

Introduction à la Programmation Python

On utilisera l'éditeur en ligne : Future Coder

A chaque utilisation de la fonction **print** ou **input**, un chaton meurt. Faites attention s'il vous plaît.

I Manipulation des objets élémentaires

Les exercices seront à réaliser *dans la console*.

Exercice 1. Entiers

Calculer le quotient et le reste de la division euclidienne par 1454586 de

$$(567 \times 34 - (222 + 103 \times (890 - 888)) \times 5 \times 3^{3127-5^5})^4 + 17471268$$

Exercice 2. Opération Booléennes

Observer l'ordre de priorité des opérations booléennes sur ces exemples.

1. Calculer `False and True or True`
2. Calculer `False and ( True or True )`
3. Calculer `( False and True ) or True`
4. Calculer `not False or True`, puis `not False and False`.

Exercice 3. Flottants

1. Calculer $\frac{(530,4709375 \times 42)^{\frac{1}{6}}}{3,5}$
2. Comparer `150 // 12` et `150 / 12`.
3. Calculer $\sqrt{2}$
4. Calculer $\sqrt{2}^2$

Exercice 4. Néant

Taper `None` dans la console et observer l'absence de retour.

Exercice 5. Comparaisons

Observer le résultat des expressions booléennes suivantes.

```
3 < 4 and 4 < 3
3 < 4 or 4 < 3
3 < 3
3 <= 3
3 == 4 or 3 == 3
3 == 3.0
1/3 * 3 == 1
1/3 - 0.3333333333333333 == 0 # (17 chiffres après la virgule)
1 == 1 - 1e-17
```

```
1 == 0.9999999999999999 # (17 chiffres après la virgule)
0 == 1e-17
```

Observer le résultat de `False == True` et `True == True`. Quelque soit l'expression booléenne `B`, est-il pertinent d'écrire `B == True`? Trouver une écriture plus simple de `B == False`.

II Nommage

Exercice 6. Variables

Essayez de prévoir, puis observez, le résultat dans la console des instructions suivantes. L'ordre des instructions doit être préservé.

```
variable1 = 12
x = variable1
x == variable1
x + x
x == 12
x = x * x
x == 12
variable1 == 12
x != variable1
variable2 = x < variable1
variable2
z = False
y = variable2 or x > variable1 and z
y == True
y
nadine = None
nadine
```

Exercice 7. Fonctions, Appels

Dans l'éditeur, ajoutez les définitions des fonctions suivantes nommées `nadine` et `dopamine` et possédant trois paramètres de type entier.

```
def nadine(parametre1, parametre2, parametre3):
    variable1 = parametre1 + parametre2
    return variable1 ** parametre3

def dopamine(parametre1, parametre2, parametre3):
    variable1 = (parametre1 * parametre2) % parametre3
```

Exécutez le code de l'éditeur, puis, dans la console, appelez la fonction `nadine` sur les entrées suivantes :

- 3, 4, 2
- 4, 2, 3
- 0, -5, 10

Toujours dans la console, essayez de prévoir, puis observez, le résultat des instructions suivantes.

```
variable1 = 12
nadine(6,7,8)
variable1
parametre1 = 56
nadine(6,7,8)
nadine(56,7,8)
nadine(nadine(3,4,2), 10, nadine(1,1,nadine(1,2,1)))
dopamine(3,4,2)
dopamine(2,4,5) == None
```

Exercice 8. Fonction, Définition

Définissez les fonctions suivantes. Vous devrez appeler vos fonctions sur *plusieurs jeux* d'entrées bien choisies pour vérifier le résultat.

1. racine qui prend en entrée un flottant x , et un entier naturel non nul n , et qui renvoie la racine $n^{\text{ème}}$ de x .
2. ordonnees qui prend en entrée quatre entier, a , b , c et d et qui renvoie un booléen vrai si et seulement si $a < b < c < d$.
3. quasiegal qui prend en entrée deux flottants x , y et un flottant positif e et qui renvoie un booléen vrai si et seulement si $|x - y| < e$

Exercice 9. Fonctions, Environnement

Dans l'éditeur, ajoutez les définitions des fonctions suivantes nommées f et g et possédant des paramètres de type entier. On ajoutera aussi la définition des variables globales z et x .

```
z = 12
x = 42

def f(x,y) :
    z = 2
    return z * x ** g(y)

def g(x) :
    return z - x

y = f(2,10)
```

Essayez de prévoir les valeurs de x , y et z , puis exécutez le code de l'éditeur et observez la valeur de y . Simulez le code de l'éditeur dans Python Tutor et observez les environnements.

Combien d'environnements existeront simultanément au plus lors de l'exécution de la fonction $f(f(g(21),4),f(f(1,2),g(5)))$? Vérifiez en utilisant Python Tutor.

III Flot d'exécution

Exercice 10. Appels

Dans l'éditeur, écrivez le code suivant.

```
def f(x,y) :  
    return h(x) + g(h(y))  
  
def g(x) :  
    return h(x*2)  
  
def h(x) :  
    return x + 5  
  
y = f(12,21)
```

Listez les lignes exécutées de l'éditeur, dans l'ordre d'exécution. Certaines lignes seront exécutées plusieurs fois, elles apparaîtront à chaque exécution dans la liste.

Vérifiez avec Python Tutor ou snoop.

Exercice 11. Conditionnelles

Dans l'éditeur, écrivez le code suivant.

```
def f(x,y) :  
    z = x*y  
    if z == 12 or z > 100 :  
        z = z * 2  
    elif z < 50 :  
        if z < 0 :  
            z = - z  
        z = z * 3  
    else :  
        z = z * 12  
    return z
```

Listez les lignes exécutées de cette fonction, dans l'ordre d'exécution, lors de son appel sur les entrées suivantes :

1. 10 et 20
2. 5 et 6
3. 3 et 4
4. 7 et 10
5. -3 et 2

Vérifiez avec Python Tutor ou snoop.

Exercice 12. Bissextile

Définissez une fonction bissextile, qui prend en entrée un entier positif annee, et qui renvoie un booléen vrai si et seulement si l'année est bissextile, selon les règles actuelles. *i.e* si l'un des deux critères suivants est respecté :

1. L'année est divisible par 4 mais pas par 100
2. L'année est divisible par 400

https://fr.wikipedia.org/wiki/Ann%C3%A9e_bissextile

Exercice 13. Boucles

Dans l'éditeur, écrivez le code suivant.

```
def f(x) :  
    z = 1  
    r = 0  
    while z <= x :  
        if z % 5 == 0 :  
            r = r + 1  
        z = z + 1  
    n = x  
    while r != 0 :  
        n = n - 5  
        r = r - 1  
    return n
```

Listez les lignes exécutées de cette fonction, dans l'ordre d'exécution, lors de son appel sur les entrées suivantes :

1. 12
2. 21
3. 0
4. -5
5. 30

Vérifiez avec Python Tutor ou snoop.

Que fait la fonction f ?