



Algorithmes et programmation en SVT.

Nouveaux programmes du lycée, rentrée 2019.

SVT : une discipline en prise avec l'évolution des connaissances et des technologies.

- **Préambule du programme** (2^{nde} et 1^{ère}) : le **numérique et les SVT** [...] développement de **compétences numériques** : l'usage [...] de la **programmation** [...]. Vers de nouvelles perspectives de formation comme la **bio-informatique** ou **l'exploitation de données**. Les SVT intègrent naturellement dans leurs pratiques les acquis des autres disciplines scientifiques, en particulier la physique-chimie et **l'informatique**.
- **Des capacités à mettre en œuvre** : Calculer le nombre de combinaisons possibles de séquences de n nucléotides de longueur quand n grandit. Concevoir un algorithme de traduction d'une séquence d'ARN et éventuellement le programmer dans un langage informatique (par exemple Python)
- **Toute activité mettant en œuvre des calculs ou traitements de données** permet de programmer un algorithme afin d'obtenir le résultat automatiquement pour plusieurs situations.

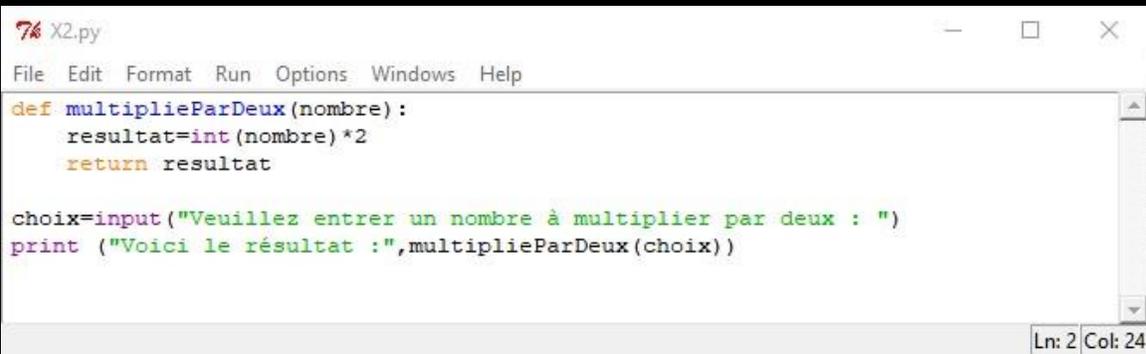
Un langage de programmation : PYTHON



- Utilisé en mathématiques.
- Libre et gratuit. Simple pour débiter.
- Utilisé dans l'enseignement supérieur.
- Possibilité d'ajouter de nouvelles bibliothèques.
- Il suffit de l'installer et il est fonctionnel.
- Il existe différentes interfaces pour l'utiliser.

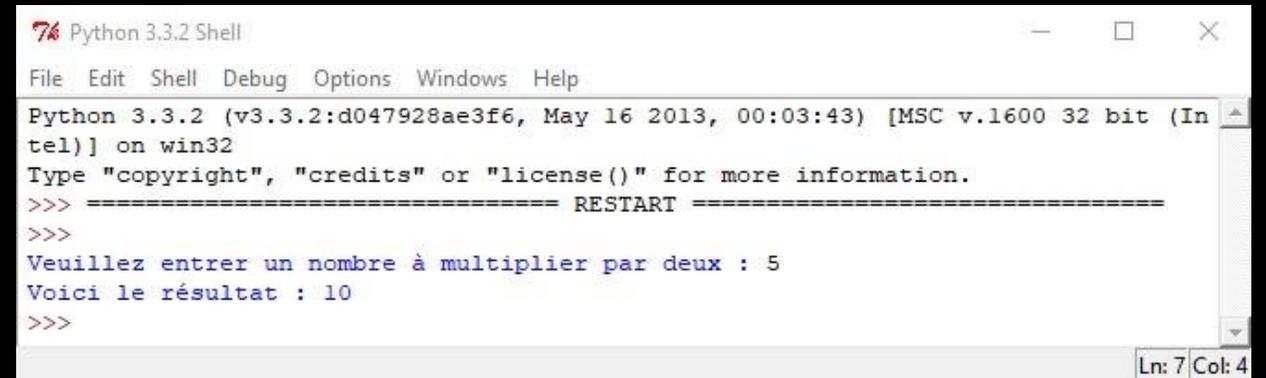
Les interfaces pour programmer en PYTHON

- **IDLE** : le Python « de base ». Il s'agit en fait d'un simple éditeur de texte dans lequel on écrit son code. Puis on enregistre son code et on l'exécute. L'exécution se fait alors dans une autre fenêtre (l'interpréteur ou shell).



```
7% X2.py
File Edit Format Run Options Windows Help
def multiplierParDeux(nombre):
    resultat=int(nombre)*2
    return resultat

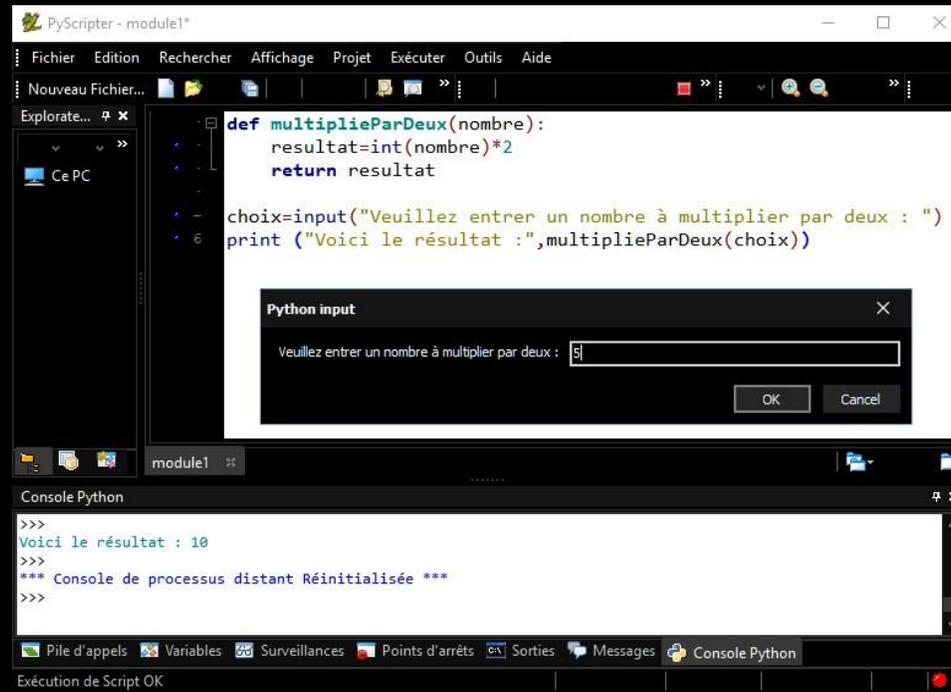
choix=input("Veuillez entrer un nombre à multiplier par deux : ")
print ("Voici le résultat :",multiplierParDeux(choix))
Ln: 2 Col: 24
```



```
7% Python 3.3.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:03:43) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Veuillez entrer un nombre à multiplier par deux : 5
Voici le résultat : 10
>>>
Ln: 7 Col: 4
```

Les interfaces pour programmer en PYTHON

- **Edupython** : il s'agit d'une interface spécialement étudiée pour l'éducation. Dans une fenêtre divisée en trois on trouve l'arborescence des disques du PC, l'éditeur de code (équivalent de IDLE) et la console d'exécution (équivalent du shell). Contient déjà plusieurs bibliothèques préinstallées et de nombreuses documentations sur <https://edupython.tuxfamily.org/> .Peut fonctionner sans installation en version portable (373Mo).



```
def multiplierParDeux(nombre):  
    resultat=int(nombre)*2  
    return resultat  
  
choix=input("Veuillez entrer un nombre à multiplier par deux : ")  
print ("Voici le résultat :",multiplierParDeux(choix))
```

Python input

Veuillez entrer un nombre à multiplier par deux : 5

OK Cancel

Console Python

```
>>>  
Voici le résultat : 10  
>>>  
*** Console de processus distant Réinitialisée ***  
>>>
```

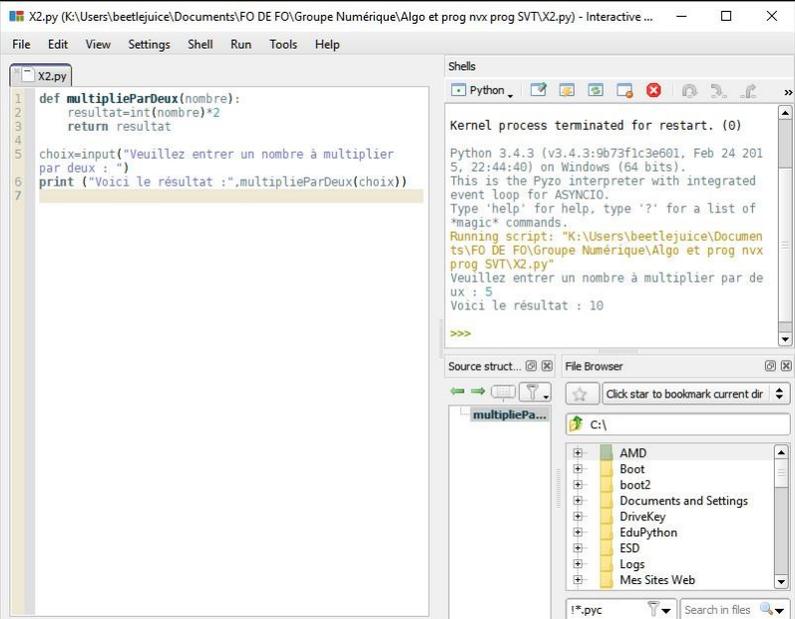
Pile d'appels Variables Surveillances Points d'arrêts Sorties Messages Console Python

Exécution de Script OK



Les interfaces pour programmer en PYTHON

- **Pyzo** : il s'agit d'un environnement de développement. Il doit être installé après avoir installé Python. On le paramètre ensuite pour qu'il utilise Python. A l'instar d'EduPython, on retrouve une fenêtre divisée en trois avec l'arborescence des disques du PC, l'éditeur de code (équivalent de IDLE) et la console d'exécution (équivalent du shell).



```
File Edit View Settings Shell Run Tools Help

x2.py
1 def multiplieParDeux(nombre):
2   resultat=int(nombre)*2
3   return resultat
4
5 choix=input("Veuillez entrer un nombre à multiplier
6 par deux : ")
7 print ("Voici le résultat :".multiplieParDeux(choix))

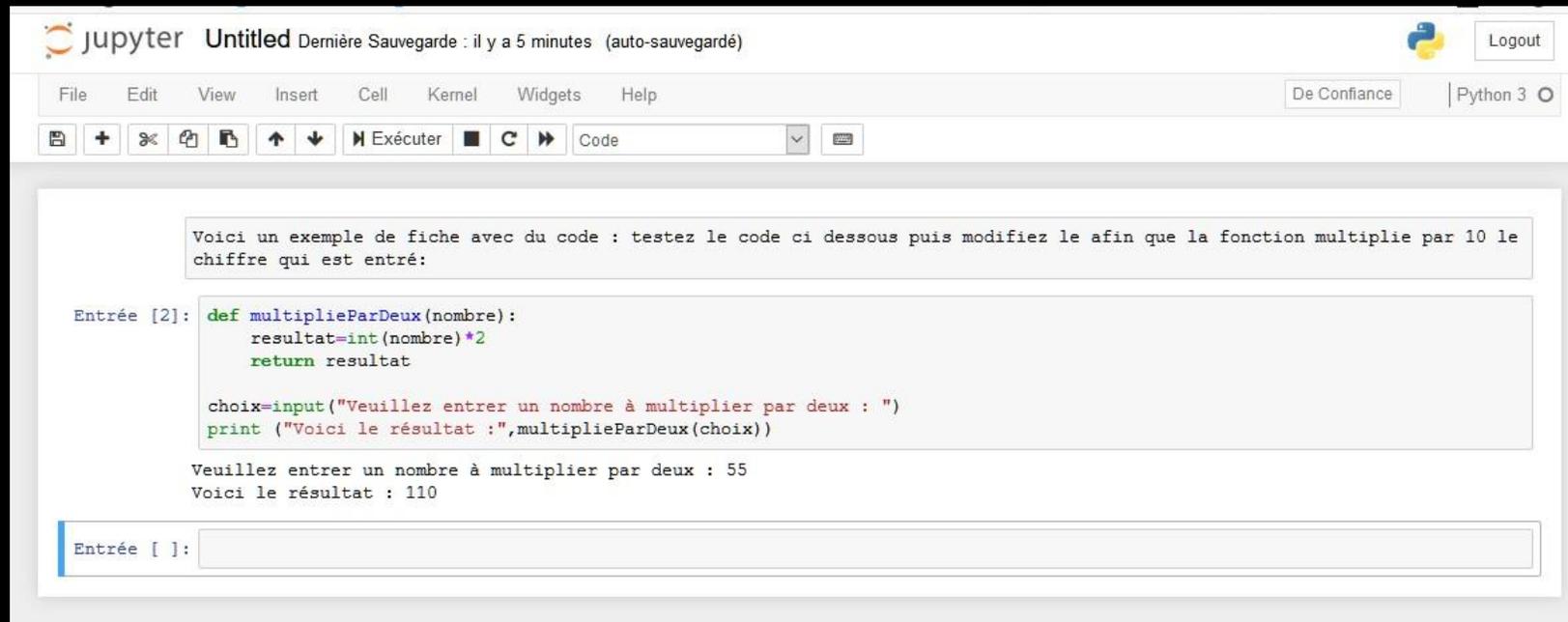
Shells
Python
Kernel process terminated for restart. (0)
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 201
5, 22:44:40) on Windows (64 bits).
This is the Pyzo interpreter with integrated
event loop for ASYNCIO.
Type 'help' for help, type '?' for a list of
'magic' commands.
Running script: "K:\Users\beetlejuice\Documen
ts\FO DE FO\Groupe Numérique\Algo et prog nvx
prog SVT\X2.py"
Veuillez entrer un nombre à multiplier par de
ux : 5
Voici le résultat : 10
>>>

Source struct... File Browser
Click star to bookmark current dir
C:\
AMD
Boot
boot2
Documents and Settings
DriveKey
EduPython
ESD
Logs
Mes Sites Web
1*.pyc Search in files
```



Les interfaces pour programmer en PYTHON

- **JUPYTER notebook** : il s'agit d'un environnement qui permet de créer des fiches dans lesquelles le code peut être proposé, modifié, créé et exécuté. Il s'installe comme une bibliothèque python.



The screenshot shows the Jupyter Notebook interface. At the top, there is a header with the Jupyter logo, the text "jupyter Untitled", and a save status "Dernière Sauvegarde : il y a 5 minutes (auto-sauvegardé)". On the right, there is a Python logo and a "Logout" button. Below the header is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar, there is a "De Confiance" button and a "Python 3" dropdown menu. Below the menu bar is a toolbar with various icons for file operations, navigation, and execution. The main area contains a code cell with the following text:

Voici un exemple de fiche avec du code : testez le code ci dessous puis modifiez le afin que la fonction multiplie par 10 le chiffre qui est entré:

```
Entrée [2]: def multiplieParDeux(nombre):  
            resultat=int(nombre)*2  
            return resultat  
  
            choix=input("Veuillez entrer un nombre à multiplier par deux : ")  
            print ("Voici le résultat :",multiplieParDeux(choix))
```

Voici le résultat : 110

Entrée []:



Des exemples d'applications :

- On peut se demander à partir d'un nombre de bactéries de départ dans un aliment, combien on peut obtenir théoriquement de bactéries au bout d'un temps donné si on reste en phase exponentielle en partant du principe que cette bactérie se divise toutes les 20 mn.
- L'élève doit alors proposer un algorithme permettant de réaliser ce calcul :
- 1- Je demande le temps en minutes que je veux étudier.
- 2- Je demande le nombre de bactéries au départ.
- 3- Je divise ce temps par 20.
- 4- Je fais une boucle d'autant de tours que le résultat précédent.
- 5- A chaque tour de boucle je multiplie par 2 mon résultat précédent.
- 6- A la fin de ma boucle j'affiche le résultat.

Des exemples d'applications :

- La programmation en python se fait alors via une fonction que l'on va créer et qui prend comme arguments le nombre de bactéries au départ et le nombre de minutes de l'expérimentation :

```
def devBacterien(nbDepart,minutes):  
    boucles=int(minutes)/20  
    resultat=int(nbDepart)  
    for i in range(int(boucles)):  
        resultat=resultat*2  
    return resultat  
nbDepart = input ("Donnez le nombre de bactéries au départ : ")  
minutes = input ("Donnez le nombre de minutes de l'expérience :")  
print ("On obtient ",devBacterien(nbDepart,minutes), " Bactéries !")
```

```
Donnez le nombre de bactéries au départ : 1  
Donnez le nombre de minutes de l'expérience :600  
On obtient 1073741824 Bactéries !
```

Des exemples d'applications :

- On peut réaliser un programme permettant de créer l'ARN pré-messager à partir d'une séquence de nucléotides d'un gène.
- L'élève doit alors proposer un algorithme permettant de réaliser ce programme :
- 1- Je demande une séquence de nucléotides ADN.
- 2- Je parcours alors cette chaîne de caractères ADN dans une boucle et je construis à chaque tour de boucle une chaîne ARN qui ajoute le complémentaire du nucléotide lu.
- 3- A la fin de ma boucle j'affiche le résultat.

Des exemples d'applications :

```
def adnArn(seqAdn):
    arn=""
    for i in seqAdn:
        if i=="A":
            arn=arn+"U"
        elif i=="T":
            arn=arn+"A"
        elif i=="C":
            arn=arn+"G"
        elif i=="G":
            arn=arn+"C"
        else:
            print("La séquence de votre gène n'est pas correcte. Seules les nucléotides A,T,C et G sont acceptés.")
            break
    return arn
seqAdn = input ("Donnez la sequence du gène : ")
print ("On obtient l'ARN messenger suivant : ",adnArn(seqAdn))
```

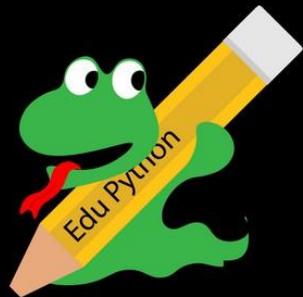
Donnez la sequence du gène : ATGGTCCCT

On obtient UACCAGGGA

Pourquoi programmer en SVT ?

- L'idée est de tester si un algorithme fonctionne...
- Avec la machine il faut être RIGOUREUX et LOGIQUE...
- Le fait de préparer quelques fonctions et de demander à l'élève de les placer dans le bon ordre pour que cela fonctionne permet de travailler la logique...
- Faire un peu de programmation permet de comprendre qu'il faut être rigoureux, qu'il faut savoir ce que l'on veut faire avant de faire, et qu'au prix de quelques efforts on arrive à faire fonctionner un programme...

*L'informatique n'est pas plus la science des ordinateurs que
L'astronomie n'est celle des télescopes.
Michael Fellows et Ian Parberry*



python

