



RÉGION ACADÉMIQUE  
PROVENCE-ALPES-CÔTE D'AZUR

MINISTÈRE  
DE L'ÉDUCATION NATIONALE  
MINISTÈRE  
DE L'ENSEIGNEMENT SUPÉRIEUR,  
DE LA RECHERCHE  
ET DE L'INNOVATION



# Aménagement du programme de Mathématiques de seconde 2017

## Atelier Algorithme et Programmation

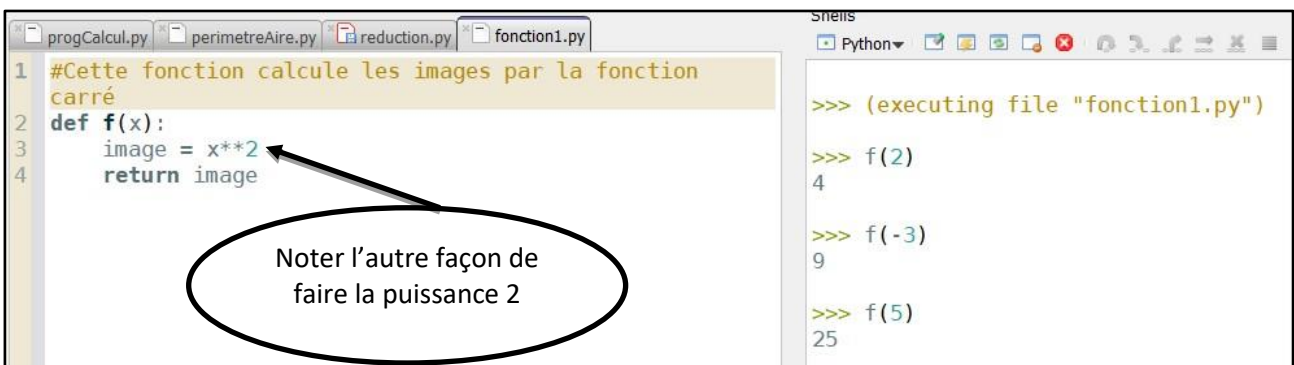
## 1- Rappels sur les fonctions en python :

La notion de fonction en programmation est la grande nouveauté de l'aménagement de programme. Elle permet de structurer le programme en le découpant en sous-programmes indépendants. Elle permet de revisiter la notion de fonction en mathématiques avec qui elle possède un lien fort.

La structure générale d'une fonction sous Python est la suivante :

<pre>def nomFonction(parametres) :     instructions     return valeur</pre>	<ul style="list-style-type: none"><li>• <i>def</i> indique le début de la fonction.</li><li>• <i>nomFonction</i> est le choix du nom de la fonction.</li><li>• Les « : » marquent le début de la fonction</li><li>• Toutes les instructions de la fonction sont reconnues grâce à l'indentation (décalage vers la droite). La fonction se termine lorsque l'indentation (décalage sur la droite) se termine. L'indentation remplace { } dans d'autres langages.</li><li>• La commande « return » permet de renvoyer une ou plusieurs valeurs</li></ul>
---	--

**Exemple 1 :** Le programme suivant calcule l'image par une fonction. Recopier puis exécuter le programme ci-dessous. Puis appeler la fonction dans le « shell » comme ci-dessous.

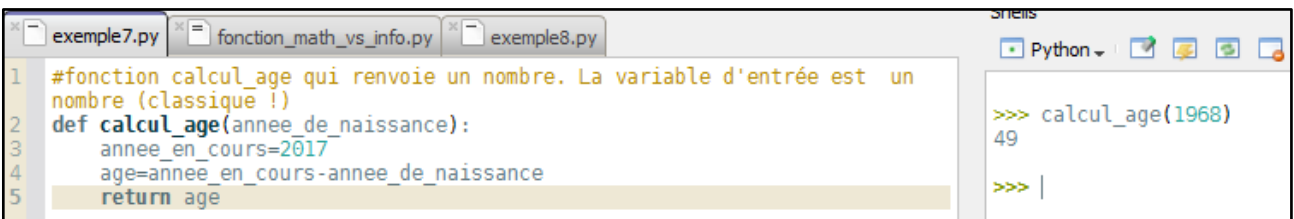


```
progCalcul.py | perimetreAire.py | reduction.py | fonction1.py  
1 #Cette fonction calcule les images par la fonction  
  carré  
2 def f(x):  
3     image = x**2  
4     return image  
  
Shell  
Python  
>>> (executing file "fonction1.py")  
>>> f(2)  
4  
>>> f(-3)  
9  
>>> f(5)  
25
```

Noter l'autre façon de faire la puissance 2

**Exemple 2:** Un autre exemple avec un paramètre (le calcul d'âge)

Recopier puis exécuter le programme ci-dessous



```
exemple7.py | fonction_math_vs_info.py | exemple8.py  
1 #fonction calcul_age qui renvoie un nombre. La variable d'entrée est un  
  nombre (classique !)  
2 def calcul_age(annee_de_naissance):  
3     annee_en_cours=2017  
4     age=annee_en_cours-annee_de_naissance  
5     return age  
  
Shell  
Python  
>>> calcul_age(1968)  
49  
>>> |
```

**Exemple 3 :** Un exemple avec plusieurs paramètres.

La fonction *calc\_long\_segment()* calcule la longueur d'un segment à partir des coordonnées des points.

Recopier puis exécuter le programme ci-dessous. Puis appeler la fonction dans le « shell ».

```

exemple7.py  fonction_math_vs_info.py  exemple8.py
1 #importation du module "math" pour l'utilisation des fonctions puissance
2 (pow) et racine carré (sqrt)
3 from math import *
4 #fonction calcul_long_segment qui renvoie un nombre. 4 variables d'entrée
5 def calcul_long_segment (xA,yA,xB,yB):
6     long=sqrt(pow(xB-xA,2)+pow(yB-yA,2))
7     return long

Shells
Python
>>> calcul_long_segment(1,3,5,6)
5.0
>>> |

```

**Remarques :**

- La fonction utilise plusieurs paramètres et renvoie un nombre.
- Dans cet exemple, la bibliothèque *math* est appelée (aucun besoin de l'installer, elle est présente par défaut dans la distribution Pyzo. Il est néanmoins nécessaire de la charger). Dans le paragraphe 3-4, nous verrons qu'il existe d'autres bibliothèques qui nécessitent parfois une installation supplémentaire.

**Exemple 4:** Deux exemples sans paramètres avec une ou plusieurs valeurs renvoyées.

Recopier puis exécuter le programme ci-dessous. Puis appeler les deux fonctions dans le « shell ».

```

from random import randint

#fonction pile qui renvoie un vrai si la face pile est tirée . pas de variable d'entrée
def pile():
    piece=randint(0,1)
    if piece==0:
        result=True
    else:
        result=False
    return result

#fonction de_a_5 qui renvoie un vrai si la face 5 d'un dé à 6 faces est tirée et le numéro
de dé tiré . pas de variable d'entrée
def de_a_5():
    face=randint(1,6)
    if face==5:
        result=True
    else:
        result=False
    return (result,face)

>>> pile()
True
>>> de_a_5()
(False, 4)
>>>

```

**Remarque :**

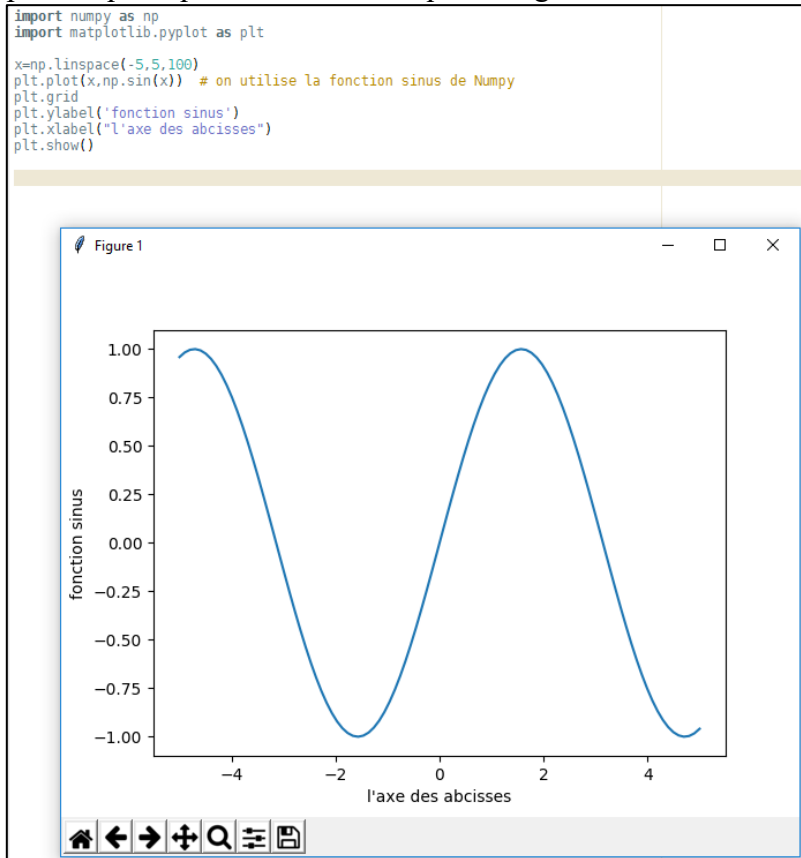
- Dans cet exemple, la bibliothèque *random* est appelée (aucun besoin de l'installer, elle est présente par défaut dans python). Dans cette méthode d'appel, seul la fonction *randint()* est chargée. Cela évite de charger toute la bibliothèque pour n'utiliser qu'une seule fonction.
- La fonction *randint()* permet de tirer un nombre entier aléatoire entre deux bornes comprises.
- Les deux fonctions *pile()* et *de\_a\_5()* n'ont pas de paramètres d'entrée . *pile()* renvoie un booléen et *de\_a\_5()* renvoie un entier et un booléen.
- Plusieurs fonctions peuvent coexister dans le même script.

**2- Utilisation des bibliothèques :**

Certaines fonctions existent déjà en python ( fonctions mathématiques, fonctions aléatoires, obtenir des représentations graphiques, faire du calcul matriciel )....Pour les utiliser, Il est nécessaire d'utiliser des bibliothèques Python. Nous avons vu précédemment que les bibliothèques *math* ou *randint* étaient intégrées dans Python et qu'il suffisait de les charger en début de programme. C'est différent pour les deux bibliothèques utilisées dans l'exemple ci-dessous : **Numpy** et **matplotlib** nécessitent une installation qui est simple dans *pyzo*. Il suffit d'écrire dans le shell :

```
>>> pip install numpy
>>> pip install matplotlib
```

Dans l'exemple ci-dessous les bibliothèques *numpy* et *matplotlib* sont appelées et utilisées. La déclaration faite de la manière suivante (**Import matplotlib.pyplot as plt** ) permet d'utiliser toutes les fonctions de la bibliothèque *matplotlib.pyplot* en lui associant le préfixe *plt* (*plt.plot*, *plt.grid* ....) . On peut imaginer qu'une autre bibliothèque pourrait utiliser une fonction plot. En la déclarant ainsi, il n'y a pas d'équivoque sur la bibliothèque d'origine.



Recopier puis exécuter le programme ci-dessus.

### 3- A vous de programmer

#### Vecteurs Egaux....

L'objectif est de concevoir et implanter des fonctions informatiques sur les vecteurs

- 1) Concevoir et implanter une fonction informatique qui calcule les coordonnées d'un vecteur connaissant les coordonnées des extrémités d'un représentant. La fonction renvoie les 2 coordonnées du vecteur.
- 2) Concevoir et implanter une fonction informatique qui teste si deux vecteurs sont colinéaires. La fonction renvoie « true » ou « false ».
- 3) Concevoir et implanter une fonction informatique qui teste 3 points de coordonnées données sont alignés. La fonction renvoie « true » ou « false ».

#### Recherche dichotomique....

Concevoir et implanter une fonction informatique qui détermine une valeur approchée de la solution de l'équation  $f(x) = 0$  par dichotomie.

- 1) Commencer par concevoir et implanter une fonction qui renvoie l'image d'un réel  $x$  par la fonction  $f$  définie par  $f(x) = x^2 - 2$ .
- 2) Puis concevoir et implanter la fonction « dichotomie » ayant pour paramètre les bornes de l'intervalle de recherche et la précision. Cette fonction renvoie les bornes de l'intervalle encadrant la solution à la précision près.
- 3) Enfin modifier la fonction précédente pour que la fonction  $f$  soit un paramètre de la fonction de « dichotomie ».

#### Dé....

- 1) On lance « n » fois un dé à 6 faces équilibrés. Concevoir et implanter une fonction informatique qui simule les « n » lancers et renvoie le nombre de fois qu'on obtient la face « k ».
- 2) **Situation Problème :** (tiré d'une épreuve sur Dossier Capes 2016)  
Arnaud et Bernard jouent à un jeu de dés. Le jeu consiste pour chacun d'eux à choisir un nombre compris entre 3 et 18. Puis chaque joueur lance 3 dés cubiques, calcule la somme des nombres portés par les 3 faces supérieures.  
Arnaud choisit le nombre 9 et Bernard le nombre 10. Qui a le plus de chance de gagner ?
  - a. Concevoir et implanter une fonction informatique qui simule « n » tirages de la somme de 3 dés équilibrés et renvoie 9 si 9 est plus fréquent, 10 si 10 est plus fréquent, et 0 si 9 et 10 sont aussi fréquents.
  - b. Simuler plusieurs fois en modifiant le nombre de tirages ( $n=100, 1000, 10000, 100000$ ). Que peut-on conclure ?
  - c. Concevoir et implanter une fonction qui permet de répéter « nExperiences » fois l'expérience ci-dessus en faisant apparaître le nombre qui a été le plus fréquent. Reprendre la question b.

## Echantillonnage ...

L'objectif est de concevoir et d'implanter des fonctions informatiques permettant d'étudier l'intervalle de fluctuation de seconde.

- 1) Concevoir et implanter « n » répétitions indépendantes d'une expérience dont le succès est d'obtenir un caractère de proportion « p ». Cette fonction renvoie la fréquence des expériences réussies.
- 2) Concevoir et implanter une fonction informatique qui calcule l'intervalle de fluctuation de la fréquence de « n » répétitions indépendantes d'une expérience dont le succès est d'obtenir un caractère de proportion « p ». La fonction renvoie les bornes de l'intervalle de fluctuation
- 3) Concevoir et implanter une fonction informatique qui simule la prise de « nbEchantillons » échantillons de taille « n » lié à une proportion « p » d'un caractère. Elle renvoie la proportion d'échantillons qui donnent une fréquence du caractère dans l'intervalle de fluctuation.
- 4) Concevoir une fonction informatique qui représente graphiquement la simulation précédente. (axe des abscisses : numéro de l'échantillon, axe des ordonnées : fréquence du caractère dans l'échantillon). On fera apparaître l'intervalle de fluctuation

## Changer de base ....

- 1) Concevoir et implanter une fonction informatique qui convertit un nombre écrit en base 10 en base 2.
- 2) Concevoir et implanter une fonction informatique qui convertit un nombre écrit en base 2 en base 10.

## Moyenne et médiane

Concevoir et implanter une fonction informatique qui détermine La moyenne et la médiane d'une série contenue dans une liste. On pourra générer les éléments de la série à partir d'une fonction produisant une liste de 100 nombres aléatoires compris entre 1 et 200 par exemple....