

INFO-H-100 - Programmation

TP 2 - Scripts et fonctions

Ex. 1. Écrivez un script python qui demande à l'utilisateur d'entrer le rayon d'un cercle, et qui s'en sert pour calculer l'aire et la circonférence du cercle. Pour ce faire, utilisez la valeur de π trouvée dans le module `math`.

Ex. 2. Écrivez un programme qui demande à l'utilisateur d'entrer successivement les abscisses et ordonnées de deux points, et qui calcule la distance euclidienne entre ces deux points. Pour mémoire, la distance euclidienne est définie par :

$$d(A_{[x_a, y_a]}, B_{[x_b, y_b]}) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Cherchez, dans la documentation du module `math` sur `docs.python.org`, la fonction qui permet de calculer une racine carrée.

Ex. 3. Dans la documentation, trouvez le moyen de générer :

- un nombre aléatoire réel dans l'intervalle $[0, 1[$ et
- un nombre aléatoire entier dans l'intervalle $[1, 6]$.

Pour chacun des exercices suivants, écrivez la fonction demandée et testez-la avec plusieurs valeurs pertinentes.

Ex. 4. Écrivez une fonction qui, étant donnés deux points (x_1, y_1) et (x_2, y_2) , calcule et retourne la distance euclidienne entre ces deux points.

Ex. 5. Étant donnés 3 points entrés par l'utilisateur, écrivez une fonction qui calcule et affiche le périmètre du triangle correspondant. On suppose que les trois points ne sont pas alignés.

Ex. 6. Écrivez une fonction qui renvoie la valeur absolue d'un nombre donné en paramètre.

Ex. 7. Écrivez une fonction qui renvoie le maximum de deux valeurs données en paramètres.

Ex. 8. Écrivez une fonction qui détermine l'aire d'un cercle de rayon donné comme paramètre.

Ex. 9. Étant donnés deux nombres (entiers), écrivez une fonction qui détermine si le premier est (exactement) divisible par le second.

Ex. 10. Étant donnés l'aire d'un disque, centré à l'origine, et les coordonnées X et Y d'un point. Écrivez une fonction qui reçoit ces 3 informations comme paramètres et qui détermine si le point se situe, oui ou non, à l'intérieur du disque.

Ex. 11. Écrivez une fonction qui renvoie le maximum de trois valeurs données en paramètres.

Ex. 12. Écrivez une fonction qui permet d'afficher un mot, passé en paramètre, en l'entourant d'étoiles. Par exemple, s'il s'agit du mot `machine`, la fonction devra afficher :

```
*****
* machine *
*****
```

Ensuite écrivez un programme qui demande à l'utilisateur d'entrer un mot, et qui affiche ce mot en l'entourant d'étoiles seulement si ce mot contient plus de 5 caractères, sinon il affichera le mot tel quel. Utilisez la fonction que vous avez programmée à l'étape précédente.

Indice : vous aurez besoin de pouvoir calculer la longueur d'une chaîne de caractères ; par exemple, `len("hello")` renvoie 5. Pour produire une chaîne de n fois la même lettre, vous pouvez multiplier une chaîne d'une lettre par n . Ainsi :

```
>>> 'a' * 5
'aaaaa'
>>>
```

Ex. 13. Exécutez le code suivant en utilisant *Visualize* de *PythonTutor*. (<http://www.pythontutor.com/visualize.html>) Observez le passage des valeurs en paramètres.

```
def get_ratio(x, y):
    return x / y
    print('done.')

print(get_ratio(3,4))
print(get_ratio(4,3))

x = 4
y = 3
print(get_ratio(y, x))
```

À quels endroits les variables `x` et `y` sont-elles locales ou globales ? L'instruction `print('done.')` est-elle exécutée ? Si non, pourquoi ?

Ex. 14. Décrivez ce que fait la fonction `swap`. Utilisez un tableau pour faire un diagramme d'états-transitions de ce programme. Ensuite, exécutez le code suivant en utilisant *Visualize* de *PythonTutor*. (<http://www.pythontutor.com/visualize.html>)

```
def swap(a, b):
    c = a
    a = b
    b = c

value1 = 5
value2 = 10

print(value1, value2)
swap(value1, value2)
print(value1, value2)
```

Obtient-on le résultat escompté ? Si non, pourquoi est-ce que cela ne fonctionne pas ?

INFO-H-100 - Programmation

TP 2 - Scripts et fonctions

Corrections

Solution de l'exercice 1:

```
import math
radius = input("Entrez le rayon d'un cercle : ")
radius = float(radius)
area = 2 * math.pi * radius
circum = math.pi * radius ** 2
print("L'aire vaut : " + str(area))
print("La circonférence vaut : " + str(circum))
```

Solution de l'exercice 2:

```
import math
x1 = float(input("Entrez x1 : "))
y1 = float(input("Entrez y1 : "))
x2 = float(input("Entrez x2 : "))
y2 = float(input("Entrez y2 : "))
distance = math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
print("La distance vaut " + str(distance))
```

Solution de l'exercice 12:

```
word = input("Entrez un mot : ")
length = len(word)
print('*' * (length + 4))
print('* ' + word + ' *')
print('*' * (length + 4))
```

Solution de l'exercice 3:

```
>>> import random
>>> random.random()
0.584479191192803
>>> random.random()
0.5966986731268477
>>> random.random()
0.3668177600286743
>>> random.randint(1,6)
1
>>> random.randint(1,6)
1
>>> random.randint(1,6)
6
>>> random.randint(1,6)
5
```

Solution de l'exercice 4:

```
import math

def distance(x1, y1, x2, y2):
    return math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)

print(distance(0, 0, 1, 1)) #prints 1.41421356237
print(distance(0, 0, 0, 1)) #prints 1.0
print(distance(1, 1, 1, 1)) #prints 0.0
```

Solution de l'exercice 5:

```
import math

def distance(x1, y1, x2, y2):
    return math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
```

```
def perimeter(x1, y1, x2, y2, x3, y3):
    side_1 = distance(x1, y1, x2, y2)
    side_2 = distance(x2, y2, x3, y3)
    side_3 = distance(x3, y3, x1, y1)
    return side_1 + side_2 + side_3

print(perimeter(0, 0, 0, 1, 1, 0)) #prints 3.41421356237
```

Solution de l'exercice 6:

```
def absolute_value(x):
    value = x
    if value < 0 :
        value = -value
    return value

print(absolute_value(0))      #prints 0
print(absolute_value(-1.5))   #prints 1.5
print(absolute_value(10))     #prints 10
```

Solution de l'exercice 7:

```
def maximum(a,b):
    maxi = b
    if a > maxi :
        maxi = a
    return maxi

print(maximum(0,0))          #prints 0
print(maximum(1.5,2))        #prints 2
print(maximum(1.3,0.2))      #prints 1.3
```

Solution de l'exercice 8:

```
import math

def circle_area(radius):
    return math.pi * radius ** 2

print(circle_area(1))        #prints pi
print(circle_area(2))        #prints 12.5663706144
print(circle_area("Bart"))    #produce an error
```

Solution de l'exercice 9:

```
def is_divisible(a, b):
    return a % b == 0

print(is_divisible(10, 2))   #prints True

if is_divisible(1, 2):
    print("1 est divisible par 2")
else:
    print("1 n'est pas divisible par 2")

print(is_divisible(5, 3))    #prints False
```

Solution de l'exercice 10:

```
import math

def distance(x1, y1, x2, y2):
    return math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)

def radius(area):
    return math.sqrt(area / math.pi)

def is_in_disk(area, x, y):
    return distance(0, 0, x, y) <= radius(area)

print(is_in_disk(13, 1, 1)) #prints True
print(is_in_disk(2, 2, 2)) #prints False
```

Solution de l'exercice 11:

```
def max3(a, b, c):
    maximum = a
    if b > maximum:
        maximum = b
    if c > maximum:
        maximum = c
    return maximum

print(max3(1, 2, 3)) #prints 3
print(max3(3, 2, 2)) #prints 3
print(max3(1, 5, 2)) #prints 5
print(max3(5, 5, 5)) #prints 5
```

Explications : on fait l'hypothèse que la première valeur est le maximum. Si la deuxième est plus grande que le maximum courant, alors on a un nouveau maximum. Si la troisième est encore plus grande on a également un nouveau maximum.