

TI 83 CE Edition Python

jpr, 11/10/2020

Complément pour les **Mémentos Lycée**
(http://jean.paul.roy.free.fr/PAA/python_lycee.html)



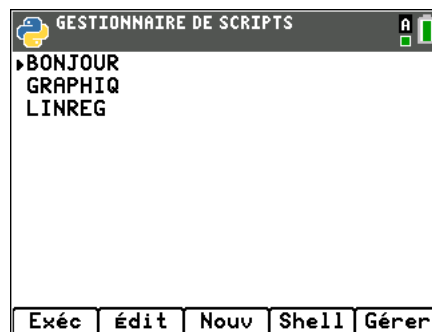
L'application Python est déjà installée.

PRISE EN MAIN DE PYTHON

On allume la calculatrice (*on*), on passe en mode programme (*prgm* deux fois). On a alors le choix entre programmer en BASIC (1) ou en Python (2). On opte clairement pour 2.

Le langage Python de la TI-83 se nomme **AppVar Python**. Ce n'est pas un Python entièrement standard (la norme est sur www.python.org) mais un dérivé de CircuitPython, un dialecte Python utilisé pour programmer les microcontrôleurs (micro-ordinateurs sur une puce). Les fichiers Python usuels *xxx.py* rédigés sur ordinateur peuvent être traduits en fichier AppVar Python *xxx.8xv* et inversement, avec le logiciel **TI CONNECT** installé sur un ordinateur et connecté à la TI 83 par un câble USB. Il faut bien entendu que le fichier *xxx.py* soit compatible avec AppVar Python (pas de module *numpy* par exemple)...

Bref, on arrive alors dans le [gestionnaire de scripts](#) (script == programme) :



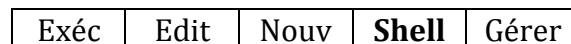
Avec le programme TI-CONNECT, j'ai fait une mise à jour du TI83CEBundle-5.5.2.44.b83 que j'ai téléchargé sur le site¹ éducation de TI. Un bundle comporte le dernier système (OS) de la calculatrice ainsi que les programmes de base, dont Python. Il suffit, la calculatrice connectée par USB, d'utiliser le menu Actions/Envoyer l'OS Bundle... de TI-CONNECT.

En entrant dans le gestionnaire de scripts, je vois ci-dessus que la mise à jour a installé trois scripts : BONJOUR, GRAPHIQ et LINREG. Notez les majuscules et le nombre limité de caractères dans le nom d'un script.

1 Le Shell

Avant d'écrire un programme Python avec des fonctions etc. nous examinons le *shell* (ou *toplevel*) qui est un langage de commande permettant notamment faire des calculs comme avec n'importe quelle calculatrice mais en langage Python...

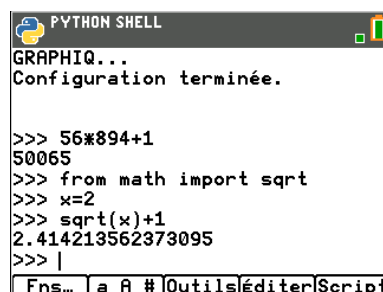
En bas de la fenêtre du gestionnaire de scripts, vous avez remarqué les 5 "boutons" :



qui ne sont pas vraiment des boutons car la calculatrice n'est pas tactile (oui, un stylet serait le bienvenu). Les boutons sont en-dessous, sur la première ligne de la calculatrice, nommés F1, F2, F3, F4, F5 de manière générale :



Nous pressons donc *trace* pour entrer dans le *shell* qui présente son prompt `>>>` qui vous invite à entrer une expression Python qui sera soumise à évaluation lorsque vous aurez pressé la touche *entrer* en bas à droite :



Pas de problème pour entrer au clavier les chiffres, opérateurs, parenthèses. Mais comment entrer des lettres (minuscules en général en Python) ? Une touche porte

¹ <https://education.ti.com/fr/search?q=bundle>

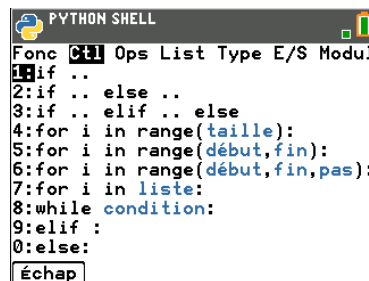
souvent une action principale (pour 0, envoyer le chiffre 0 à l'écran), et deux autres actions, **bleue** (*catalog* pour lister les fonctions Python) ou **verte** (l'espace `_`). Les touches **bleue** (2nde) et **verte** (alpha) permettent d'y avoir accès. Un peu pénible...

Une autre solution consiste à taper sur le bouton *a A #* (F2) pour faire apparaître une fenêtre permettant de composer un texte qui sera envoyé au shell en appuyant sur le bouton *Coller* (F5). Le bouton *Echap* (F1) permet toujours de revenir à l'écran précédent. Encore plus pénible...

Heureusement la calculette offre des raccourcis pour entrer des noms de fonctions, de modules, rédiger une boucle, etc. en tapant sur le bouton *Fns* (F1) qui fait apparaître une fenêtre avec des boutons en haut cette fois et accessibles avec les flèches :

Fonc	Ctl	Ops	Type	E/S	Modul
------	-----	-----	------	-----	-------

Par exemple pour importer le module *math*, plutôt que taper la ligne entière, appuyez sur le bouton *Modul* suivi du choix 1. Pour définir une fonction, tapez sur *Fonc* et complétez la définition. Pour débiter une boucle *for*, tapez sur *Ctl*, etc. Familiarisez-vous avec cette méthode ! La touche *catalog* (0 bleu) du clavier est aussi bien pratique.

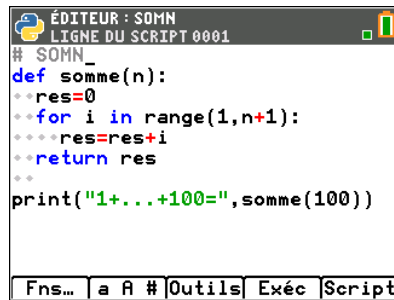


Le **shell** vous propose donc une calculette en Python. Mais en général, on n'écrit pas de texte compliqué ni de définition de fonction au shell, on le fait dans un **script** : un texte de programme Python, sur plusieurs lignes, qui sera exécuté et sauvegardé (dans la calculette ou sur l'ordinateur). L'exécution enverra les affichages du programme dans le shell, où l'on pourra continuer à tester les fonctions du programme. Mais il est recommandé de prévoir les tests dans le script lui-même, pour ne pas avoir à les retaper chaque fois. Passons donc à l'éditeur de scripts.

2 L'éditeur de scripts

Depuis le shell, je retourne dans l'éditeur de scripts en tapant la touche *Script* (F5). Je peux voir les scripts existants. Nous allons rédiger un nouveau script calculant la somme des entiers de 1 à n. Tapons sur *Nouv* (F3) et nommons-le SOMN, suivi de la touche *Ok* (F5). Je suis dans une fenêtre vierge, dans laquelle je rentre mon programme. Cet *Appvar Python* a une indentation de 2 espaces, dont acte. Sur ordinateur, c'est plutôt 4 espaces, mais nous sommes un peu à l'étroit, alors...

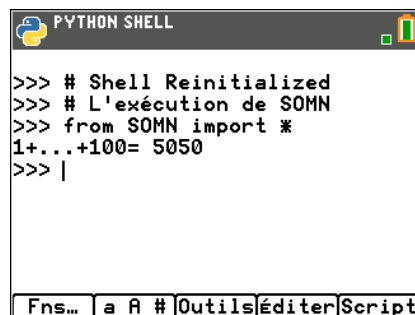
Rédigeons ce script SOMN contenant une fonction *somme(n)* avec n entier naturel et qui retourne la somme $1+2+\dots+n$. Ce script contiendra un test d'exécution. Le résultat est bien connu $n(n+1)/2$ (Gauss l'avait montré à son prof à 7 ans), mais c'est juste histoire d'écrire une boucle...



```
ÉDITEUR : SOMN
LIGNE DU SCRIPT 0001
# SOMN_
def somme(n):
    res=0
    for i in range(1,n+1):
        res=res+i
    return res
print("1+...+100=",somme(100))

Fns... a A # Outils Exéc Script
```

Le commentaire # en première ligne indique le nom du script SOMN, sans l'extension car ce sera .py sur ordinateur et .8xv sur la calculette. La dernière ligne sera le test, parfois il en faudra plusieurs. Il reste à exécuter ce script. Euh, le bouton *Exec* ?



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de SOMN
>>> from SOMN import *
1+...+100= 5050
>>> |

Fns... a A # Outils Éditer Script
```

Entre nous, je me serais bien passé de cet *import*, mais pourquoi pas. S'il y avait eu une erreur (mais vous n'en faites jamais, n'est-ce pas ?), il suffirait de reprendre l'édition avec le bouton *Editer*. qui vous renvoie dans l'éditeur de script pour entrer dans une infernale boucle de mise au point : modifier-exécuter-modifier-exécuter-... Evitez cela en réfléchissant calmement sur votre programme, ne jouez pas les essais rapides avec des corrections successives, surtout avec des éditeurs aussi pénibles !

Une fois que vous êtes satisfait de votre script, il sera visible dans le gestionnaire de scripts, à la suite des scripts précédents et sera toujours là lorsque vous rallumerez votre calculette plus tard. Mais il est conseillé de sauvegarder vos scripts sur un **ordinateur** si vous en possédez un, ce qui servira au passage à continuer le travail sur ordinateur (avec le python de www.python.org par exemple), ce qui est bien plus confortable, pour envoyer le programme final sur la calculette pour l'exécuter au lycée.

3 Sauvegarder un script sur ordinateur

Votre calculette est connectée par USB à l'ordinateur, et vous lancez le programme TI-CONNECT sur l'ordinateur. Notez au passage que ce programme connecté à la calculette, permet de prendre des photos de l'écran de la calculette, c'est ce que j'ai fait dans ce document.

3.1 Calculette Ordinateur

Mon script SOMN est dans le gestionnaire de scripts, je veux l'envoyer vers mon ordinateur. Soit je conserve le format .8xv de *AppVar Python* si je souhaite simplement stocker le script sur ordinateur sans y toucher. Soit je le transforme en vrai fichier *Python* usuel d'extension .py. Dans TI-CONNECT, je clique dans l'icône sur la gauche :



qui me montre le contenu de la calculette. Je localise le fichier SOMN et par un clic droit, j'opte pour *Envoyer à l'ordinateur...*, en choisissant le format *Python (.py)* ou *AppVar*

Python (.8xy). Pour cet essai, j'opte pour Python dans mon répertoire PYTHON-TI83 afin de continuer à modifier et exécuter le script sur ordinateur. Je note que l'indentation de deux espaces est conservée, mais mon Python ne bronche pas. So far so good.

3.2 Ordinateur Calculatrice


J'ai modifié le fichier SOMN.py sur mon ordinateur avec un Python simple et solide (www.python.org). Maintenant je veux l'envoyer vers le gestionnaire de scripts de la calculatrice. Dans le menu *Actions* de TI-CONNECT, je choisis *Ajouter des fichiers depuis un ordinateur...* en n'oubliant pas de cocher la case du bas *Remplacer le contenu...* car un fichier de même nom existe déjà. Il vous faudra peut-être quitter le mode Python sur la calculatrice. Je vérifie que mon fichier est bien celui qui provient de l'ordinateur.


Ah, peut-être avez-vous remarqué le choix possible de l'endroit de la calculatrice où charger le fichier : **Mémoire RAM** ou **Archive**. Quelle est la différence ?

L'application Python de la calculatrice ne peut modifier et exécuter que les fichiers qui sont en mémoire RAM, ce qui est le choix usuel pour travailler. Vous conserverez votre fichier SOMN comme *archive* si :

- ou bien la calculatrice vous prévient qu'il n'y a plus assez de mémoire RAM.
- ou bien vous n'utilisez pas le fichier dans un avenir proche.

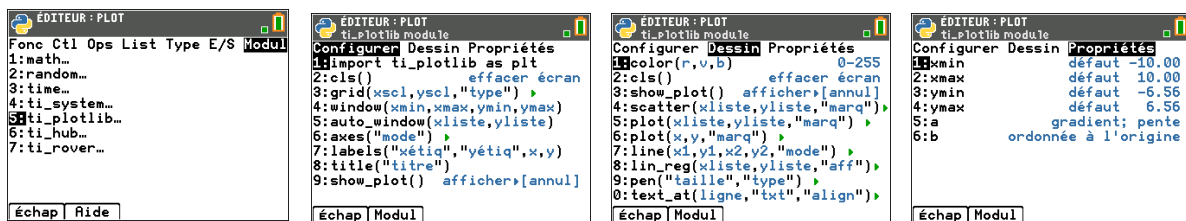
Les **archives** contiennent donc du stockage à long terme, la mémoire **RAM** étant dévolue aux travaux à court terme. Pour déplacer un fichier entre ces deux zones :

RAM  Archives : Appuyez sur la touche bleue **mém** (bleu) situé sur la touche +, puis *Gest. Mémoire/Suppr...* puis sur *B:Var App* puis sur le fichier SOMN. Vous vérifierez alors que le script SOMN n'est plus dans le gestionnaire de scripts.

Archives  RAM : Mon fichier SOMN est archivé. Je veux l'utiliser dans la calculatrice, donc je dois l'installer dans la RAM. Même opération que ci-dessus en cliquant sur le fichier SOMN. Vous voyez d'ailleurs un indicateur PY sur la droite.

4 Le module **ti_plotlib**

La quasi-norme scientifique en Python pour tracer des courbes est le gros module **matplotlib** qui n'est pas disponible pour la *TI-83 CE Edition Python*. Pour y pallier, TI fournit un module maison **ti_plotlib** qui reprend quelques fonctions de base de matplotlib, mais en versions réduites. Vous avez la liste des fonctions Python de ce module en tapant *Fns...* (F1) puis vous dirigez à droite sur *Modul* puis *5:ti_plotlib...*



Regardez la vidéo https://www.youtube.com/watch?v=_HVHPyVkrRw qui montre aussi l'utilisation de l'émulateur *TI-Smartview* cité plus bas. Soyez conscients que notre science regarde vers le numérique et travaillez Python chez vous sur ordinateur, et au lycée sur calculatrice ("TI-83" ou "TI-Nspire CX II-T CAS") ou tablette (iPad ou Android).

NB. Il existe deux **émulateurs** de la calculatrice *TI-83 CE* avec Python :

- un émulateur officiel **TI-Smartview** de Texas Instruments (90 jours gratuits) sur :

https://tiplanet.org/forum/archives_voir.php?id=2644799

- une recherche sur le web propose aussi un émulateur "CEmu" sur "GitHub". Non testé.