

INFO-H-100 - Programmation

TP 3 - Tests et fonctions (1)

sur base d'exercices du cours INFO-F-101

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

Ex. 1. Ecrivez 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. 2. Etant 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. 3. Ecrivez une fonction qui renvoie la valeur absolue d'un nombre donné en paramètre.

Ex. 4. Ecrivez une fonction qui renvoie le maximum de deux valeurs données en paramètres.

Ex. 5. Ecrivez une fonction qui détermine l'aire d'un cercle de rayon donné comme paramètre.

Ex. 6. Etant donnés deux nombres (entiers), écrivez une fonction qui détermine si le premier est (exactement) divisible par le second.

Ex. 7. Etant 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. 8. Écrivez une fonction qui renvoie le maximum de trois valeurs données en paramètres.

INFO-H-100 - Programmation

TP 3 - Tests et fonctions (1)

Corrections

sur base d'exercices du cours INFO-F-101

Solution de l'exercice 1:

```
import math

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

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

Solution de l'exercice 2:

```
import math

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

def perimetre(x1,y1,x2,y2,x3,y3):
    cote1 = distance(x1,y1,x2,y2)
    cote2 = distance(x2,y2,x3,y3)
    cote3 = distance(x3,y3,x1,y1)
    return cote1 + cote2 + cote3

print perimetre(0,0,0,1,1,0) #affiche 3.41421356237
```

Solution de l'exercice 3:

```
def valeur_absolue(x):
    if x < 0 :
        return -x
    return x

print valeur_absolue(0) #affiche 0
print valeur_absolue(-1.5) #affiche 1.5
print valeur_absolue(10) #affiche 10
```

Autre solution :

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

print valeur_absolue(0) #affiche 0
print valeur_absolue(-1.5) #affiche 1.5
print valeur_absolue(10) #affiche 10
```

Solution de l'exercice 4:

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

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

Solution de l'exercice 5:

```

import math

def aire_cercle(rayon):
    return math.pi * rayon ** 2

print aire_cercle(1)           #affiche pi
print aire_cercle(2)           #affiche 12.5663706144
print aire_cercle("Bart")      #provoque une erreur

```

Solution de l'exercice 6:

```

def est_divisible(a,b):
    return a % b == 0

print est_divisible(10,2) #affiche True

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

print est_divisible(5,3) #affiche False

```

Solution de l'exercice 7:

```

import math

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

def rayon(aire):
    return math.sqrt(aire/math.pi)

def est_dans_disque(aire,x,y):
    return distance(0,0,x,y) <= rayon(aire)

print est_dans_disque(13,1,1) #affiche True
print est_dans_disque(2,2,2) #affiche False

```

Solution de l'exercice 8:

```

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

print max3(1,2,3) #affiche 3
print max3(3,2,2) #affiche 3
print max3(1,5,2) #affiche 5
print max3(5,5,5) #affiche 5

```

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