

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
2- Naissance et premiers pas.....	3
3- Répétition - la boucle bornée "for i in range()".....	5
4- Répétition - la boucle non bornée "while i<=50 :".	7
5- Python et fonctions.....	7
6- Python, la tortue et le dessin.....	9

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 lycée
Afficher <i>A</i>	<code>print(A)</code>	
Affecter à <i>B</i> la valeur de <i>A</i>	<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>=B</code>	<code>A<=B</code>
<i>A</i> et <i>B</i> / <i>A</i> ou <i>B</i>	<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>i</i> variant de 1 à <i>n</i> {instructions} Fin Pour	<code>for i in range(1,n+1): {instructions}</code>	<ul style="list-style-type: none"> L'instruction <code>for i in range(n)</code> fait parcourir à la variable <i>i</i> tous les entiers de 0 à <i>n</i> - 1. L'instruction <code>for i in range(n₀,n)</code> fait parcourir à la variable <i>i</i> tous les entiers de <i>n</i>₀ à <i>n</i> - 1. L'instruction <code>for i in range(n₀,n,p)</code> fait parcourir à la variable <i>i</i> les entiers de <i>n</i>₀ jusqu'à l'entier immédiatement inférieur ou égal à <i>n</i> - 1 avec un pas de <i>p</i>. <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 <i>f</i>	<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 <i>x</i>	<code>sqrt(x)</code> dans la bibliothèque lycée ou le module math	
<i>x</i> à la puissance <i>n</i>	<code>x**n</code>	
π	<code>pi</code> dans la bibliothèque lycée ou le module math	
Reste de la division de <i>a</i> par <i>b</i>	<code>a%b</code> avec la bibliothèque lycée	<code>fmod(a,b)</code> pour la calculatrice NumWorks
Nombre décimal aléatoire compris entre <i>a</i> et <i>b</i>	<code>uniform(a,b)</code> avec la bibliothèque lycée ou le module random	
Entier aléatoire compris entre <i>a</i> et <i>b</i>	<code>randint(a,b)</code> avec la bibliothèque lycée ou le module random	
Longueur d'une chaîne de caractères <i>C</i> / Extraire un caractère de <i>C</i>	<code>len(C)</code>	<code>C[k]</code> renvoie le (<i>k</i> + 1) ^e élément de <i>C</i>
Concaténation de deux chaînes de caractères	<code>"abcd" + "efg"</code> donne <code>"abcdefg"</code>	
Booléens	<code>True</code> (Vrai)	<code>False</code> (Faux)
Exécuter un programme	Onglet Exécuter	EduPython : cliquer sur ►

Invite de saisie

```
texte=input("Saisir votre texte :")
entier=int(input("Saisir un entier :"))
decimal=float(input("Saisir un décimal – séparateur . :"))
```

2- Naissance et premiers pas

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

- **Objectif :**

Afficher le texte "Hello World"

Votre code:

```
print("hello world")
```

Vos remarques:

- pour exécuter le programme il faut avant l'enregistrer
- exécution programme voir outil > ou F6
- commande print pour afficher dans console=shel
- menu affichage variable pour fenêtre variable

- **Objectif :**

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:

```
a="bonjour "
```

```
b="pierre"
```

```
print(a+b)
```

Vos remarques:

- mode des bugs sympa
- a="ggg" affecte à la variable a la valeur texte "

- **Objectif :**

Afficher sur 2 lignes :

```
*
```

```
**
```

Votre code:

Thème Transversal - Python - Programmation - TD découverte

Vos remarques:

- 1 print = une ligne

- **Objectif :**

Tester le code suivant

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

Expliquez chaque ligne:

- **Objectif :**

Tester le code suivant :

```
texte=input("Saisir une chaîne de caractère : ")  
print(texte)  
print(texte+texte)
```

Relancer le programme en saisissant lors de l'invite de commande « 2 »

Modifier la première ligne de votre code :

```
nbre=int(input("Saisir une chaîne de caractère : "))
```

Vos remarques:

Objectif :

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 :**

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)`

i commence par la valeur 0 puis 1 puis... terminera à 9 donc de 0 à 9 il y a bien 10 valeurs

Thème Transversal - Python - Programmation - TD découverte

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

1-Créer un programme qui propose à l'utilisateur de saisir un nombre (par exemple 2) dans a

2-puis qui ajoute à nouveau ce même nombre (ici 2) dans a

3-puis...

...

1000-le tout 1000 fois

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

Nbres de tentatives =

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

1-Créer un programme qui propose à l'utilisateur de saisir un nombre (par exemple 2) dans b

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

3-puis...

...

10-le tout 10 fois

Mon résultat doit être du genre:

```
>>> %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 : Python est très fort avec les entiers**

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

Pouvez vous vraiment lire ceci ;))))

200 – 1606938044258990275541962092341162602522202993782792835301376

4- Répétition - la boucle non bornée “while i<=50 :”

Symbole de comparaison :

Ce sont les mêmes que ceux utilisés avec les instructions conditionnelles :

== : ce symbole teste si deux valeurs sont égales.

!= : ce symbole teste si deux valeurs sont différentes.

> : ce symbole teste si la première valeur est strictement supérieure à la deuxième.

>= : ce symbole teste si la première valeur est supérieure ou égale à la deuxième.

< : ce symbole teste si la première valeur est strictement inférieure à la deuxième.

<= : ce symbole teste si la première valeur est inférieure ou égale à la deuxième.

Attention, une boucle while peut ne jamais s'arrêter => Bouton STOP dans Thonny

- **Objectif :**

Tester le code suivant

```
i=0
b=2
while b<=10000000000:
    print(i+1,"-",b)
    i=i+1
    b=2*b
```

Expliquez :

Modifiez ce code pour n'afficher dans le shell que la valeur de 2^{20} !

Votre code:

5- Python et fonctions

- **Objectif :**

Tester le code suivant

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

Thème Transversal - Python - Programmation - TD découverte

Il ne se passe rien;) mais regardez vos variables !

Dans le shell, saisir :

```
etoile(3)
```

Expliquez selon vous ce qui se passe

- **Objectif :**

Tester le code suivant

```
def etoile(n):  
    etoile=""  
    for i in range (n):  
        etoile=etoile+"*"  
    print(etoile)  
  
print(etoile(5))
```

Expliquez selon vous ce qui se passe

- **Objectif : Synthèse**

Créez un programme qui réponde aux consignes suivantes :

- Votre programme commencera par une fonction qui calculera les valeurs de $2x^2 - 5x + 35$ en fonction de la variable x
- Après la déclaration de cette fonction, vous demanderez à l'utilisateur de saisir une valeur entière quelconque
- Cette valeur sera utilisée pour calculer la valeur de votre fonction, votre programme devra afficher le résultat !

Votre code:

Vos remarques:

6- Python, la tortue et le dessin

Tester le script suivant

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

Thème Transversal - Python - Programmation - TD découverte

```
forward(120)
```

```
left(90)
```

```
forward(120)
```

```
left(90)
```

Reproduire le dessin

Expliquez les 4 premières lignes du programme du dessus

- **Objectif : 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 ;)))

- **Super Challenge Final**

Voir l'Activité 3 de votre manuel