

Apprendre à programmer avec Python

L5FL024 / LYOU008 - Groupe L

Séance 5 : boucles et variables

Marine DELABORDE

Sorbonne
Nouvelle 
université des cultures



Informations générales

Rappels

- **Horaires** : le mardi de 17h à 19h (discord)
- **Modalités d'enseignement** : distanciel synchrone
- **Contact** : marine.delaborde@gmail.com
- **Ressources** :
 - icampus + <http://www.linguist.univ-paris-diderot.fr/~amsili/Ens/L5FL024.php>
 - discord <https://discord.gg/37vxR6>
- **Fiche d'informations** : <https://framaforms.org/fiche-dinformations-1600359707>
- **Fiche de présence**: <https://lite.framacalc.org/9jbo-presence-python-gl>
- **Modalités de contrôle des connaissances** :
Quiz (4 meilleurs sur 5) 50% + Examen (dernière séance) 50%

Quiz sur icampus

- **Quiz 2/5 : 10 minutes**
- **Les notes des 4 meilleurs quiz seront prises en compte pour représenter 50% de la note finale du cours**

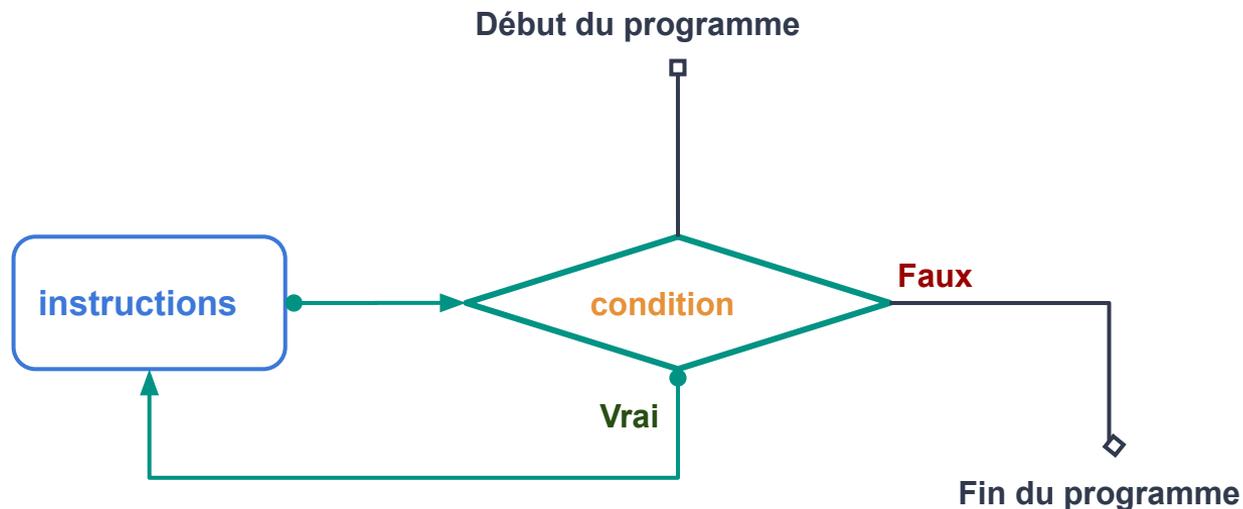
Turtle sur repl.it : https://repl.it/languages/python_turtle

Corrections sur :

<http://www.linguist.univ-paris-diderot.fr/~amsili/Ens/L5FL024.php>

Les boucles While en Python

- Répéter un bloc d'instructions, tant qu'une condition est vraie :



Les boucles While en Python

- Répéter un bloc d'instructions, tant qu'une condition est vraie :

`while condition:`

`instructions`

`instructions`

`...`

Les boucles While en Python : octogone turtle

- Répéter un bloc d'instructions, tant qu'une condition est vraie :

compteur = 0 #initialisation de la variable "compteur" (c'est comme une constante dont la valeur va changer)

while compteur < 8: #tant que la variable "compteur" est strictement inférieure à 8

fd(20) #trace un trait de 20 pixels

rt(45) #pivote le curseur de 45 degrés vers la droite

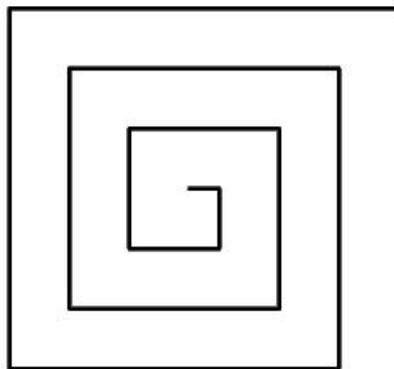
compteur = compteur + 1 #incrmente la variable "compteur" de 1 à chaque boucle

Exercice sur repl.it – correction

3. Reproduire la figure suivante à l'aide d'une boucle while sans fonction :

Premier trait : 10

Dernier trait : 130

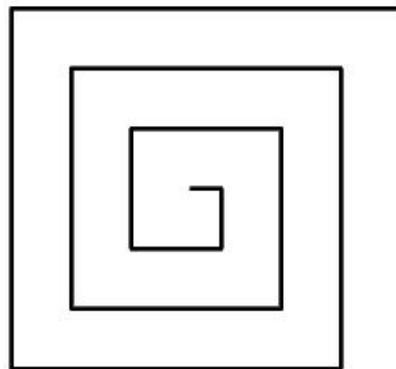


Exercice sur repl.it – correction

4. Reproduire la figure suivante à l'aide d'une boucle while avec une fonction :

Premier trait : 10

Dernier trait : 130

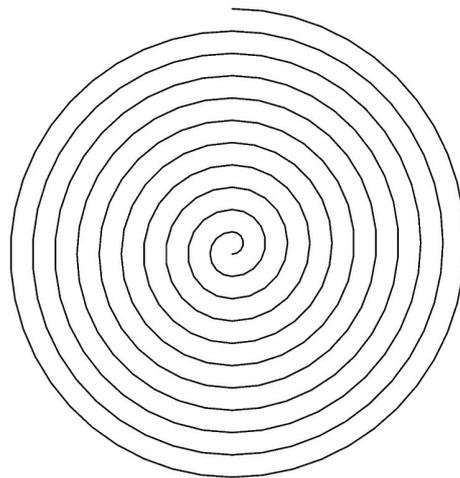


Exercice sur repl.it – correction

5. Reproduire la figure suivante à l'aide d'une boucle while :

Min : 10

Max : 200



Exercice sur repl.it – correction

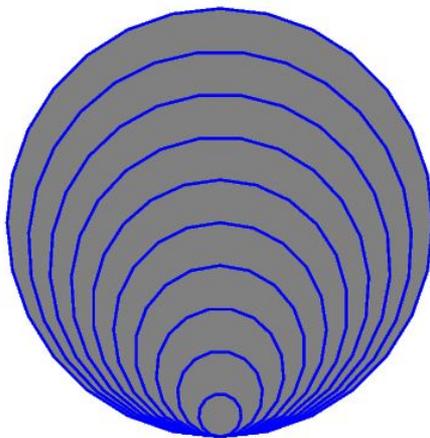
6. Reproduire la figure suivante à l'aide d'une boucle while :

10 cercles gris / contour bleu

Écart entre chaque rayon 10

Plus petit rayon : 10

Plus grand rayon : 100



Exercice sur repl.it – correction

7. Reproduire la figure suivante à l'aide d'une boucle while :

10 cercles

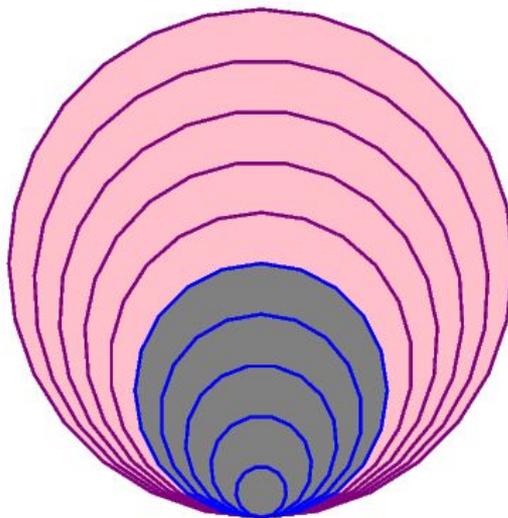
5 roses / contour violet

5 gris / contour bleu

Écart entre chaque rayon 1

Plus petit rayon : 10

Plus grand rayon : 100



Variable : définition

- **Variable** : espace de stockage pour une **valeur**
 - Ex : `compteur = 1`
 - **Opérateur d'affectation**
 - Une variable peut être **utilisée** plusieurs fois
 - La valeur stockée peut **varier** au cours du programme
 - Ex : `compteur = compteur + 1`
 - **Pour l'utilisateur** : une information sur laquelle il y a une étiquette → nom de la variable
 - **Pour l'ordinateur** : une information stockée dans la mémoire vive (RAM) → adresse mémoire

Variable : le nom

- **Nom de variable** : identificateur **unique**
 - Peut contenir des lettres, chiffres et underscore (tiret du bas)
 - Ne peut pas commencer par un chiffre
 - Différenciation entre majuscules et minuscules
 - **Conseils** :
 - un nom qui décrit plutôt le rôle que la fonction (ex : `somme` plutôt que `nombre`)
 - éviter les caractères accentués (même si ok dans python 3)
 - attention au l minuscule qui peut se confondre avec i majuscule
 - attention aux mots réservés de python 3 : `and del from None True as elif global nonlocal try assert else if not while break except import or with class False in pass yield continue finally is raise def for lambda return`
 - La norme PEP 8 décrit les conventions de nommage : <https://www.python.org/dev/peps/pep-0008/>

Variable : le type

- **Type de variable** : information sur le contenu de la variable (sa nature)
 - indique à l'interpréteur python comment manipuler cette information
 - **int** : nombre entier (*integer*)
 - **float** : nombre décimal
 - **str** : chaîne de caractère (*string*)
 - **bool** : booléen (`True` ou `False`)
 - ...
 - **Typage dynamique de Python** : il reconnaît certains types automatiquement (`int`, `float`) mais pour les strings il faut l'aider avec des guillemets.
 - Le **type** de la variable peut **changer**, il dépend de la valeur qui lui est attribuée : `str(`
 - il correspond toujours au type de sa dernière affectation
 - La fonction `type(variable)` retourne le type de la variable
 - La fonction `str(variable)` change le type de la variable en string

Variable : déclaration

- **Initialisation = déclaration :**

- Pas besoin de **déclarer** une variable pour dire qu'elle existe (contrairement à d'autres langages)
- En python il suffit **d'attribuer une valeur** à une variable pour la **déclarer**
- Ex :

```
x = 2.5 #création de la variable x initialisée à 2.5 (x est donc de type "float")
```



Le séparateur décimal est le point et non la virgule

Récapitulatif

- **Variable** = une *étiquette* que l'on colle sur un objet pour lui donner un *nom*
- **Stocker** un objet dans une variable = permettre à python de le **manipuler** (il se souvient de lui)
- Le **type** d'un objet = décrit sa nature (à quoi il ressemble, comment l'utiliser, etc.)
 - **int** = nombres entiers
 - **float** = nombres à virgule
 - **str** = chaînes de caractères (*texte*)
 - **bool** = vrai ou faux

Les **fonctions** sont des *actions* que l'on peut invoquer pour les faire exécuter par Python.

Exercices

1. Stocker le résultat du calcul suivant dans une variable et “imprimer” son résultat dans le terminal : 10/4
 2. Stocker 59 dans une variable et 23 dans une autre.
 - 2.1. Afficher la somme de ces deux valeurs dans le terminal.
 - 2.2. Afficher le type de la somme dans le terminal.
 3. Faire la même chose avec “py” et “thon”.
 4. Faire une boucle while qui permet d’incrémenter une variable `age` d’un “an” à chaque passage de boucle, en partant de 7 jusqu’à 77. Chaque valeur devra s’afficher dans le terminal. À la fin de la boucle, afficher “Fin.”
 5. Afficher “J’ai 20 ans” dans le terminal depuis deux variables : une pour le texte et une pour l’âge¹.
 6. Faire une boucle while qui permet de soustraire 1 “jour” par passage de boucle à une variable `jour` à chaque passage de boucle, en partant de 30 jusqu’à 1. Chaque valeur devra s’afficher dans le terminal selon la forme suivante : “Il reste ?? jour(s) avant les vacances”. À la fin de la boucle, afficher “C’est les vacances !.”
1. Si un message d’erreur s’affiche, lisez-le bien pour essayer de comprendre d’où vient l’erreur.