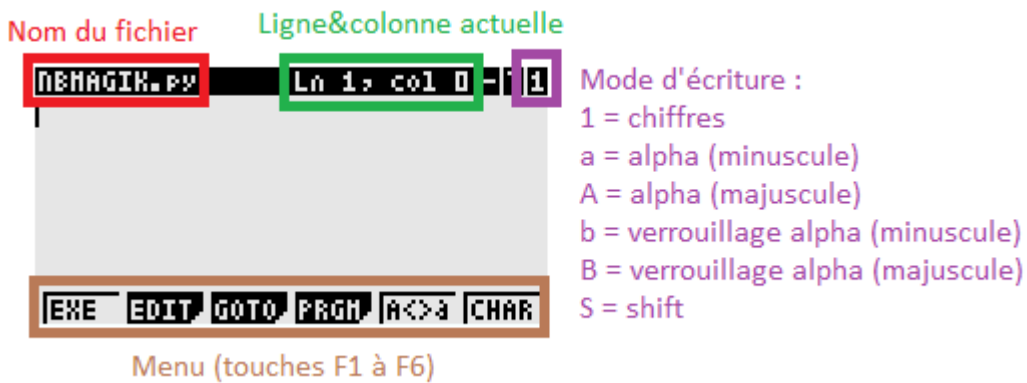
Vous arriverez à l'écran suivant ; appuyez sur la touche F3 (NEW).



Entrez ensuite le nom du fichier ; ici « NBMAGIK ». Les noms sont limités à 8 caractères (sans compter le .py), lettres et chiffres uniquement. Appuyez ensuite sur F1 (OK) pour valider.



Vous êtes maintenant dans l'éditeur de scripts :



Afin d'augmenter la vitesse de frappe, le clavier a été modifié pour inclure des caractères et des raccourcis.

Voici donc ci-dessous le clavier utilisé par CasioPython :



- 1: insère 2 espaces pour une indentation
- 2: supprime le caractère après le curseur
- 3: `math.sqrt()`
- 4: `**(1/)`
(calcule la racine n-ième)
- 5: `[0 for i in range()]`
(expression pour initialiser une liste)
- 6: `[[0 for i in range()] for j in range()]`
(expression pour initialiser une matrice)
- 7: `random.randint()`
- 8: page haut
- 9: début de ligne
- 10: fin de ligne
- 11: page bas

Note : les fonctions mathématiques (log, sin, cos, tan...) sont préfixées de "math."

À l'aide des raccourcis clavier et du menu accessible avec F1-F6, recopions ce programme.
Des touches reliées avec « + » signifient un appui successif (par exemple ALPHA+8 donne « n »).

1. Appuyez sur [F4] (PRGM), [F4] (I/O) et [F6] (rand), puis [EXE].
2. Entrez « n= » puis [SHIFT]+[3]. Entrez ensuite « 1,100 », puis déplacez le curseur au bout de la ligne et appuyez sur [EXE].
3. Entrez « i=0 » puis [EXE].
4. Appuyez sur [EXIT] pour descendre dans le menu, puis [F2] (loop).
5. Appuyez sur [F4] (while), entrez « i », appuyez sur [SHIFT]+[a b/c], puis entrez « n ». Appuyez ensuite sur la flèche du bas et veillez à ce que le niveau d'indentation soit respecté (le curseur doit être entre le « h » et le « i ») :

```

NEHAGIK.py Ln 5, col 2 | T1
import random
n=random.randint(1,100)
i=0
while i!=n:
|
|EXE|EDIT|GOTO|PRGM|A<>A|CHAR

```

6. Entrez « i=int », puis [ALPHA]+[x²]. Appuyez sur [EXIT], [F4] (I/O) puis [F2] (inpt). Entrez la chaîne de caractère « "entrez un nombre : " ». Placez le curseur à la fin de la ligne puis appuyez sur [EXE].
7. Appuyez sur [EXIT], [F1] (stmt), [F1] (if). Entrez « i », [a b/c] (<), « n », puis appuyez sur la flèche du bas. Veillez à ce que le niveau d'indentation soit respecté (4 caractères).

```

NEHAGIK.py Ln 7, col 4 | T1
n=random.randint(1,100)
i=0
while i!=n:
    i=int(input("entrez un nombre")
    if i<n:
|
|if|else|if/else|elif|def|return

```

8. Appuyez sur [EXIT], [F4] (I/O) puis [F1] (prnt). Entrez la chaîne de caractères « "trop petit" ». Placez le curseur à la fin de la ligne, appuyez sur [EXE], puis appuyez une fois sur [DEL] pour diminuer l'indentation.
9. Appuyez sur [EXIT], [F1] (stmt), [F4] (elif). Entrez « i », [F↔D] (>), « n », puis appuyez sur la flèche du bas. Veillez à ce que le niveau d'indentation soit respecté (4 caractères).
10. Appuyez sur [EXIT], [F4] (I/O) puis [F1] (prnt). Entrez la chaîne de caractères « "trop grand" ». Note : vous pouvez utiliser le copier/coller et copier la ligne « trop petit ».
11. Placez le curseur à la fin de la ligne, appuyez sur [EXE], puis appuyez deux fois sur [DEL]. Vous devriez être revenu au niveau d'indentation de base (0).
12. Appuyez sur [F1] (prnt) puis entrez la chaîne de caractères « "vous gagnez" ».

Le programme final devrait ressembler à cela :

```
nbmagik.py  Ln 9, col 19 | T | B
i=int(input("entrez un nombre")
if i<n:
    print("trop petit")
elif i>n:
    print("trop grand")
print("vous gagnez")
print | input | iert | f/i x | nath | rand
```

Appuyez enfin sur [EXIT] deux fois pour revenir au niveau de menu initial.

B) Exécution dans le « shell »

Une fois votre script fini, appuyez sur F1 **EXE** pour exécuter le script.

Vous vous trouverez alors dans le « shell » :

```
MicroPy 1.9.4 | Port by Zezombye
>>> from nbmagik import *
entrez un nombre : |
```

Dans ce cas particulier, vous êtes dans une fonction « input », mais pour d'autres scripts vous serez dans le « prompt » avec « >>> » :

```
entrez un nombre : 45
trop petit
entrez un nombre : 75
trop petit
entrez un nombre : 90
trop petit
entrez un nombre : 95
vous gagnez
>>> |
```

Dans le shell, vous pouvez :

- déplacer le curseur avec les flèches gauche et droite ;
- scroller avec les flèches haut et bas ;
- parcourir l'historique avec shift+haut et shift+bas.

Pour quitter le shell, appuyez sur [EXIT], vous reviendrez alors à l'éditeur.

C) Erreurs et bugs

1) Erreurs liées à Python

Les erreurs de Python (appelées « exceptions ») sont liées au script. Ce sont généralement des erreurs de syntaxe, de conversion de type, etc.

Une erreur python est composée de :

- la « stack trace » qui est un historique des fonctions ;
- la ligne et la colonne de l'erreur dans votre script.

Voici un exemple d'erreur syntaxe :

```
MicroPy 1.9.4 | Port by Zezombye
>>> from a import *
Trace (most recent call last):
File "<stdin>", line 1, col 0, in <module>
File "a.py", line 2, col 5
SyntaxError: invalid syntax
>>> |
```

On peut voir que c'est une « SyntaxError », soit une erreur syntaxe.

La « trace » nous permet de savoir d'où vient l'erreur :

- d'abord de <stdin>, qui est le shell (avec « from a import * »)
- ensuite de « a.py », ligne 2, colonne 5.

Revenons dans l'éditeur, et positionnons le curseur à la ligne 2 et colonne 5 :

```
a.py Ln 2, col 5 |S|T|b
a=1
if a=1:
    print("test")

[EXE] [EDIT] [GOTO] [PRGM] [A<>A] [CHAR]
```

Hé oui : il ne faut pas confondre le « = » (assignement) et le « == » (comparaison) !

2) Test d'égalité de flottants

Ce « bug » s'applique au langage Python (et donc également sur Graph 90+E, Numworks, PC...) ; ce n'est pas un bug venant de CasioPython.

En raison de la manière dont sont stockés les flottants en langage Python, les nombres flottants (décimaux) ne sont pas représentés de manière exacte, ce qui peut provoquer des erreurs lors d'un test d'égalité (ou d'inégalité).

Ainsi, on peut voir ci-contre que bien que $0.3+0.2==0.5$ donne Vrai, $0.5-0.3-0.2==0$ donne Faux ! En effet, les erreurs de précision s'accumulent, et pour Python, $0.5-0.3-0.2$ donne $-5.55*10^{-17}$.

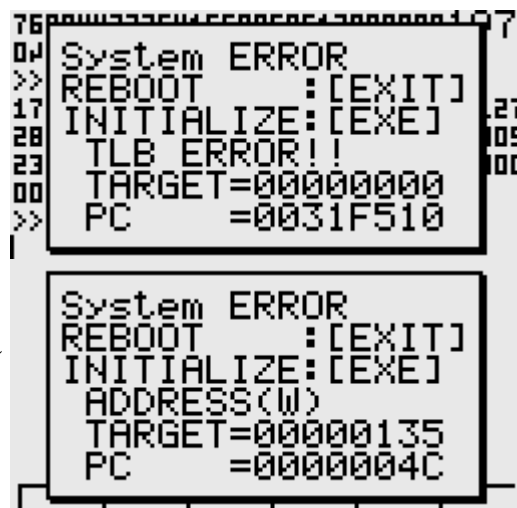
```
>>> 0.3+0.2==0.5
True
>>> 0.5-0.3-0.2==0
False
>>> 0.5-0.3-0.2
-5.551115123125783e-17
```


3) Erreurs système

CasioPython n'étant pas fini, il peut arriver qu'il y ait des erreurs système (voir ci-contre).

Si une erreur système s'affiche :

- Notez tout d'abord les informations : le type d'erreur (« TLB ERROR » ou « ADDRESS(W) »), ainsi que la valeur de TARGET et PC.
- Appuyez ensuite sur la touche [EXIT] (et non pas sur la touche [EXE], sinon vous effacerez la mémoire principale !)
- Faites un commentaire sur la page du programme en détaillant l'erreur système, ainsi que ce qui l'a causée (par exemple, l'exécution d'un script précis) ; si l'erreur est causée par un script, insérez le dans votre commentaire.



4) Boucles infinies

Lorsqu'un script est dans une boucle infinie, il n'est pour l'instant pas possible de l'arrêter ; la seule solution est d'appuyer à l'aide d'une pointe de stylo sur le bouton Reset au dos de la calculatrice.

5) Récursion

Pour une raison encore inconnue, mettre trop de niveaux de récursion (environ 50) provoque une erreur système. Lors de l'utilisation de fonctions récursives (factorielle, fibonacci...) il est conseillé de ne pas dépasser 50 appels.