SERIE : EXERCICES DE FONCTIONS

Application 1

Créez une fonction qui renvoie True si un entier est divisible par 5, sinon renvoie False.

Exemple:
isDivisibleBy5(5) → True
isDivisibleBy5(-5) → True
isDivisibleBy5(3) → False

```
def isDivisibleBy5(n):
return not n % 5
```

Application 2

Créez une fonction qui accepte une liste et renvoie le dernier élément de la liste.

Exemple:
getLast([1, 2, 3]) → 3
getLast(["A", "B", "C"]) → "C"
getLast([10, "WayToLearnX", True]) → True

```
def getLast(liste):
return liste[-1]
```

Application 3

Corrigez le code suivant (Erreur de syntaxe), pour calculer le carré.

Exemple:
carre(2) → 4
carre(4) → 16

```
def carre(b):
return b ** 2
```

Application 4 :

Créez une fonction qui évalue une équation.

Exemple:
evalute(1+2+3) → 6
evalute(4*2/2) → 4.0
evalute((3+2)*(3+2)) → 25

```
def evalute(op):
return op
```

Application 5 :

Créez une fonction qui renvoie la valeur ASCII du caractère transmis.

Exemple:
charToAscii("A") → 65
charToAscii("a") → 97
charToAscii("+") → 43

```
def charToAscii(c):
return ord(c)
```

Application 6 :

Créez une fonction qui renvoie True si une chaîne contient des espaces.

Exemple:
containSpaces("WayToLearnX") → False
containSpaces("Welcome to WayToLearnX") → True
containSpaces(" ") → True

```
def containSpaces(str):
return " " in str
```

Application 7 :

Créez une fonction qui prend un mot et détermine s'il est pluriel ou singulier. Un mot pluriel est celui qui se termine par « s ». S'il est pluriel renvoyer TRUE sinon FALSE.

Exemple:
checkIsPlural("enfants") → True
checkIsPlural("filles") → True
checkIsPlural("fille") → False
checkIsPlural("enfant") → False

```
def checkIsPlural(str):
return str.endswith('s')
```

Application 8 :

Créez une fonction qui prend un nombre comme argument et renvoie « pair » pour les nombres pairs et « impair » pour les nombres impairs.

Exemple:
check(2) → "pair"
check(7) → "impair"
check(22) → "pair"

```
def check(n):
```

```
return 'impair' if n % 2 else 'pair'
```

Application 9 :

Créez une fonction qui renvoie True si une chaîne est vide, sinon False.

Exemple:
isEmpty("WayToLearnX") → False
isEmpty(" ") → False
isEmpty("") → True

```
def isEmpty(str):
return not str
```

Application 10 :

Créez une fonction qui renvoie le nombre de valeurs « True » qu'il y a dans une liste.

Exemple:
count([False, False, True, True, True]) → 3
count([False, False, False]) → 0
count([]) → 0

```
def count(liste):
return sum(liste)
```

Application 11 :

Vous avez embauché trois commerciales et vous les payez. Créez une fonction qui prend trois nombres (le salaire horaire de chaque commerciale) et renvoie la différence entre le commerciale le mieux payé et le moins payé.

Exemple:
getDiff(200, 10, 90) → 190
//200 - 10 = 190
getDiff(56, 29, 16) → 40
getDiff(2, 10, 5) → 8

```
def getDiff(*args):
return max(args) - min(args)
```

Application 12 :

Créez une fonction qui valide si un pont est sûr à marcher (c'est-à-dire, qu'il n'a pas d'espace entre #).

Exemple:
marcher('###### #########') → False
marcher('#########') → True

marcher('#') ⟶ True

```
def marcher(pont):
return ' ' not in pont
```

Application 13 :

Créez une fonction qui prend une liste de nombres et renvoie le minimum et le maximum dans une liste [Min, Max].

Exemple:
getMinMax([8, 1, 9, 2, 6]) ⟶ [1, 9]
getMinMax([22, 2]) ⟶ [2, 22]
getMinMax([5]) ⟶ [5, 5]

```
def getMinMax(liste):
return [min(liste), max(liste)]
```

Application 14 :

Créez une fonction qui prend un nombre (de 1 à 10) et renvoie une chaîne de tirets correspondante.

Exemple:
convert(2) ⟶ "--"
convert(6) ⟶ "------"
convert(4) ⟶ "----"

```
def convert(n):
return "-" * n
```

Application 15 :

Créez une fonction qui prend une chaîne (un nom aléatoire). Si le dernier caractère du nom est un « s », retournez TRUE, sinon retournez FALSE.

Exemple:
checkS("Thomas") ⟶ True
checkS("Ali") ⟶ False
checkS("Alex") ⟶ False
checkS("Alvis") ⟶ True

```
def checkS(str):
return str.endswith('s')
```

Application 16 :

Créer une fonction qui divise a et b (a / b).

Exemple:
div(4,2) ⟶ 2.0
div(10,2) ⟶ 5.0
div(1,0) ⟶ "Erreur"

```
def div(a, b):
return a/b if b!=0 else "Erreur"
```

Application 17 :

Écrivez un programme Python pour renvoyer le reste de deux nombres. Il existe un seul opérateur en Python, capable de fournir le reste d'une division. Deux nombres sont transmis comme paramètres. Le premier paramètre divisé par le deuxième paramètre.

Exemple:
resteDiv(1, 3) ⟶ 1
resteDiv(2, 4) ⟶ 2
resteDiv(3, 3) ⟶ 0

```
def resteDiv(a, b):
return a % b
```

Application 18 :

Créez une fonction qui prend une liste d'éléments et renvoie le premier et le dernier éléments sous forme de nouvelle liste.

Exemple:
getFirstLast([1, 2, 3, 4, 5, 6, 7]) ⟶ [1, 7]
getFirstLast(["A", "B", "C", "D"]) ⟶ ["A", "D"]
getFirstLast(["A", 2, True, None]) ⟶ ["A", None]

```
def getFirstLast(liste):
return [liste[0], liste[-1]]
```