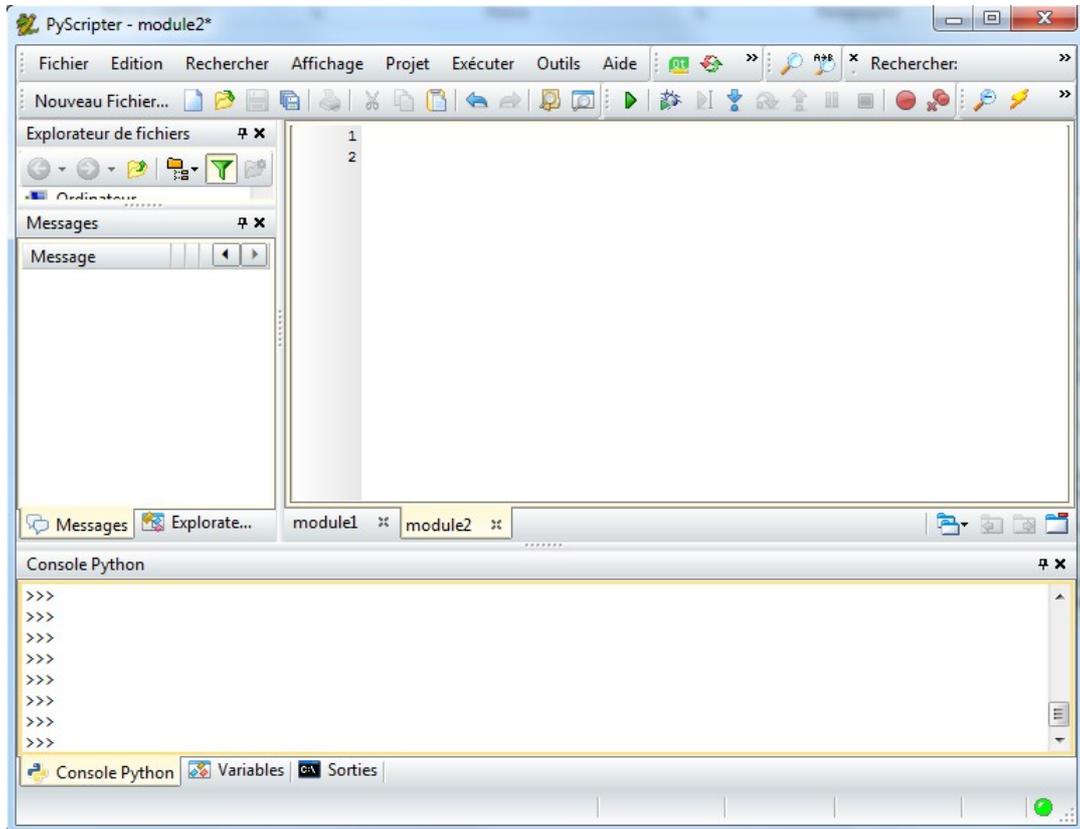


FORMATION PYTHON 2017 – 2018

PRISE EN MAIN – LES BASES

Programmation sous EDUPYTHON, distribution clé en main et portable pour programmer avec les élèves sous Python 3, EduPython peut être installé sur un ordinateur classiquement ou sur clé USB.

<https://edupython.tuxfamily.org/>



1) Dans la console :

On peut saisir directement des instructions Python qui s'exécutent automatiquement après appui sur la touche <Entrée>
Exemples :

```
>>> 6+8
14
>>> 3*8
24

>>> _+3
27
>>> 20/3
6.666666666666667

>>> 20//3
6
>>> 20%3
2
>>> 3**2
9
... |
```

Remarque :

Il est parfois nécessaire de recourir à l'utilisation de modules. Un module est une sorte de bibliothèque donnant l'accès à certaines fonctions déjà programmées, comme par exemple, le triage de nombre aléatoire ou encore la manipulation des racines carrées.

Quelques modules à importer : A écrire en haut du script ou dans l'invite de commande avant de saisir les instructions

- from math import* :
Exemples : sqrt(2), floor(3.1), abs(-5),...
- from random import*
Exemples : randint(1,6), random(),...
- from turtle import*
Permet de réaliser des dessins, un peu dans le même esprit que scratch
- from time import*
Permet de mettre des paramètres de temporalité

Pour saisir au clavier des variables, il faut faire attention à expliciter le type de variable qu'on souhaite saisir. Par défaut, ce que saisit l'utilisateur est une chaîne de caractères.

```
>>> message = input(« Veuillez saisir votre message ») # saisir au clavier une chaîne de caractère
>>> entier = int(input(« Veuillez saisir un nombre entier »)) # saisir au clavier un nombre entier
>>> reel = float(input(« Veuillez saisir un nombre réel »)) # saisir au clavier un nombre réel flottant
```

Pour connaître le type d'une variable, on peut utiliser la fonction « type ». (cf Exercice 3)

Exercice 1 :

Saisir les instructions suivantes dans l'exécuteur de commande Python afin de comprendre leur rôle :
(On saisira `from math import*` avant de saisir les instructions)

Remarque : cette liste est loin d'être exhaustive

<pre>>>> 1+1 >>> 7*5 >>> 4/2 >>> 7/2 >>> 7//2</pre>	<pre>>>> round(4/3,2) >>> 10**2 >>> 23%5 >>> sqrt(2) >>> a = "Bon"</pre>	<pre>>>> b = "jour" >>> c = a+b >>> print(c) >>> 5==3 >>> 10!=5</pre>
--	---	--

2) Dans l'éditeur :

Exercice 2 :

Après avoir saisi les lignes suivantes, quelles sont les valeurs de a et b ?

(Utiliser `print(a)` puis `print(b)` pour vérifier)

Quelle suite d'instructions faudrait-il saisir pour échanger le contenu de deux variables ?

```
>>> a,b=4,5
>>> a=b
>>> b=a
```

L'éditeur sert à taper des programmes ou scripts en Python que l'on pourra sauvegarder dans des fichiers dont l'extension est `.py`

Exercice 3 :

Saisir le script suivant :

```
2 a=2 #entier
3 b=15.0 #flottant
4 c="bonjour" #chaîne de caractère
5 print(a)
6 b=b+3.2
7 print(b)
8 print(c)
9 type(42) #donne le type de l'expression
10 print(int(8.5)+1) #int:partie entière
11 print(float(7//2)) #convertir une expression d'un type à l'autre
12 print(5%2==0) #booléen (= : affectation)( == :test d'égalité)
13 l=[1,2,3,4] #liste (pas d'enseignement spécifique en seconde)
14 print(l)
```

Vous observerez la coloration syntaxique, l'ajout automatique de parenthèses fermantes. Exécuter ce programme.

Vérifiez que vous obtenez :

```
Console Python
>>>
2
18.2
bonjour
9
3.0
False
[1, 2, 3, 4]
>>>
```

Les variables et les fonctions

La définition d'une fonction commence par le mot clef **def**. Une fonction peut avoir un nombre quelconque d'arguments (ou paramètres), elle peut renvoyer une ou plusieurs valeurs à l'aide de l'instruction **return** qui stoppe l'exécution de la fonction.

Il n'est pas nécessaire de déclarer les variables en début de programme.

Pour chaque programme proposé dans ce document, il est vivement conseillé de le copier ou de le taper dans le script Python et de l'exécuter. N'hésitez pas à modifier les exemples proposés.

Remarque importante : En Python, pour écrire à l'intérieur d'une fonction, un test, une boucle, on met « : » et on passe à la ligne. C'est l'indentation, c'est-à-dire les décalages qui conditionnent l'appartenance d'une instruction à une fonction test, une boucle. Cette indentation se fait automatiquement.

Exemple 1 :

Script	Console
<pre>2 def fonctionaffine(a,b,x): 3 image=a*x+b 4 return image 5 print(fonctionaffine(2,1,3)) 6 7 def bonjour(nom,age): 8 phrase="bonjour"+nom+",vous avez "+str(age)+" ans" 9 #age est entier donc str transforme en chaine de caractère 10 return phrase 11 print(bonjour(' toto',15))</pre>	<pre>>>> 7 bonjour toto,vous avez 15 ans >>></pre>

Exemple 2 :

Script	Console
<pre>0 a=3 1 b=2 2 c=a+b 3 b=b+c 4 a=a+c 5 print(a,b,c)</pre>	<pre>>>> 8 7 5</pre>

Exemple 3 :

Script	Console
<pre>8 def mystere(A,B) : 9 A= A+B 10 B=A-B 11 A=A-B 12 return(A,B) 13</pre>	<pre>>>> mystere(5,8) (8, 5)</pre>

Échauffement

[QCM sur les variables](#)

[QCM sur les fonctions](#)

Exercice 4 :

Écrire une fonction permettant de convertir des secondes en heures/minutes/secondes.

Exercice 5 :

Écrire un programme permettant de calculer la distance entre deux points définis par leur coordonnées cartésiennes.

Les structures conditionnelles

Exemples : Tester les programmes suivants.

Exemple 4 :

Script <pre><code><i>#exemple 4 :</i> def etat(t) : if t<=0: return("C'est de la glace") if (t>0) and (t<100): return("C'est de l'eau liquide") if t>=100: return("C'est de la vapeur")</code></pre>	Console <pre><code>>>> etat(50) "C'est de l'eau liquide" >>> etat(-10) "C'est de la glace" >>> etat(200) "C'est de la vapeur"</code></pre>
--	--

Exemple 5 :

Script <pre><code><i># EXEMPLE 5 congruences #</i> print("EXEMPLE 5") a=41207 if a%11==3: print(a," est de la forme 11n+3") else: print(a," n'est pas de la forme 11n+3") <i>#L'utilisation de cette dernière structure est facultative</i> a=28 if a%3==0: print(a," est de la forme 3n") elif a%3==1: print(a," est de la forme 3n+1") else: print(a,"est de la forme 3n+2")</code></pre>	Console <pre><code>>>> 28 est de la forme 3n+2 41207 n'est pas de la forme 11n+3 >>></code></pre>
--	--

Remarque : on peut en quelque sorte considérer que les « : » font office de « then ».

Échauffement

[QCM sur les structures conditionnelles](#)

Exercice 6 :

Écrire une fonction qui renvoie la valeur absolue d'un nombre pris comme paramètre.

Exercice 7 :

Écrire une fonction qui définit une fonction affine par morceaux.

Exercice 8 :

Écrire une fonction prenant en paramètres trois nombres et qui les affiche dans l'ordre croissant.

Exemples : Tester les programmes suivants.

Exemple 6 :

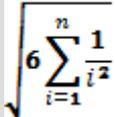
Script	Console
<pre># EXEMPLE 6 # print("EXEMPLE 6") print("exemple1") for i in range(5): # i prend les valeurs 0 à 4 print(i) print("exemple2") S=0 for i in range (1,4): # i prend les valeurs 1 à 3 S=S+i print(S)</pre>	<pre>exemple1 0 1 2 3 4 exemple2 1 3 6</pre>

Exemple 7 :

Script	Console
<pre># EXEMPLE 7 # print("EXEMPLE 7") from random import* for i in range(1,7): # 6 lancers càd i entier de [1;7[d=randint(1,6) #nombre entier aléatoire entre 1 et 6 print(d) if d==1: print("gagné!") elif d==6: print("gagné !") else: print("perdu")</pre>	<pre>>>> 3 perdu 4 perdu 1 gagné! 2 perdu 3 perdu 5 perdu >>></pre>

Exercice 9 :
Écrire une fonction prenant en paramètre un nombre entier naturel et qui affiche la table de multiplication de ce nombre.

Exercice 10 : (*Eduscol ressources probas stats*)
Une puce se déplace sur un axe gradué, elle part de l'origine et se déplace de manière aléatoire et équiprobable vers la droite ou la gauche. Quelle est sa position après 30 sauts ?

Exercice 11 :
Écrire une fonction prenant en paramètre un nombre entier naturel n non nul et qui calcule 

Exercice 12 :
Écrire une fonction prenant en paramètre un nombre entier naturel et qui effectue un compte à rebours à partir de ce nombre.

Exemples : Tester les programmes suivants.

Exemple 8 :

Script	Console
<pre> 33 print("exemple1") 34 i = 1 35 while i <=6 : 36 print(i) 37 i = i+1 38 39 print("exemple2") 40 i=7 41 while i!=20 : 42 i=i+3 </pre>	<pre> >>> exemple1 1 2 3 4 5 6 exemple2 # Ce programme boucle à l'infini # car la condition i!=20 est toujours vérifiée. </pre>

Exemple 9 :

Script	Console
<pre> print("EXEMPLE 9") liste=[1,3,5,7,9,11]# 1 a pour rang 0 dans la liste i=0 while i<=5: #tant que i est inférieur ou égal à 5 print(liste[i]) # affiche le i ème terme de la liste i=i+1 print(liste) #affiche la liste print (liste[0]) # affiche le premier terme de la liste print(len(liste)) # affiche la longueur de la liste </pre>	<pre> >>> 1 3 5 7 9 11 [1, 3, 5, 7, 9, 11] 1 6 >>> </pre>

Échauffement

[QCM sur les boucles](#)

Exercice 13 :

Écrire un programme qui simule des lancers de dé jusqu'à ce que l'on obtienne la valeur 6.

Exercice 14 :

Écrire un programme pour estimer le nombre de lancers nécessaires pour obtenir toutes les faces d'un dé cubique.

Exercice 15 :

Écrire un programme qui demande à l'utilisateur de saisir un mot de passe pour se connecter. Le modifier pour ne laisser à l'utilisateur que trois essais.

Ce module permet d'ajouter un interface graphique et de réaliser des figures.

Le module Turtle

En sélectionnant « nouveau fichier » on peut sélectionner le modèle tortue directement, on obtient alors :

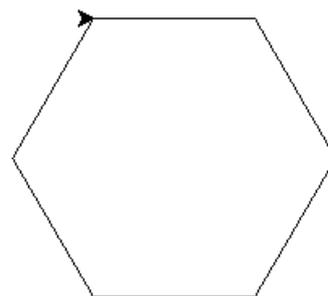
```
2 from lycee import *
3 import turtle as tortue
4
5 |
6
7 tortue.mainloop()
```

Exemple : Tester le programme suivant.

Exemple 10 :

```
1 from lycee import*
2 import turtle as tortue
3 tortue.fd(50)
4 tortue.rt(90)
5 tortue.fd(50)
6 tortue.reset()
7 tortue.speed(0)
8 for i in range (6):
9     tortue.fd(100)
10    tortue.ret(360/6)
11 tortue.exitonclick()
12 # Quitter la fenêtre graphique en cliquant
13
```

Voici ce qu'on obtient :



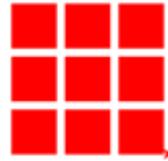
Dans le cas précédent, notre figure s'appelle tortue et on utilise les différentes commandes avec le nom de la figure : tortue.fd, tortue.lt, etc...

Quelques commandes utiles avec turtle

Fonction	Effet
fd(n)	avance de n
bk(n)	recule de n
rt(n)	tourne à droite de n degrés
lt(n)	tourne à gauche de n degrés
clear()	efface l'écran
penup()	lève le stylo
pendown()	baisse le stylo
reset()	efface l'écran, remet la tortue au centre et réinitialise ses paramètres
showturtle()	montre la tortue
hideturtle()	cache la tortue
speed(n)	Change la vitesse de 1(lent) à 10 (rapide). La valeur spéciale 0 est la plus rapide.
tracer(n,d)	
update()	Force l'affichage des graphismes en attente
bye()	Referme la fenêtre
setup(w,h)	Ouvre une fenêtre de taille wxh

Le module Turtle peut être intéressant en début d'année car il permet de retravailler les notions d'algorithmique vues en collège avec Scratch et d'introduire par exemple la notion de fonction en informatique

Exemple 11 : la figure, le code Python, le code Scratch



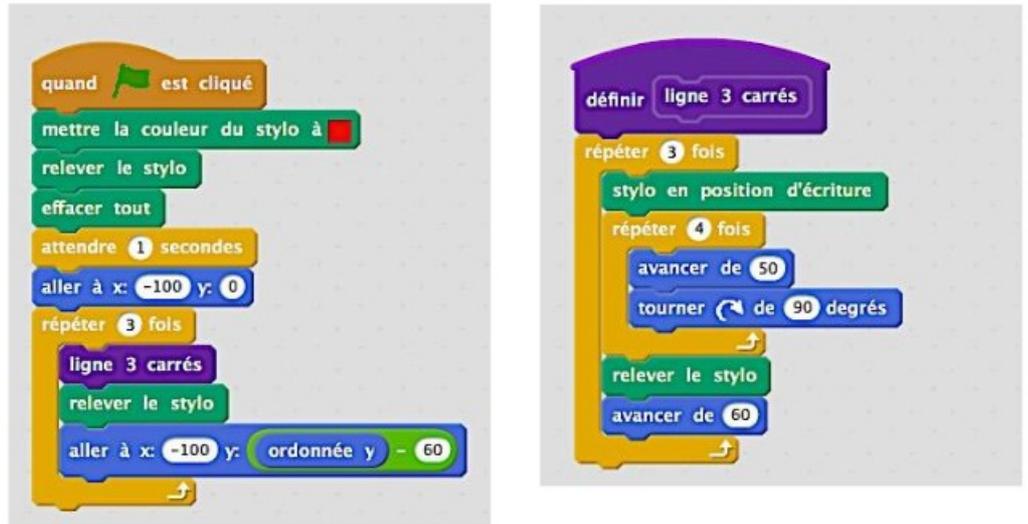
Python

```
from lycee import *
from turtle import*
def carre():
    color("red")
    begin_fill()
    for i in range(4):
        down()
        fd(50)
        lt(90)
    end_fill()

def ligne():
    for i in range(3):
        carre()
        up()
        fd(60)

x=0
y=0
for i in range (3):
    goto(x,y)
    ligne()
    x=0
    y=y-60
exitonclick()
```

Scratch



Exercice 16 :

Réaliser des dessins semblables aux figures suivantes

