


<p>TP</p> <p>TSIN</p>	<p>Programmation en langage structuré</p> <p>TP1-2 : Python – Initiation (les boucles)</p>	
-----------------------	--	--

1 Les boucles

1.1 Les boucles itératives

Les boucles permettent de répéter des instructions autant de fois que l'on souhaite. Lorsqu'on connaît par avance le nombre de fois que l'on veut répéter les instructions, on utilise une boucle for.

```

1 for element in ensemble_a_parcourir:
2     <instructions de la boucle>
3 <suite du programme>

```

1.2 Exercices boucle For :

1. Ecrire, tester puis commenter dans un programme les syntaxes suivantes

```

# Les boucles

# Les boucles iteratives
for voyelle in ['a','e','i','o','u','y']:
    print (voyelle)

for n in range (10):
    print( n)

for n in range(2,7):
    print(n)

for pair in range(100,110,2):
    print(pair)

```

2. Ecrire, tester et dire ce que fait cette suite d'instructions ?

```

s=0
for i in range(5):
    x=int(input('Entrez un nombre : '))
    if x==0:
        break
    s = s+x
    print(s)

```

3. Empilement d'oranges

Afin de faire une promotion sur les oranges, un supermarché décide de les présenter en faisant un empilement esthétique sous la forme d'une pyramide à base carré.

Au sommet (étage 1), il y a une orange, au-dessous (étage 2), il y a 4 oranges, au-dessous encore (étage 3) il y a 9 oranges, etc.

Compléter le script suivant afin qu'il calcule le nombre d'oranges nécessaires pour une pyramide de n étages

1. Combien d'oranges y a-t-il à l'étage i ?
2. Combien d'oranges sont nécessaires pour une pyramide de 3 étages ?
3. Compléter le script suivant afin qu'il calcule le nombre d'oranges nécessaires pour une pyramide de n étages.

```
1 n = int(input(" Entrer n : "))
2 nbre_oranges = 0
3 for i in range(.....) :
4     nbre_oranges = .....
5 Afficher (.....)
```

4. Combien d'oranges faut-il pour construire une pyramide de 10 étages ? et de 20 étages ?

4. Périmètre d'un polygone

Ecrire un programme qui calcule le périmètre d'un polygone à n côtés.

Le programme demande la valeur de n puis les longueurs de chaque côté pour afficher le périmètre

1.3 Les boucles conditionnelles

- 1.4 Lorsque le nombre de tours de boucle dépend d'une condition, on utilise l'instruction **while**.

```
1 while <condition>:
2     <instructions de la boucle>
3 <suite du programme>
```

1.5 Exercices boucle While :

1. Evolution de la valeur d'une voiture

Au 1^{er} janvier 2018, le prix d'une voiture neuve était de 20 000 €. Chaque année, la valeur de la voiture diminue de 20%. On souhaite écrire un algorithme qui calcule le nombre d'années au bout duquel la valeur de cette voiture passe sous le seuil de 2 000€.

1. Quelle est la valeur de cette voiture au bout de 2 ans ?
2. Compléter le programme pour qu'il calcule le nombre d'années voulu. On note V la valeur de la voiture et N le nombre d'années.

```
1  V = .....
2  N = .....
3  while ..... :
4      V = .....
5      N = .....
6  print(.....)
```

3. Tester le programme. Quelle valeur obtient-on ?
4. Modifier le programme pour trouver au bout de combien d'années la valeur de la voiture a été divisée par 2
5. Au bout de combien d'années la valeur de la voiture est divisée par 2 ?

2. Caisse enregistreuse

On souhaite calculer le total journalier d'une caisse enregistreuse

- On utilisera deux variables :
valeur pour les montants enregistrés
somme pour le total
- Et une boucle infinie

```
while True: # boucle infinie
    ... # instructions à exécuter
    if valeur==0: # condition de sortie de la boucle
        break    # sortie de la boucle
```

3. Somme des n entiers :

En utilisant une boucle While ; écrire un programme qui calcule la somme des n premiers entiers naturels

1.6 Exercices de synthèse

En choisissant la syntaxe la plus appropriée, ...

1. Ecrire un programme qui saisit un entier ≥ 1 et qui affiche la factorielle
$$n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$$
2. Ecrire un programme qui affiche un tableau de valeurs pour la fonction f définie par $f(x) = x^2 - 2x - 2$.
L'utilisateur choisit les bornes de l'intervalle sur lequel on calcule ces valeurs, ainsi que le pas entre deux valeurs.
3. Ecrire un algorithme qui détermine le PGCD (plus grand diviseur commun) de deux nombres A et B entiers positifs
→ Par l'algorithme d'Euclide qui utilise le reste de la division A/B :
soit $Q=A/B$ et R le reste de A/B
si $R=0$ alors le PGCD est B
sinon on remplace A par B, B par R et on recommence.
4. Ecrire un programme qui demande un nombre entier à l'utilisateur. Si la valeur est comprise entre 10 et 20, on sort de la boucle ; sinon la demande est réitérée.