

Seq. 0 - Notions transversales de programmation python

Nom :

Prénom :

/10

Implication séance 1	--	-	=	+	++
Implication séance 2					
Implication séance 3					
Implication séance 4					
Implication séance 5					
Implication séance 6					

0 Première approche de la pensée algorithmique

Pour s'exercer à la pensée algorithmique et apprendre à travailler la programmation avec Python, nous allons utiliser le site France IOI :

<http://www.france-ioi.org/>

0.1 S'inscrire :

A faire vous même 1.

- Allez sur France IOI
- Créez votre identifiant et votre mot de passe :
 - Id : prenomclassecordeliers (ex : jean2gcordeliers, tout en minuscules)
 - Mot de passe : nom_JJMMAAAA (nom de famille_date de naissance)
- Notez votre identifiant et votre mot de passe :
 - Id :
 - mdp :
- Connectez-vous
- Cliquez sur Mon profil (en haut à gauche)
- Décochez Afficher seulement les champs requis ou recommandés et de nouveaux champs apparaissent
- Complétez les champs Prénom et Nom de famille

0.2 S'enregistrer dans le groupe des Cordeliers

A faire vous même 2.

- Dans l'onglet Enseigner, cliquer sur Groupes et classes
- Pour rattacher votre compte à votre groupe classe :
 - Cherchez le nom de votre groupe : Cordeliers2C_2022
 - Cliquez sur le nom du groupe
 - Mot de passe : 2C2022
 - Pour une autre classe il suffit de changer la lettre.

Ainsi l'enseignant peut suivre ta progression.

0.3 Progresser :



A faire vous même 3.

- Dans l'onglet Progresser, cliquez sur Cours et Problèmes
- Choisissez le langage Python (en haut à droite)
- Puis le parcours lycée
- C' est parti !

Attention, effectue les exercices dans l'ordre !

1 Le langage informatique Python



1.1 Lancer un programme pour utiliser Python

Il est possible de travailler en ligne à partir d'un site web :

- basthon : <https://console.basthon.fr/>
- repl.it : <https://repl.it/languages/Python3>
- Sofuspy : <https://irem.univ-reunion.fr/blockly/plurialgo/blockly/extensions/sofuspy/run.html>

Pratique pour débiter sans rien installer mais cela montre vite ses limites (impossible d'installer des modules supplémentaires, plus difficile de créer/lire/modifier des fichiers, ...).

A faire vous même 4.

- Allez sur Basthon
- Sur la page web, il y a deux cases qui s'ouvrent
- Cherchez celle ou il y a le signe : >>>
- Cliquez juste à côté de ce sigle
- Écrivez ceci et tapez sur Entrée :

```
>>> print("Hello world !")
```
- Que se passe-t-il ?
- Vous venez de saisir votre première commande !

IMPORTANT : Dans toute la documentation python, vous allez retrouver ce signe >>> Cela correspond à l'invite de commande python.

A faire vous même 5.

- Allez dans le menu Démarrer
- Cherchez et lancez Edupython ou Spyder
- Un fenêtre s'ouvre. Elle est partagée en 3 sous-fenêtres
- Cherchez celle ou il y a le signe : >>>
- Cliquez juste à côté de ce sigle
- Écrivez ceci et tapez sur Entrée :

```
>>> print("Hello world !")
```
- Que se passe-t-il ? Vous venez de saisir votre première commande !
- Montrez au professeur



1.2 Deux commandes de base Python



1.2.1 La commande print()

Cette commande permet d'afficher (imprimer) un message.

Attention ! Ce message doit être mis entre guillemets : " ... "

A faire vous même 6.



- Écrivez ceci :


```
>>>print(Voici un premier message)
```
- Que se passe-t-il ? Quel est le problème ? Comment le résoudre ? Écrivez la bonne commande


```
.....
```

```
.....
```

```
.....
```
- Écrivez ceci :


```
>>>print("Affichage du deuxième message")
```
- Tapez la commande pour afficher : Python c'est super !
- Essayez les touches flèches haut / flèches bas. Que se passe-t-il ?


```
.....
```
- Imprimez les messages suivants :
 - Vive la rentrée !
 - SNT c' est génial.
 - Les Cordeliers : Meilleur lycée de France !!!
- Appelez le professeur.

1.2.2 La commande input()



Cette commande permet de demander à un « utilisateur » de saisir un texte.

Attention ! A partir de maintenant, même si vous n'êtes pas schizophrène, vous devez jouer 2 rôles, le rôle du développeur python et le rôle de l'utilisateur python.

A faire vous même 7.



- Écrivez ceci :


```
>>>input("Comment t'appelle-tu ?")
```
- Que se passe-t-il ? Que faire ?


```
.....
```

```
.....
```

```
.....
```
- Écrivez ceci :


```
>>>input("Quel âge as-tu ?")
```
- Tapez la commande pour afficher la question : Quelle matière préfère-tu ?
- Essayez les touches flèches haut / flèches bas. Que se passe-t-il ?


```
.....
```
- Appelez le professeur.

2 Notion de variable en Python



2.1 Affectation de variable

Comme dans Scratch, Python manipule des variables (étiquettes dans lesquelles on stocke des valeurs).

Pas besoin de les créer ! Elles se créent automatiquement à la 1ère affectation de variable.

A faire vous même 8.



- Écrivez ceci :


```
>>>toto
```
- Que se passe-t-il ?

- Écrivez ceci :

```
>>>toto=2005
>>>toto
```

- La variable toto est créé et contient la valeur 2005

- Écrivez ceci :

```
>>>toto=1998
>>>toto
```

- La variable toto contient une autre valeur
- Créez une variable date_coupe_du_monde en lui affectant 2018
- Créez une variable poids_de_mon_chat en lui affectant 6.125
- Créez une variable nom_de_mon_chien en lui affectant Médor (Attention ! Il faut encadrer la valeur avec des guillemets !)
- Choisissez un nom de variable et affectez-lui une valeur
- En vous aidant de A faire vous-même 6, trouvez une commande Python qui affiche/imprime une variable déjà créé
- Appelez le professeur.

2.2 Variable et commande input()



Nous pouvons combiner l' affectation de variable et la commande input.

A faire vous même 9.

- Écrivez ceci :

```
>>>prenom=input("Quel est votre prénom")
>>>prenom
```

- Que se passe-t-il ?

- Écrivez une commande permettant de demander la couleur préférée et de la mettre dans une variable couleur
- Écrivez une commande permettant de demander l'âge et de la mettre dans une variable age
- Testez les commandes suivantes :

```
>>>lycee="Les Cordeliers"
>>>print("Mon lycée c'est ", lycee)
```
- De la même façon, trouvez une commande qui permette d'afficher une phrase (que vous inventerez) qui intègre les 3 variables (prenom, couleur et age)

- Appelez le professeur.

2.3 Réviser l' affectation de variable

A faire vous même 10.

- Allez sur python-lycée – cours 1 : <https://www.python-lycee.com/parcours-apprentissage-pl1>
- Regardez la vidéo et reproduisez les commandes sur la console intégrée à la page
- Faire l' exercice 1
- Faire l' exercice 2
- Faire l' exercice 3
- Faire l' exercice 4 (pour les costauds)
- Faire l' exercice 5 (pour les costauds)

3 Les séquences - for ... in range():



Quand on veut répéter une commande un certain nombre de fois (3 fois 7 fois, 632 fois, ...), on utilise une séquence.

A faire vous même 11.

- Écrivez ceci sans vous trompez :

```
>>>for i in range(3) :  
... print("Coucou")
```

- Que se passe-t-il ?

.....

- Écrivez ceci sans vous trompez :

```
>>>for i in range(5) :  
... print("A la ", i)
```

...

- Que se passe-t-il ?

.....

- Appelez le professeur.

4 Mes premiers scripts



La fenêtre avec les >>> s'appelle la *console*.

La console est intéressante quand on veut tester des commandes simples.

Dès qu'il faut enchaîner plusieurs commandes, il vaut mieux changer de sous-fenêtre et aller à droite pour écrire un *script*.

A faire vous même 12.



- Dans la fenêtre script écrivez une deuxième fois :

```
# -*- coding: UTF-8 -*-  
print("Début du script")  
for i in range(5) :  
    print("A la ", i)  
print("Fin du script")
```

- ATTENTION ! METTEZ 4 ESPACES AU DEBUT DE LA 4e LIGNE !

- Enregistrez ce script quelque part
- Exécutez-le (bouton avec une flèche verte)
- Retrouvez-le avec votre explorateur de fichiers
- Quel extension ont les scripts Python ?

.....

- A noter : La 1ère ligne permet simplement d'utiliser les caractères accentués (é, è, ë, ê, à, ...). Elle est à ajouter à chaque début de script.
- Appelez le professeur.

A faire vous même 13.



- A noter : la fonction int() convertit une donnée saisie en entier :

```
>>>int('21')  
21
```

- Écrivez un script Python qui réalise les choses suivantes :

- Afficher un message : Début de programme
- Demander à un utilisateur : Jusqu'à combien dois-je compter ?

- Convertir la réponse en nombre entier
- Afficher le comptage (Et de 0. Et de 1. Et de 2 ...)
- Afficher un message : Fin de programme

5 Calculer avec Python

Il est possible de calculer avec Python. C'est ce que fait la calculatrice Numworks.

- Les classiques : addition, soustraction, multiplication, division

```
>>>2+3
5
>>>11-2
9
>>>46*5
230
```

- Quotient seuil ou quotient entier de division euclidienne :

```
>>>27 // 7
3
```

- Modulo ou reste de division euclidienne :

```
>>>27 % 7
6
```

- Puissance :

```
>>>5 ** 3
125
```

A faire vous même 14.

Effectuez les calculs suivants

- $\frac{4^3+65*13}{27-7}$
- $3x^3-2019x+\frac{50}{3}$ avec $x=2,7$

A faire vous même 15.

- Écrivez un programme qui affiche les 20 premiers termes de la table de multiplication par 7.

A faire vous même 16.

- Écrivez un programme qui calcule le volume d'un parallépipède rectangle dont on demande au départ la largeur, la hauteur et la profondeur.

A faire vous même 17. Pour les costauds

Écrivez un programme qui convertisse un nombre entier de secondes fourni au départ, en un nombre d'années, de mois, de jours, de minutes et de secondes. (Utilisez l'opérateur modulo : %).

6 Les fonctions (première approche)

6.1 Les fonctions mathématiques

Les fonctions python sont définies grâce au mot clé *def*.

Chaque fonction prend des paramètres (que l'on retrouve dans les parenthèses et retourne des valeur (juste après le mot-clé *return*).

F est la fonction définie sur \mathbb{R} par :

$$\left\{ \begin{array}{l} \text{si } x \leq 0 \text{ alors } f(x) = x^2 \\ \text{si } 0 < x \leq 1 \text{ alors } f(x) = x \\ \text{si } 1 < x \text{ alors } f(x) = -2x + 3 \end{array} \right.$$

- Écrivez le programme python d'une fonction f qui donne pour résultat f(x) suivant la valeur de x.

```
premiere_fonction.py
def f(x) :
    if x <= 0 :
        resultat = x**2
    elif x <= 1 :
        resultat = x
    else :
        resultat = -2*x + 3
    return resultat
```

- Puis testez

```
>>> from premiere_fonction import f
>>> toto = f(5)
>>> toto
-7
>>> toto = f(0.5)
>>> toto
0.5
```

6.2 Réviser les fonctions python

A faire vous même 18.

- Allez sur python-lycée – cours 2 : <https://www.python-lycee.com/parcours-apprentissage-pl2>
- Regardez la vidéo et reproduisez les commandes sur la console intégrée à la page
- Faire l'exercice 1
- Faire l'exercice 2
- Faire l'exercice 3 (pour les costauds)
- Faire l'exercice 4 (pour les costauds)

A faire vous même 19.

- Sur une droite graduée, A et B sont deux points d'abscisses a et b.
 - Exprimez la distance AB en fonction de a et b
 - Écrivez la fonction python d'une fonction distanceDeuxPoints(a,b) qui donne pour résultat la distance AB



A faire vous même 20.

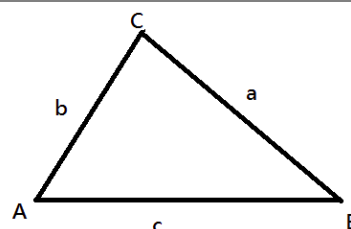
Même chose que l'exercice précédent mais dans le plan avec A(xa, ya) et B(xb, yb)



A faire vous même 21.

ABC est un triangle dont on connaît les longueurs en cm, des côtés.

Écrire le script python d'une fonction rectangleOuPas(a, b, c) qui donne pour résultat True (vrai) si le triangle est rectangle ou False (faux) sinon.



7 Comparez deux valeurs



- Inférieur

```
>>> a = 1
>>> b = 2
```

- Supérieur

```
>>> a = 1
>>> b = 2
```

```
>>> a < b
True
```

- Inférieur ou égal

```
>>> a = 1
>>> b = 2
>>> 2*a <= b
True
```

- Egalité de valeur

```
>>> a = 1
>>> b = 2
>>> 2*a == b
True
```

```
>>> a > b
False
```

- Supérieur ou égal

```
>>> a = 1
>>> b = 2
>>> 2*a >= b
True
```

- Différence de valeur

```
>>> a = 1
>>> b = 2
>>> 2*a != b
False
```

A faire vous même 22.

- Avec python, testez si les 2 valeurs suivantes sont égales (valeurs True/False)

$$\frac{3}{5} + \frac{2}{7}$$
$$\frac{31}{35}$$

- Appelez le professeur.

8 Instructions conditionnelles

Dans un programme, si une condition est vraie alors on exécute des commandes et si elle est fausse on exécute d'autres commandes.

A faire vous même 23.

- Dans la fenêtre script écrivez :

```
1. # -*- coding: UTF-8 -*-
2. nombre1 = 55
3. nombre2=input("Quel est votre nombre ?")
4. nombre2=int(nombre2)
5. if nombre2 > nombre1 :
6.     print("Votre nombre est plus grand que le nombre du prog.")
7. elif nombre2 < nombre1 :
8.     print("Votre nombre est plus petit que le nombre du prog.")
9. else :
10.    print("Bravo ! Vous avez trouvé !")
```

- Attention! Les lignes 6, 8 et 10 ont 4 espaces en début de ligne
- Testez ce script
- Si vous avez compris, vous passez à la suite. Sinon vous appelez le professeur.

A faire vous même 24.

- L'ordinateur joue un dé. Le joueur joue un dé aussi. Le script dit qui est le vainqueur.
- Dans la fenêtre script écrivez ce début de script :

```
from random import randint
valeur_de1=randint(1,6)
```

- Complétez le script avec :
 - Afficher la valeur du dé de l'ordi
 - Lancer et afficher la valeur du dé du joueur
 - Comparer les 2 valeurs
 - Imprimer qui a gagné

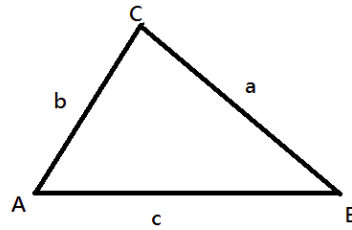
A faire vous même 25. Pour les costauds

- Écrivez un programme qui calcule les 50 premiers termes de la table de multiplication par 13, mais n'affiche que ceux qui sont des multiples de 7.

A faire vous même 26. Pour les costauds

ABC est un triangle dont on connaît les longueurs en cm, des côtés.

Écrire le script python qui donne pour résultat True (vrai) si le triangle est rectangle ou False (faux) sinon.



8.1 Réviser les instructions conditionnelles

A faire vous même 27.

- Allez sur python-lycée – cours 3 : <https://www.python-lycee.com/parcours-apprentissage-pl3>
- Regardez la vidéo et reproduisez les commandes sur la console intégrée à la page
- Faire l' exercice 1
- Faire l' exercice 2
- Faire l' exercice 3 (pour les costauds)

8.2 Réviser les boucles bornées

A faire vous même 28.

- Allez sur python-lycée – cours 4 : <https://www.python-lycee.com/parcours-apprentissage-pl4>
- Regardez la vidéo et reproduisez les commandes sur la console intégrée à la page
- Faire l' exercice 1
- Faire l' exercice 2
- Faire l' exercice 3
- Faire l' exercice 4 (pour les costauds)

9 Affectation augmentée (pour les costauds)

- Exemple simple :

```
« a += 1 » est équivalent à « a = a + 1 »
>>>a = 15
>>>a += 1
>>>a
16
```

- Opérateurs compatibles : + - * /

```
>>>a = 15
>>>a *= 2
>>>a
30
```

10 Boucles non bornées – while (pour les costauds)

Une boucle non bornée ne s'arrête que quand une condition est remplie.

A faire vous même 29.

- Dans la fenêtre script écrivez

```
# -*- coding: UTF-8 -*-
choix = "o"
nombre=1
while choix=="o" :
    nombre = nombre *2
    print("Puissance de 2 : ", nombre)
    choix = input("Tapez o pour continuer ")
print("AU REVOIR !")
```

- A noter : Se trouvent dans la boucle les 3 commandes qui sont décalés de 4 vers la

11 IMPORTANT ! L'indentation dans Python



L'indentation, c'est-à-dire le décalage vers la droite, des lignes est fondamental en Python. C'est cela qui détermine les blocs de commandes qui s'applique à tel ou tel instruction conditionnelle ou qui appartient à telle ou telle boucle.

A faire vous même 30.



- Dans la fenêtre script écrivez

```
# -*- coding: UTF-8 -*-
choix = input("Choisissez une couleur : rouge ou bleu")
if choix=="rouge" :
    print("Vous avez choisi : ", choix)
    print("Vous avez bien choisi")
print("AU REVOIR !")
```

- Dans la fenêtre script modifiez comme ceci :

```
# -*- coding: UTF-8 -*-
choix = input("Choisissez une couleur : rouge ou bleu")
if choix=="rouge" :
    print("Vous avez choisi : ", choix)
print("Vous avez bien choisi")
print("AU REVOIR !")
```

- Quelle est la différence entre les deux scripts ?

.....

- Appelez le professeur.

12 Consolidation des connaissances à l'aide du module Turtle (pour les costauds)



12.1 Premiers scripts

Turtle est un module Python qui permet de tracer des traits dans une fenêtre, un peu comme dans Scratch.

Il va nous permettre d'illustrer une bonne partie de ce que nous avons déjà étudié.

Quelques commandes :

<code>reset()</code>	effacer le dessin
<code>goto(x, y)</code>	aller à l'endroit de coordonnées x, y
<code>forward(distance)</code>	avancer d'une distance donnée
<code>backward(distance)</code>	reculer d'une distance donnée
<code>up()</code>	relever le crayon (pour ne plus dessiner)
<code>down()</code>	abaissier le crayon (pour dessiner)
<code>color(couleur)</code>	$\langle couleur \rangle$ est une chaîne : 'red', 'blue', 'green',...
<code>left(angle)</code>	tourner à gauche d'un angle exprimé en degrés
<code>right(angle)</code>	tourner à droite d'un angle exprimé en degrés

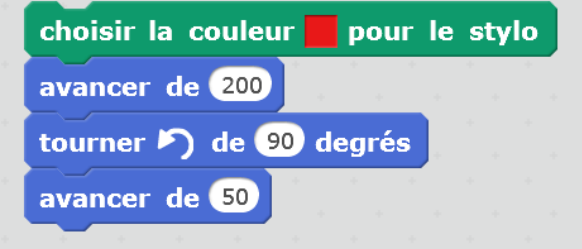
Attention : pour que l'utilisateur puisse fermer la fenêtre de la tortue suivant le processus usuel il faut achever le programme par l'instruction :

```
mainloop()
```

A faire vous même 31.



- Voici un 1^{er} exemple à saisir :

Python	Scratch
<pre>#!/usr/bin/env python # -*- coding: UTF-8 -*- from turtle import * color('red') forward(200) left(90) forward(50) done()</pre>	

- Exécutez-le
- A noter : La 1ère ligne permet de bien préciser à l'ordinateur qu'il s'agit d'un programme python.

A faire vous même 32.

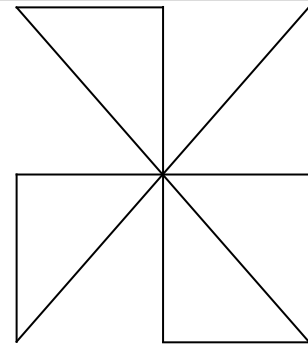
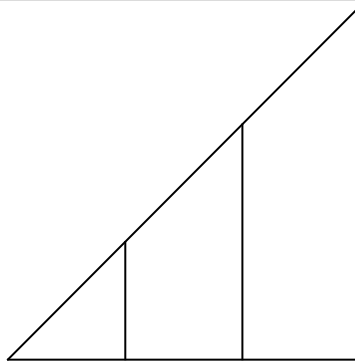
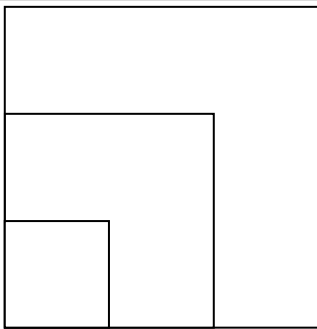
- Voici un 2e exemple avec une boucle while :

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

from turtle import *
color('red', 'yellow')
begin_fill()
forward(200)
left(170)
while abs(pos()) > 1:
    forward(200)
    left(170)
end_fill()
done()
```

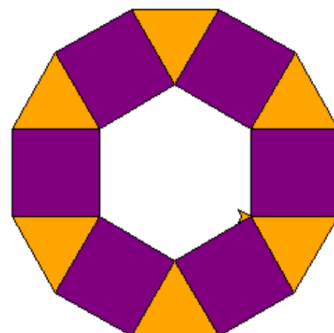
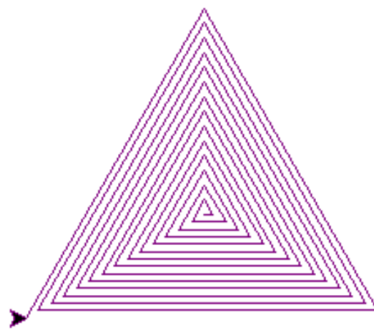
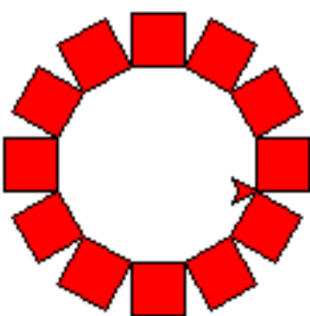
A faire vous même 33.

- Ecrire et programmer les algorithmes permettant de réaliser les figures ci-dessous en Python. Vous pourrez utiliser l'instruction « répéter »



A faire vous même 34. Pour les costauds

- Réalisez les figures suivantes :



12.2 Une autre approche des fonctions avec Turtle

Il est possible de mettre des instructions que l'on utilise plusieurs fois dans une fonction. Une fonction est définie avec le mot-clé : **def**

A faire vous même 35.



- Dans la fenêtre script écrivez

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
from turtle import *
def dessineUnCarre() :
    fillcolor('orange')
    begin_fill()
    for i in range(4) :
        forward ( 75 )
        left(90)
    end_fill ()
```

- Exécutez-le ... Il ne se passe rien !
- Mais l'ordinateur a maintenant en tête les instructions pour faire un carré.
- Dans la console, vous pouvez saisir (n'oubliez pas les parenthèses à la fin) :

```
>>>dessineUnCarre()
>>>done()
```

- On vient de faire appel à la fonction `dessineUnCarre`

A faire vous même 36.



- Dans la fenêtre script écrivez

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

from turtle import *
def dessineUnCarre2(couleur_choisie):
    pendown()
    fillcolor(couleur_choisie)
    begin_fill()
    for i in range(4) :
        forward(75)
        left(90)
    end_fill()
    penup()

for j in range(3):
    dessineUnCarre2('blue')
    forward(100)
done()
```

- Exécutez-le
- L'ordinateur a en tête les instructions pour faire un carré et, en fin de script, appelle cette fonction.
- On vient de faire appel 3 fois à la fonction `dessineUnCarre2`

A faire vous même 37.



- Modifiez le script ci-dessus et ajoutez une fonction `dessineUnRectangle` qui utilisera 3 paramètres : couleur, longueur, largeur

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
def dessineUnRectangle(couleur_choisie, longueur, largeur):
    pendown()
    fillcolor(couleur_choisie)
    begin_fill()
    ...
    ...
    end_fill()
    penup()
```