

Nom : ..... Classe : .....

## PARTIE 1: NOTION DE PROCÉDURE

(SANS ORDINATEUR)

**Pb** 1. PROBLÉMATIQUE ET NOTION DE PROCÉDURE

Le programme suivant utilise 7 fois une structure de code similaire.

```
print("lundi est un jour de la semaine. Bonne journée !!!")
print("mardi est un jour de la semaine. Bonne journée !!!")
print("mercredi est un jour de la semaine. Bonne journée !!!")
print("jeudi est un jour de la semaine. Bonne journée !!!")
print("vendredi est un jour de la semaine. Bonne journée !!!")
print("samedi est un jour de la semaine. Bonne journée !!!")
print("dimanche est un jour de la semaine. Bonne journée !!!")
```

C'est quand même un peu idiot de taper 7 fois la même ligne. Peut-on simplifier ce code ?

Il arrive qu'un programme ait besoin d'utiliser plusieurs fois le même morceau de code. Il serait fastidieux de devoir le recopier plusieurs fois à l'identique.

Pour simplifier la tâche, on écrit ce que l'on appelle une **procédure**. Elle contient le morceau de code en question. Une fois la procédure définie, on l'appellera autant de fois que nécessaire dans le programme.

Pour alléger le code précédent, on va créer un bloc d'instruction appelé **une procédure**.

- Étape 1 : on **définit** notre procédure nommée par exemple **phrase()** qui a un argument : la chaîne de caractères *jour* (le jour de la semaine)

```
def phrase(jour):
    print(jour, "est un jour de la semaine. Bonne journée !!!")
```

- Étape 2 : on **appelle** ensuite cette procédure autant de fois que nécessaire afin d'afficher les phrases correspondantes.

```
phrase("lundi")
phrase("mardi")
phrase("mercredi")
phrase("jeudi")
phrase("vendredi")
phrase("samedi")
phrase("dimanche")
```

→

```
lundi est un jour de la semaine. Bonne journée !!!
mardi est un jour de la semaine. Bonne journée !!!
mercredi est un jour de la semaine. Bonne journée !!!
jeudi est un jour de la semaine. Bonne journée !!!
vendredi est un jour de la semaine. Bonne journée !!!
samedi est un jour de la semaine. Bonne journée !!!
dimanche est un jour de la semaine. Bonne journée !!!
```

**!** 2. SYNTAXE DE CRÉATION D'UNE PROCÉDURE EN PYTHON

Une procédure se définit ainsi :

```
def nom_procedure(argument1, argument2, ...):
    instruction 1
    instruction 2
    etc...
```

Ex

### 3. EXERCICE (COMPLÉTER LES POINTILLÉS CI-DESSOUS)

Bob a écrit le programme suivant :

```
print("Bonjour Sheldon")
print("Quel temps fait-il à Londres ?")

print("Bonjour Penny")
print("Quel temps fait-il à Paris ?")

print("Bonjour Leonard")
print("Quel temps fait-il à Rome ?")

print("Bonjour Bernadette")
print("Quel temps fait-il à Berlin ?")
```

Alice a remarqué que le programme précédent pouvait être allégé à l'aide d'une procédure. Elle propose à Bob :

- De commencer par définir une procédure nommée par exemple **salutations()**

```
def salutations(prenom, ville) :

    print(.....)

    print(.....)
```

- Puis d'appeler cette procédure 4 fois à la suite du programme.

```
salutations(.....)

salutations(.....)

salutations(.....)

salutations(.....)
```

Pb

### 4. PROBLÈME

Comment écrire un programme Python qui affiche le texte suivant ?

10 articles à 2 euros me coutent : 20 euros  
 5 articles à 3 euros me coutent : 15 euros  
 5 articles à 4 euros me coutent : 20 euros  
 100 articles à 5 euros me coutent : 500 euros

La procédure à définir est par exemple :

```
def achat(n, prix) :

    ....

    ....
```

Les 4 appels de cette procédure sont :

```
.....

.....

.....

.....
```



**E** 1. EXERCICE

1. Saisir le code suivant et écrire ci-contre les affichages qu'il produit à l'exécution.

```
def affichemoyenne(a,b,c) :
    m = (a+b+c)/3
    print("moyenne =",m)

affichemoyenne(8,12,19)
affichemoyenne(14,12,13)
affichemoyenne(5,15,10)
```

Voici ce qu'il s'affiche sur l'écran :

2. Quel est l'intérêt, selon vous, d'avoir défini la procédure **affichemoyenne()** ?

.....  
 .....

3. Modifier la procédure **affichemoyenne()** pour qu'elle puisse afficher la moyenne de 4 notes a,b,c et d.

```
def affichemoyenne(.....) :

    m = .....

    print("moyenne =",m)

affichemoyenne(8,15,13,10)
affichemoyenne(10,7,13,15)
affichemoyenne(20,20,19,18)
```

4. Saisir vos modifications sur l'ordinateur et exécuter votre programme.

**Pb** 2. PROBLÈME

1. Un élève a défini une certaine procédure nommée **renseignements()** puis il l'a appelée 3 fois dans son programme afin d'obtenir les affichages suivants :

renseignements("Hisoka", 10)	→	Bonjour <b>Hisoka</b> Tu as <b>10</b> ans. Par conséquent, tu es né.e en 2012
renseignements("Kurapika", 15)	→	Bonjour <b>Kurapika</b> Tu as <b>15</b> ans. Par conséquent, tu es né.e en 2007
renseignements("Léolio", 20)	→	..... ..... .....

2. Compléter les pointillés ci-dessus.  
 3. Compléter ci-dessous, le code de cette procédure.

```
def renseignements(prenom, age) :

    print (.....)

    print (.....)

    anneenaissance = .....

    print("par conséquent, tu es né.e en ", anneenaissance)
```

**APPEL**

→ Appeler le professeur pour vérification



- Se connecter à l'ordinateur.
- Créer un dossier nommé **PythonProcedures**
- Lancer le logiciel **Thonny**

### Pb 1. MAJEUR/MINEUR

Compléter ci-dessous la procédure **majeur(age)** qui affiche si une personne est majeure ou pas

```
def majeur (age) :
    if age ..... :
        .....
    else :
        .....
```

Pour tester la procédure, saisir dans la console :

```
>>> majeur (18)
>>> majeur (10)
```

DEPÔT 1

déposer majeur.py sur <https://entraide-ella.fr>

### Pb 2. SOMME DES ENTIERS

Compléter ci-dessous une procédure **somme(n)** qui affiche la somme des entiers de 1 à n

```
def somme (n) :
    s = 0
    k = 1
    while k <= n:
        s = ....
        k = ...
    print ("la somme est :", s)
```

Pour tester la procédure, saisir dans la console :

```
>>> somme (100)
```

DEPÔT 2

déposer somme.py sur <https://entraide-ella.fr>

### Pb 3. PÉRIMÈTRE D'UN RECTANGLE

Compléter ci-dessous un procédure **périmètre(x,y)** affiche le périmètre d'un rectangles de côtés x et y

```
def périmètre (x,y) :
    p = ...
    print ("le périmètre est ", p)
```

Pour tester la procédure, saisir dans la console :

```
>>> périmètre (30, 50)
```

DEPÔT 3

déposer perimetre.py sur <https://entraide-ella.fr>