

# Exercices CTD1 - DIU-EIL 2020-2021



## 1 Exercices sur les variables, entrées/sorties

In [ ]: *#Pour les lignes suivantes : Que vaut x ? Quel est son type ?*

```
x = 2+3
print(x,type(x))
x = "abc"
print(x,type(x))
x = 'abc'+ "d"
print(x,type(x))
x = True and (3<2)
print(x,type(x))
x = 3.1 - 2.5 - 0.6 # attention au piège des float
print(x,type(x))
x = 3,1415926
print(x,type(x))
x = 2 + 3.0
print(x,type(x))
x = 3**4
print(x,type(x))
x = 12%3
print(x,type(x))
x = abs(32-42)
print(x,type(x))
x = "abc" + 3
print(x,type(x))
x = "abc" * 3
print(x,type(x))
```

In [ ]: *# Pour les lignes suivantes : Que se passe-t'il à l'écran ?*

```
x=3+4
y='3+4'
z=True
print(x,y,z)

#suivi de
print(x)
print(y)
print(z)

x=True
x=False and x
y='False and x'
print(x)
print(y)
```

```

pi=3.14159
r=2
surface=pi*r*r
print('Surface =',surface)

```

```

In [ ]: # Pour les lignes suivantes :
        # que vaut x quand on saisit 04 au clavier ? quel est son type ?
x = input()
print(x,type(x))

x = input("04?")
print(x,type(x))

x = int(input())
print(x,type(x))

x = float(input())
print(x,type(x))

x = bool(input()) #surprise !!! et si on saisit False ? bool("False") ??? bool(False)
print(x,type(x))

```

## 2 Exercices sur les booléens, sur la sélection

```

In [ ]: #1ère série : pour chaque ligne quelle valeur ?
#a)
not(True)
#b)
True and False
#c)
False or False
#d)
True and True
#e)
not (False or True)
#f)
False and ((True and True) or (True and (False or (True and False))))
#g)
True or (True and (False or ((False and True) or True)))

#2ème série : pour chaque ligne quelle valeur ?
#a)
not(not(a))
#b)
not(not(not(b)))
#c)
False or a
#d)
False and a
#e)
True and a
#f)
True or a
#g)
a and not(a)
#h)
a or not(a)
#i)
not(a==b) and (a==b)

```

```
In [ ]: #Donner un code équivalent
        #hypthèse la condition n'est pas modifiée
        if condition :
            instruction1
        if not condition :
            instruction2
```

### 3 Exercices de répétition while et à double condition

```
In [ ]: #Ecrire un programme qui permet de calculer (en une seule lecture)
        #la moyenne et le minimum d'entiers positifs ou négatifs saisis au clavier
        #la saisie s'arrête quand l'utilisateur saisit 0
```

```
In [ ]: #Ecrire un programme qui permet de compter (en une seule lecture)
        #le nombre d'entiers pairs saisis
        # La saisie s'arrête quand l'utilisateur saisit 0
        # ou quand on a saisi 10 nombres.
        # Faire au moins une version avec au moins un booleen
```

### 4 Exercice avec while/for (double boucle)

```
In [ ]: #Ecrire le programme qui permet d'afficher les entiers naturels strictement positifs et
        #strictement inférieurs à n qui vérifient la propriété suivante :
        #somme des chiffres = produit des chiffres.
        #Bien évidemment les naturels inférieurs à 10 respectent cette propriété.
        # Exemple : 123 respecte cette propriété car 1+2+3=1*2*3=6.
        #121 ne respecte pas cette propriété car 1+2+1=4 ≠ 1*2*1=2.
```

### 5 Exercices sur les fonctions

Qu'affichent les programmes suivants ?

```
In [ ]: def fonction1(a):
        z = a + 2
        return z

        x = 20
        b = fonction1(x)
        print(b)
```

```
In [ ]: def fonction2(a):
        if (a % 2 == 0):
            return "pair"
        return "impair"

        print(fonction2(2))
```

```
In [ ]: #METTRE CELLULE EN "Code" POUR EVALUER
        def foo():
            print("Hello World!")

        def bar():
            print("Hello Toulouse!")
```

```

foo ()

In [ ]: #Quelle fonction ne renvoie pas la somme des deux paramètres ?
#Prog 1
def somme(a,b):
    print(a+b)

#Prog 2
def somme(a,b):
    return a+b

#Prog 3
def somme(a,b):
    print(a+b)
    return a+b

#Prog 4
def somme(a,b):
    return a+b
    print(a+b)

In [ ]: def fonction_z1(a):
        a=8
        return 7

        x = 20
        y = fonction_z1(x)
        print(x)

In [ ]: def fonction_z2(a):
        b=8
        return 7

        b = 20
        y = fonction_z2(b)
        print(b)

In [ ]: def calcul(w,x,z):
        a=x
        b=w+1
        return a+b+3

        z=calcul(3,2,0)
        print(z)

In [ ]: a=10
        def somme(a,b):
            return a+b

        a=3
        a=somme(a,5)
        print(a)

```

## 6 Exercices sur les listes

Qu'affichent les codes suivants ?

```

In [ ]: seq1 = [1,2,3]
        seq2 = [0,seq1[2],100]
        seq1[2] = 10
        print(seq2[1])

```

```

In [ ]: seq1 = [1,2,3]
        seq2 = [4,5,6]
        seq3 = seq2
        seq2[0] = 10

        seq4 = seq1 + seq3
        print(seq4)
        seq1[0]=45
        print(seq4)

In [ ]: def fctBeteSurListe(maliste):
        maliste[2] = 0
        return maliste

        x = [1,2,3]
        y = fctBeteSurListe(x)
        x[0]=13
        print(y)

In [ ]: ma_liste = [1,2,3,4,5,6]
        autreListe = maListe[1:3]
        encoreListe = maListe
        maListe[1]=27
        print(maListe[1],autreListe[1], encoreListe[1])

In [ ]: ma_liste = [1,2,3,4,5,6]
        def fctSurListe(uneListe):
            uneListe[2]=27
            return uneListe

        autreListe = maListe[0:6]
        resultatListe = fctSurListe(maListe)
        print(maListe[2],autreListe[2],resultatListe[2])

```