



Cliquer sur le lien
du site officiel

Premiers pas avec Python

Download Python 3.6.1

Download Python 2.7.13

Cliquer sur le lien
du téléchargement



L'origine de Python

Le langage Python est né dans les années 1990, au CWI Amsterdam, développé par Guido Van Rossum. Il est nommé ainsi par référence aux « Monty Python », célèbre groupe d'humoristes britanniques, très appréciés par Van Rossum.

Les deux versions récentes de Python sont : la version 2.7 et la version 3.6.

A noter que les versions 3.x ne sont pas compatibles avec les versions 2.x.



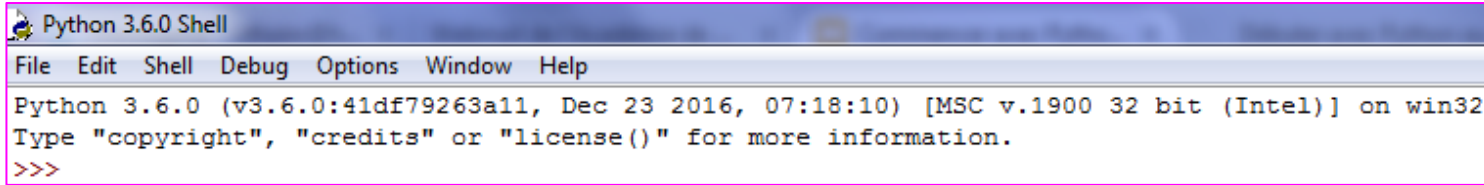
Python : un langage de programmation

- Python est un langage de programmation professionnel, libre et gratuit.
- C'est un **langage interprété** ce qui signifie qu'il n'y a pas de phase de compilation qui traduit le programme en langage machine (comme c'est le cas pour les langages C ou C++) mais les instructions sont traitées au fur et à mesure de leur lecture par l'interprète.
- Il est portable, c'est à dire qu'il peut fonctionner sous différents systèmes d'exploitation comme Windows, linux, Mac Os, ...
- Quel que soit le système d'exploitation, on peut utiliser Python dans un terminal ou avec **IDLE** (environnement de développement fourni à l'installation).

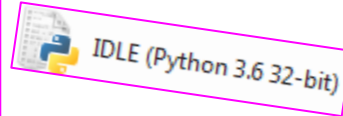
```
import math

def perimetre_cercle(un_rayon):
    """Calculer le périmètre d'un cercle à partir de son rayon.
    :param un_rayon: Le rayon du cercle (positif)
    :return Le périmètre d'un cercle de rayon un_rayon
    """
    diametre = 2 * un_rayon
    return math.pi * diametre
```

Lancer la console Python, après installation



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```



→ Sous Windows :

Passer par les menus Démarrer > Tous les programmes > Python 3.6 > **IDLE** (Python 3.6-32 bit)

→ **Sous Linux** : Lorsque vous l'avez installé sur votre système, Python a créé un lien vers l'interpréteur sous la forme **python3.X** (le X étant le numéro de la version installée).

→ **Sous Mac OS X** : Cherchez un dossier Python dans le dossier Applications. Pour lancer Python, ouvrez l'application IDLE de ce dossier.

Un exemple de programme en 2nde

Énoncé : Ecrire un programme qui affiche la longueur d'un segment [AB].

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v
tel] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from math import*
>>> # Calcul de la longueur d'un segment [AB]

>>> XA=float(input("Donner l'abscisse de A : "))
Donner l'abscisse de A : 3
>>> YA=float(input("Donner l'ordonnée de A : "))
Donner l'ordonnée de A : -2
>>> XB=float(input("Donner l'abscisse de B : "))
Donner l'abscisse de B : 4
>>> YB=float(input("Donner l'ordonnée de B : "))
Donner l'ordonnée de B : -1
>>> L=sqrt((XB-XA)**2+(YB-YA)**2)
>>> print("La longueur L est égale à:"), L
La longueur L est égale à:
(None, 1.4142135623730951)
```

Il est possible d'utiliser directement Python en mode commande. Voici un exemple répondant au problème posé.


Programme édité avec Idle sous Windows

- ★ Le plus souvent, le programme est édité avec IDLE, puis sauvegardé avec l'extension `.py` et lancé à l'aide de la touche **F5 (Run/Run Module)**.
- ★ Voici un exemple de programme en version 2.7:
Le script se trouve dans l'IDLE et il s'exécute dans l'interpréteur « shell ».

```
7% intersection_listes.py - C:\Repertoires\Sylvaine\Python\intersection_listes.py
File Edit Format Run Options Windows Help
#intersection
T1 = list(input("entrer la première liste d'entiers : "))
T2 = list(input("entrer la deuxième liste d'entiers : "))
T3 = []
for i in range(len(T1)) :
    for j in range(len(T2)) :
        if T1[i] == T2[j]:
            T3.append(T1[i])
print T3

7% Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7 (r27:82525, Jul  4 2010, 09:01:
59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()"
for more information.
>>> ===== RESTA
RT =====
>>>
entrer la première liste d'entiers : 8,25,
91,45,63,48,18,22
entrer la deuxième liste d'entiers : 42,10
1,.37,89,63,25,87
[25, 63]
>>>
```

Et pour Linux :



The image shows a screenshot of a Python IDE with two windows. The left window, titled 'Hello_world.py - /home/jcottin/Bureau/Hello_world.py', contains the following Python code:

```
print("Hello world !")
print("What is your name ?")
name=input()
print("Nice to meet you",name)
```

The right window, titled '*Python Shell*', shows the execution of the script. The output is as follows:

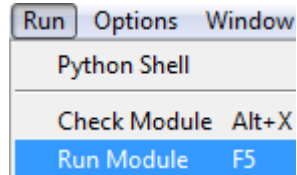
```
Python 2.7.3 (default, Apr 20 2012, 22:39:59)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
=====
>>>
Hello world !
What is your name ?
|
```

The status bars at the bottom of the windows indicate the current cursor position: 'Ln: 5 Col: 0' for the editor and 'Ln: 6 Col: 0' for the shell.

Un exemple de fonction en 2nde

Énoncé : Connaissant les coordonnées des points A, B, C et D, déterminer si les vecteurs \overrightarrow{AB} et \overrightarrow{CD} sont colinéaires.

Exécution du programme "vecteurs_colineaires.py" dans la fenêtre shell, en appelant la fonction `colineaire()`



```
>>> colineaire()  
abscisse de A 1  
ordonnée de A 2  
abscisse de B 3  
ordonnée de B 4  
abscisse de C 6  
ordonnée de C -1  
abscisse de D 2  
ordonnée de D -3  
Les vecteurs ne sont pas colinéaires
```

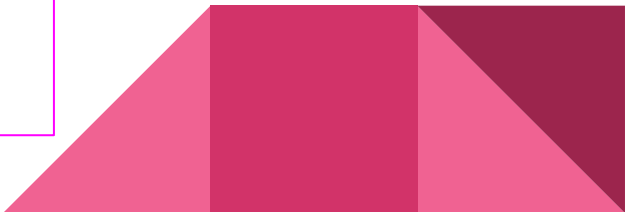
```
>>> colineaire()  
abscisse de A 1  
ordonnée de A 2  
abscisse de B 3  
ordonnée de B 4  
abscisse de C 0  
ordonnée de C -2  
abscisse de D 2  
ordonnée de D 0  
Les vecteurs sont colinéaires
```



```
*vecteurs_colineaires.py - C:\Users\Virginie GALLIEN\Documents\Mission_TICE_2016-2017\Python\ve...
File Edit Format Run Options Window Help

#Programme qui teste si deux vecteurs (AB) et (CD) sont colinéaires :
def colineaire():
# On rentre les coordonnées des points A, B, C, D
xA=float(input("abscisse de A "));
yA= float(input("ordonnée de A "));
xB= float(input("abscisse de B "));
yB= float(input("ordonnée de B "));
xC= float(input("abscisse de C "));
yC= float(input("ordonnée de C "));
xD= float(input("abscisse de D "));
yD= float(input("ordonnée de D "));
# On calcule les coordonnées des vecteurs (AB) et (CD).
m=xB-xA
n=yB-yA
p=xD-xC
q=yD-yC
if m*q-n*p==0 :
    print ("Les vecteurs sont colinéaires")
else :
    print ("Les vecteurs ne sont pas colinéaires")
```

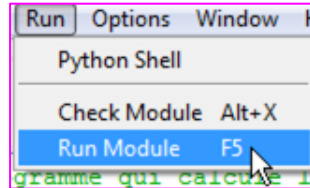
Capture
d'écran du
programme



Un exemple de programme en 1ère

Enoncé : Ecrire un programme qui calcule le discriminant delta puis en conséquence trouve les solutions éventuelles de l'équation.

Exécution du programme "racines_trinome.py" dans la fenêtre shell



```
Programme qui calcule les racines d'un polynome du second degré .  
Tel que  $Ax^2+Bx+C=0$   
A=2  
B=-2  
C=-1  
Delta= 12.0  
Deux solutions  
X1= -0.3660254037844386  
X2= 1.3660254037844386  
Fin du programme!  
Voulez-vous recommencer ?  
0. Oui  
1. Non  
0
```

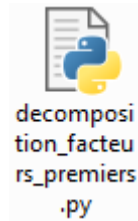
```
Programme qui calcule les racines d'un polynome du second degré .  
Tel que  $Ax^2+Bx+C=0$   
A=2  
B=1  
C=1  
Delta= -7.0  
Pas de solutions  
Fin du programme!
```

```
from math import *
# Permet d'importer toutes les fonctions du module "math"
# Ici, on a besoin de la fonction racine carré "sqrt(nombre)"

recommencer=0 # Permet de redémarrer le programme lorsqu'il est fini
while recommencer!=1:
    print("Programme qui calcule les racines d'un polynome du second degré .")
    print("Tel que  $Ax^2+Bx+C=0$ ")
    A=float(input('A=')) # On rentre la valeur de A
    B=float(input('B=')) # On rentre la valeur de B
    C=float(input('C=')) # On rentre la valeur de C
    delta=B*B-4*A*C # On calcule delta, le discriminant, en fonction de A,B et C
    print("Delta=",delta)# On affiche la valeur de delta)
    if delta <0:
        print("Pas de solutions")# Lorsque delta est négatif, il n'y a pas de solutions
    if delta ==0:
        print("Une solution") # Lorsque delta est égale à 0, il y a une solution X
        x=-B/2*A # Calcul de X
        print("X=",x) # On affiche la solution
    if delta >0:
        print("Deux solutions") # Lorsque delta est positif,il y a deux solutions, X1 et X2
        racine_carre_delta=sqrt(delta) # On calcule la racine carré de delta
        k=-B-racine_carre_delta # Variable qui va intervenir dans le calcul de X1
        l=-B+racine_carre_delta # Variable qui va intervenir dans le calcul de x2
        m=2*A # Variable qui va intervenir dans le calcul de X1 et X2
        x1=k/m # Calcul de X1
        x2=l/m # Calcul de X2
        print("X1=",x1) # On affiche la première solution
        print("X2=",x2) # on affiche la deuxième solution
    print("Fin du programme!")
    print("Voulez-vous recommencer ?")
    recommencer=input('0. Oui\n1. Non\n')
```

Capture
d'écran du
programme

Un exemple de programme en Tle S spé Maths



Énoncé : Ecrire un programme qui affiche la décomposition en facteurs premiers d'un nombre.

Rappel du théorème :

Tout nombre $n \geq 2$ se décompose en produit de facteurs premiers.

$$n = p_1^{r_1} p_2^{r_2} \dots p_k^{r_k}$$

où tous les nombres p_i sont des nombres premiers distincts avec $p_1 < p_2 < \dots < p_k$

et où les r_i désignent des exposants entiers ≥ 1 .

```
decomposition_facteurs_premiers.py - C:/Users/Virginie GALLIEN/Documents/Mission_TICE_2016-20...
File Edit Format Run Options Window Help
# liste des facteurs premiers
n = 1 # n'importe quelle valeur sauf 0
while n!=0:
    n=int(input("Entrez un nombre entier:"))
    if n<0:
        print("n est négatif, erreur, recommencez")
    elif n>0:
        print("n est positif, suite du programme")
        d = 2
        while n>1:
            while n%d==0: # % calcule le reste de division de n sur d
                n = n//d
                print("facteur trouvé:", d)
            d += 1 # d prend la valeur d+1
        print()
print("Fin du programme")
```

Capture
d'écran du
programme

```
Entrez un nombre entier:340
n est positif, suite du programme
facteur trouvé: 2
facteur trouvé: 2
facteur trouvé: 5
facteur trouvé: 17
```

Exécution du programme
"decomposition_facteurs_premiers.py"
dans la fenêtre shell

Webographie pour progresser avec Python

- ★ Pour débiter avec Python au lycée : <http://python.lycee.free.fr/>
- ★ Python pour débutant ou pour avancé : <http://apprendre-python.com/>
- ★ Apprendre Python en vidéos (par Denis Sanson) : Les pythonneries
http://www.dailymotion.com/playlist/x22t3u_universal_avs_apprendre-le-python/1#video=xbk6lj
- ★ [Des exercices de base avec Python : proposés par l'IREM de la Réunion.](#)
- ★ Des ressources sur le site Planète Maths :
 - [Initiation au langage Python](#)
 - [Créer et calculer avec des matrices sous Python. Module Numpy](#)
 - [.Algorithmique, découvrir les structures](#)