

TP 03

Expression, variable, fonctions et structures algorithmiques

Savoirs et compétences :

- ☐ AA.C4 : Comprendre un algorithme et expliquer ce qu'il fait
- ☐ AA.C5 : Modifier un algorithme existant pour obtenir un résultat différent
- ☐ AA.C6 : Concevoir un algorithme répondant à un problème précisément posé
- ☐ AA.C8 : Écrire des instructions conditionnelles avec alternatives, éventuellement imbriquées
- ☐ AA.C9 : Choisir un type de données en fonction d'un problème à résoudre
- ☐ AA.S7 : Expressions et instructions simples
- ☐ AA.S8 : Instructions conditionnelles
- ☐ AA.S9 : Instructions itératives
- ☐ AA.S10 : Notion de fonction informatique

Activité 1 : Fonctions en Python

Q1:

```
def moy_extr(L):
    """Renvoie la moyenne du premier et du dernier élément de L"""
    return (L[0]+L[-1])/2
```

Q2:

```
def incr_sans_effet_de_bord(L):
    """Renvoie une nouvelle liste identique à L,
    sauf le premier terme incrémenté de 1"""
    c = L.copy()
    c[0] = c[0]+1
    return c

def incr_avec_effet_de_bord(L):
    """Incrémente le premier élément de L de 1"""
    L[0] = L[0]+1
    return None
```

Activité 2 : Boucles IF, FOR, WHILE

Q3: Indentation vraie.

```
x=0
y=5
t=False
if x>=1:
    t=True
if y<=6:
    t=True
```

Indentation fausse.

```
x=0
y=5
```

```
t=False
if x>=1:
    t=True
    if y<=6:
        t=True
```

Q4:

```
from random import randrange
n= randrange(100) # Un entier aléatoire entre 0 et 99

if n <= 10:
    print("Trop petit")
elif n >= 50:
    print("Trop grand")
else:
    print("Juste comme il faut")
```

Q5:

```
def inv(n):
    """Somme les inverses des n premiers entiers naturels non nuls"""
    s = 0
    for k in range(n):
        x = 1/(k+1)
        s = s+x
    return s
```

Activité 3 : Algorithme glouton – Problème du rendu de monnaie

```
def rendre_monnaie(cout,somme_client):
    """
    Retourne un dictionnaire contenant le type de billets
    ou pièces à rendre ainsi que le nombre de chacun d'entre eux

    Keywords arguments :
    cout : somme à payer
    somme_client : argent donné par le client
    """
    dico_monnaie={}
    montant_a_rendre = somme_client - cout
    if montant_a_rendre < 0:
        print("Le client doit ajouter au moins :"+str(montant_a_rendre*(-1)))
        return dico_monnaie

    #Liste des différentes valeurs
    valeurs=[20,10,5,2,1,0.5,0.2,0.1,0.05,0.02,0.01]

    #Dictionnaire contenant le type de monnaie et la valeur
    dico_valeurs={}
    dico_valeurs[20]="Billet 20 euros"
    dico_valeurs[10]="Billet 10 euros"
    dico_valeurs[5]="Billet 5 euros"
    dico_valeurs[2]="Pièce 2 euros"
    dico_valeurs[1]="Pièce 1 euros"
    dico_valeurs[0.5]="Pièce 50 centimes"
    dico_valeurs[0.2]="Pièce 20 centimes"
    dico_valeurs[0.1]="Pièce 10 centimes"
    dico_valeurs[0.05]="Pièce 5 centimes"
    dico_valeurs[0.02]="Pièce 2 centimes"
```

```
dico_valeurs[0.01]="Pièce 1 centime"

decompte=montant_a_rendre
for i in range(0,len(valeurs)):
    nb_billet=nb_billets(decompte,valeurs[i])
    print(valeurs[i],nb_billet)
    if nb_billets!=0:
        dico_monnaie[dico_valeurs[valeurs[i]]]=nb_billet
    decompte = decompte-nb_billet*valeurs[i]
    print(montant_a_rendre,decompte)

for elements in dico_monnaie:
    if dico_monnaie[elements]!=0:
        print(elements+" : "+str(dico_monnaie[elements]))
return dico_monnaie

def nb_billets(montant,valeur):
    """
    Calcule le nombre de billets ou de pièce à rendre pour un
    montant donné
    Keywords arguments :
    montant : montant à payer
    valeur : valeur de la pièce ou du billet
    """
    return montant//valeur

def rendre_monnaie_entier(cout,somme_client):
    """
    On travaille ici avec des nombres entiers

    Retourne un dictionnaire contenant le type de billets
    ou pièces à rendre ainsi que le nombre de chacun d'entre eux

    Keywords arguments :
    cout : somme à payer
    somme_client : argent donné par le client
    """
    dico_monnaie={}
    montant_a_rendre = int(100*somme_client) - int(100*cout)
    if montant_a_rendre < 0:
        print("Le client doit ajouter au moins :"+str(montant_a_rendre*(-1)))
        return dico_monnaie

    #Liste des différentes valeurs
    valeurs=[2000,1000,500,200,100,50,20,10,5,2,1]

    #Ditionnaire contenant le type de monnaie et la valeur
    dico_valeurs={}
    dico_valeurs[2000]="Billet 20 euros"
    dico_valeurs[1000]="Billet 10 euros"
    dico_valeurs[500]="Billet 5 euros"
    dico_valeurs[200]="Pièce 2 euros"
    dico_valeurs[100]="Pièce 1 euros"
    dico_valeurs[50]="Pièce 50 centimes"
    dico_valeurs[20]="Pièce 20 centimes"
    dico_valeurs[10]="Pièce 10 centimes"
    dico_valeurs[5]="Pièce 5 centimes"
    dico_valeurs[2]="Pièce 2 centimes"
    dico_valeurs[1]="Pièce 1 centime"
```

```
decompte=montant_a_rendre
for i in range(0,len(valeurs)):
    nb_billet=nb_billets(decompte,valeurs[i])
    #print(valeurs[i],nb_billet)
    if nb_billets!=0:
        dico_monnaie[dico_valeurs[valeurs[i]]]=nb_billet
    decompte = decompte-nb_billet*valeurs[i]
    #print(montant_a_rendre,decompte)

for elements in dico_monnaie:
    if dico_monnaie[elements]!=0:
        print(elements+" : "+str(dico_monnaie[elements]))
return dico_monnaie

rendre_monnaie(23.32,30)
print()
rendre_monnaie_entier(23.32,30)
```