

# Cours - Thème: Python - Découverte de la programmation

Lien vers une interface de programmation en ligne

<https://repl.it/languages/python3>

Lien vers un site permettant d'installer une interface de programmation

<https://thonny.org/>



QrCode pour la prise en main de Thonny →

## Objectif

Créer des petits programmes en python

## Table des matières

[1 Rappel des commandes principales](#)

[2 Naissance et premiers pas](#)

[3 Répétition - la boucle bornée "for i in range\(\)"](#)

[4 Python, la tortue et le dessin](#)

## 1 Rappel des commandes principales

### Programmation en PYTHON



Créer un nouveau programme	On saisit le programme dans l'éditeur.	
	On va à la ligne après chaque instruction, mais on peut taper plusieurs instructions sur la même ligne en les séparant par un point-virgule.	
Importer un module	<code>from nom_du_module import*</code>	<code>from lycee import*</code> avec EduPython pour la bibliothèque <b>lycée</b>
Afficher $A$	<code>print(A)</code>	
Affecter à $B$ la valeur de $A$	<code>B=A</code>	
Écrire des commentaires	On écrit les commentaires sur une ligne, précédés de #	
Tester si $A = B$ / si $A \neq B$	<code>A==B</code>	<code>A!=B</code>
Tester si $A \geq B$ / si $A \leq B$	<code>A&gt;=B</code>	<code>A&lt;=B</code>
$A$ et $B$ / $A$ ou $B$	<code>A and B</code>	<code>A or B</code>
Si {condition $C$ } Alors {instructions $A$ } Sinon {instructions $B$ } Fin Si	<code>if {condition C}: {instructions A} else: {instructions B}</code>	Il n'y a pas d'instruction de fin : c'est l'indentation (le décalage vers la droite) qui indique les instructions faisant partie de la structure conditionnelle.
Pour $i$ variant de 1 à $n$ {instructions} Fin Pour	<code>for i in range(1,n+1): {instructions}</code>	<ul style="list-style-type: none"> <li>• L'instruction <code>for i in range(n)</code> fait parcourir à la variable <math>i</math> tous les entiers de 0 à <math>n - 1</math>.</li> <li>• L'instruction <code>for i in range(<math>n_0</math>,n)</code> fait parcourir à la variable <math>i</math> tous les entiers de <math>n_0</math> à <math>n - 1</math>.</li> <li>• L'instruction <code>for i in range(<math>n_0</math>,n,p)</code> fait parcourir à la variable <math>i</math> les entiers de <math>n_0</math> jusqu'à l'entier immédiatement inférieur ou égal à <math>n - 1</math> avec un pas de <math>p</math>.</li> </ul> <p>Comme pour <code>if</code>, on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.</p>
Tant que {condition $C$ } {instructions} Fin Tant que	<code>while {condition C}: {instructions}</code>	Comme pour <code>if</code> et <code>for</code> , on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.
Définition d'une fonction $f$	<code>def f(a,b,c,...): y= ... return(y)</code>	On peut aussi écrire : <code>return y</code> au lieu de <code>return(y)</code>
Racine carrée de $x$	<code>sqrt(x)</code> dans la bibliothèque <b>lycée</b> ou le module <b>math</b>	
$x$ à la puissance $n$	<code>x**n</code>	
$\pi$	<code>pi</code> dans la bibliothèque <b>lycée</b> ou le module <b>math</b>	
Reste de la division de $a$ par $b$	<code>a%b</code> avec la bibliothèque <b>lycée</b>	<code>fmod(a,b)</code> pour la calculatrice NumWorks
Nombre décimal aléatoire compris entre $a$ et $b$	<code>uniform(a,b)</code> avec la bibliothèque <b>lycée</b> ou le module <b>random</b>	
Entier aléatoire compris entre $a$ et $b$	<code>randint(a,b)</code> avec la bibliothèque <b>lycée</b> ou le module <b>random</b>	
Longueur d'une chaîne de caractères $C$ / Extraire un caractère de $C$	<code>len(C)</code>	<code>C[k]</code> renvoie le $(k + 1)$ <sup>e</sup> élément de $C$
Concaténation de deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"	
Booléens	<code>True</code> (Vrai)	<code>False</code> (Faux)
Exécuter un programme	Onglet <b>Exécuter</b>	EduPython : cliquer sur ▶

## 2 Naissance et premiers pas

Votre objectif est d'écrire des petits programmes répondant aux objectifs suivants

- **Objectif 1:**

Afficher le texte "Hello World"

Votre code:



Vos remarques:



- **Objectif 2:**

Dans votre script, affecter la chaîne de caractère "Bonjour" à la variable a  
Faire de même pour le prénom de votre voisin et la déclarer dans la variable b  
Afficher la somme de a+b

Votre code:



Vos remarques:



- **Objectif 3:**

Afficher sur 2 lignes :

\*  
\*\*

Votre code:



# Thème Transversal - Python - Programmation - TD découverte

Vos remarques:



- **Objectif 4:**

Tester le code suivant

```
etoile=""  
print(etoile)  
etoile=etoile+"*"  
print(etoile)
```

Expliquez chaque ligne:



- **Objectif 5:**

Tester le code suivant

```
etoile=""  
print(etoile)  
etoile=etoile*3+" "+etoile*3  
print(etoile)
```

Expliquez chaque ligne et dessinez le résultat:



## 3 Répétition - la boucle bornée "for i in range()"

- **Objectif 6:**

Tester le code suivant

```
etoile=""  
for i in range (10):  
    etoile=etoile+"*"  
    print(etoile)
```

Expliquez chaque ligne du programme du dessus



**Notez le décalage des lignes dans la boucle!**



Voici ce que j'obtiens

```
>>> %Run etoile.py
```

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

Que valent les valeurs de i successives? Comment en être certain? `print(i,etoile)`

# Thème Transversal - Python - Programmation - TD découverte

- **Objectif 7: De la suite dans les idées**

1-Créer un programme qui affecte 2 à la variable a puis affiche la valeur de la variable a...

2-puis qui ajoute à nouveau 2 à la variable a puis affiche la valeur de a...

3-puis...

...

**1000**-le tout 1000 fois

Votre code: (indiquez le nombre d'essais que vous avez fait)

Nbres de tentatives =



- **Objectif 8: De la suite dans les idées**

1-Créer un programme qui affecte 2 à la variable b puis affiche la valeur de la variable b...

2-puis qui multiplie par 2 la valeur de b et met le résultat dans b puis affiche la valeur de b...

3-puis...

...

**10**-le tout 10 fois

Mon résultat doit être du genre "afficher la valeur de i dans la boucle puis la valeur de b" :

```
>>> %Run somme.py
```

```
1 - 2
```

```
2 - 4
```

```
3 - 8
```

```
4 - 16
```

```
5 - 32
```

```
6 - 64
```

```
7 - 128
```

```
8 - 256
```

```
9 - 512
```

```
10 - 1024
```

Votre code: (indiquez le nombre d'essais que vous avez fait)

Nbres de tentatives =



# Thème Transversal - Python - Programmation - TD découverte

- **Objectif 9: Python est très fort avec les entiers**

Appliquez le même principe que pour l'objectif 8 mais en bouclant 200 fois!!!

Pouvez vous vraiment lire ceci ;))))

200 - 1606938044258990275541962092341162602522202993782792835301376

## 4 Python, la tortue et le dessin

Tester le script suivant

```
from turtle import *  
forward(120)  
left(90)  
forward(120)  
left(90)  
forward(120)  
left(90)  
forward(120)  
left(90)
```

Reproduire le dessin

Expliquez les 4 premières lignes du programme du dessus



- **Objectif 10: Le Python est plein d'astuce**

Simplifiez le programme pour reproduire le même dessin avec un boucle for et 4 itérations!

Votre code: (indiquez le nombre d'essais que vous avez fait)

Nbres de tentatives =

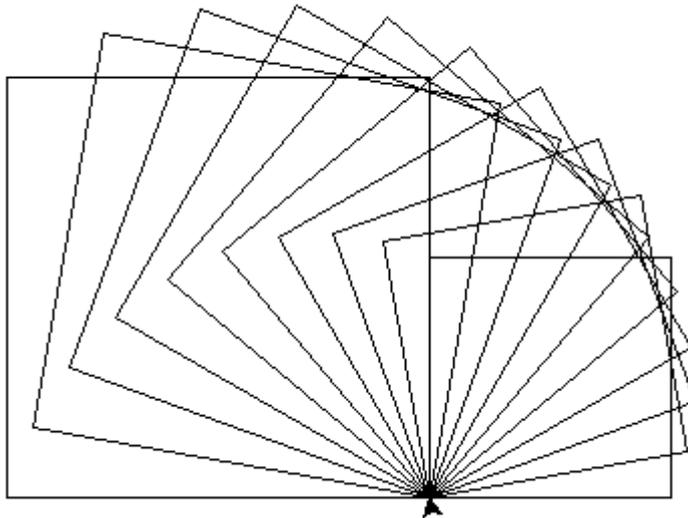


# Thème Transversal - Python - Programmation - TD découverte

- **Challenge final**

Soit le script suivant:

```
from turtle import *  
for j in range (10):  
  
    for i in range(4):  
        forward(120+j*10)  
        left(90)  
  
    left(10)
```



Expliquez à vos parents l'esprit du programme du dessus ;)))

