|  |  |  |
| --- | --- | --- |
|  | **Python**  **License Pro GMP**  **TD 5 :** Fonctions | |
|  |  |

# Exercices

## Exécution

Pour l’exécution du code contenu dans le premier cadre, veuillez écrire dans le second le résultat affiché dans la console Python.

*Exemple :*

def helloWorld() :

print('Hello World')

helloWorld()

*Réponse :*

Hello World

### Exécution 1

def my\_max(a,b):

if a > b:

return(a)

else:

return(b)

*Affichage :*

### Exécution 2

def my\_max(a,b):

if a > b:

return(a)

else:

return(b)

my\_max(3,1)

*Affichage :*

### Exécution 3

def my\_max(a,b):

if a > b:

return(a)

else:

return(b)

print(my\_max(3,1))

*Affichage :*

### Exécution 4

def my\_max(a,b):

if a > b:

return(a)

else:

return(b)

maximum = 0

print(maximum)

maximum = my\_max(3,1)

print(maximum)

*Affichage :*

### Exécution 5

def my\_max(a,b):

if a > b:

print(a)

else:

print(b)

*Affichage :*

### Exécution 6

def my\_max(a,b):

if a > b:

print(a)

else:

print(b)

my\_max(3,1)

*Affichage :*

### Exécution 7

def my\_max(a,b):

if a > b:

print(a)

else:

print(b)

my\_max(3,1)

*Affichage :*

### Exécution 8

def my\_max(a,b):

if a > b:

print(a)

else:

print(b)

maximum = 0

print(maximum)

maximum = my\_max(3,1)

print(maximum)

*Affichage :*

### Exécution 9

def my\_max(a,b):

if a > b:

return(a)

return(b)

print(my\_max(3,1))

*Affichage :*

### Exécution 10

def my\_max(a,b):

if a > b:

print(a)

print(b)

my\_max(3,1)

*Affichage :*

### Exécution 11

def my\_max(a,b):

if a > b:

print(a)

else:

print(b)

my\_max(3,1)

*Affichage :*

### Exécution 12

def my\_max(a,b):

if a > b:

return(a)

return(b)

print(my\_max(my\_max(3,1),my\_max(0,4)))

*Affichage :*

### Exécution 13

def my\_max(a,b):

if a > b:

maximum = a

else:

maximum = b

return(maximum)

print(maximum)

*Affichage :*

### Exécution 14

def my\_max(a,b):

if a > b:

maximum = a

else:

maximum = b

return(maximum)

my\_max(1,3)

print(maximum)

*Affichage :*

### Exécution 15

maximum = 0

def my\_max(a,b):

if a > b:

maximum = a

else:

maximum = b

return(maximum)

my\_max(1,3)

print(maximum)

*Affichage :*

### Exécution 16

maximum = 0

def my\_max(a,b):

if a > b:

maximum = a

else:

maximum = b

return(maximum)

maximum = my\_max(1,3)

print(maximum)

*Affichage :*

### Exécution 17

def show\_me(a):

return(a)

print(a)

*Affichage :*

### Exécution 18

def show\_me(a):

return(a)

print(a)

show\_me('Hello')

*Affichage :*

### Exécution 19

def show\_me(a):

print(a)

return(a)

show\_me('Hello')

*Affichage :*

### Exécution 19

def show\_me(a):

print(a)

return(a)

print(show\_me('Hello'))

*Affichage :*

## Réalisation

Pour chaque énoncé, veuillez écrire le code de la fonction ou de la procédure demandée.

### Réalisation 1

Veuillez écrire le code de la fonction proc\_my\_add qui prend deux entiers en paramètres et affiche la somme de ces deux entiers. Faire un appel à cette fonction pour voir s’afficher 10.

### Réalisation 2

Veuillez écrire le code de la fonction my\_add qui prend deux entiers en paramètres et retourne la somme de ces deux entiers. Faire un appel à cette fonction pour voir s’afficher 10.

### Réalisation 3

Veuillez écrire le code de la fonction my\_abs qui prend un entier en paramètre et retourne la valeur absolue de cet entier. Faire un appel à cette fonction pour voir s’afficher 5.

### Réalisation 4

A l’aide de la fonction print et des fonctions définies dans cet exercice, veuillez afficher la somme des valeurs absolues de -3 et -7 en une seule ligne de code et quatre appels de fonction.

### Réalisation 5

A l’aide de la fonction print et des fonctions définies dans cet exercice, veuillez afficher la somme des valeurs absolues de -3 et -7 en une seule ligne de code et trois appels de fonction.