

TP noté 3 – Python (10 points)

Parties 4, 5 et 6

Partie 4 : Les conditions

Les **conditions** permettent d'exécuter différentes instructions en fonction de valeurs initiales :

- **Si** la condition est remplie, **alors** on fait telle(s) instruction(s)
- **Sinon** (la condition n'est pas remplie), **alors** on fait telle(s) autre(s) instruction(s)

	
Blocs : 	Programme : <pre data-bbox="636 746 960 927"> ma_variable = 10 if ma_variable > 50 : print("YES") else: print("Oh no...") </pre>

/!\ En Python, il ne faut **pas** oublier :

- Les **indentations** (décalages) des instructions sous les `if` et `else` (avec la touche « tab » à gauche du clavier).

Pas d'indentations = en dehors des conditions



- Les **deux points** « : » à la fin de la ligne du `if` et du `else`.

Exercice 4 :

- a. Dans Spyder, créer un nouveau fichier python.
Enregistrer-le dans vos Documents en nommant le fichier « exercice4 ».

- b. Ecrire le programme suivant dans Spyder :

```
jour = ...  
mois = ...  
annee = ...  
  
jour2 = ---  
mois2 = ---  
annee2 = ---
```

en remplaçant les ... par des entiers (int) représentant votre date de naissance et
en remplaçant les --- par des entiers (int) représentant la date d'aujourd'hui.

- c. Noter ce programme.
- d. Ajouter ce programme à la suite du programme précédent :

```
if jour == jour2 and mois == mois2 :  
    print("Je te souhaite une bonne journée !")  
  
a = annee2 - annee  
  
if mois > mois2 :  
    a = a - 1  
else:  
    if mois == mois2 and jour > jour2 :  
        a = a - 1  
print(a)
```

- e. Noter ce qu'affiche votre programme.
- f. Expliquer ce que fait le programme.
- g. Quelle date de naissance doit-on écrire dans le programme pour obtenir l'affichage suivant ?

Je te souhaite une bonne journée !

Partie 5 : Les boucles

Les **boucles** permettent d'exécuter plusieurs fois une (ou plusieurs) instructions plusieurs fois.

	
Blocs : 	Programme : <pre>ma_variable = 0 for i in range(5): ma_variable = ma_variable + 10 print(ma_variable)</pre>

/! En Python, il ne faut **pas** oublier :

- Les **indentations** (décalages) des instructions sous les `for` (avec la touche « tab » à gauche du clavier).

Pas d'indentations = en dehors de la boucle



- Les **deux points** « : » à la fin de la ligne du `for`.

Exercice 5 :

- a. Dans Spyder, créer un nouveau fichier python.
Enregistrer-le dans vos Documents en nommant le fichier « exercice_5_6 ».

- b. Ecrire le programme suivant dans Spyder :

```
texte = "bonjour"  
nombre = 0  
compter = 0  
  
for i in range(5):  
    nombre = nombre + 10  
    compter = compter + 1  
    print(i)  
    print(texte)  
print("Fin")  
print(nombre)  
print(compter)
```

- c. Quelle est la valeur de la variable `nombre` à la fin du programme ?
- d. Expliquer pourquoi.
- e. Quelle est la valeur de la variable `compter` à la fin du programme ?
- f. Expliquer pourquoi.
- g. Combien de fois "bonjour" s'affiche-t-il ?
- h. Expliquer pourquoi.
- i. Combien de fois "Fin" s'affiche-t-il ?
- j. Expliquer pourquoi.
- k. A votre avis, pourquoi la variable `i` n'affiche pas la même valeur à chaque tour de boucle ?

I. Modifier le programme pour que :

- "bonjour" et "Fin" s'affichent 10 fois ;
- nombre contienne la valeur 60 à la fin du programme ;
- compter contienne la valeur 20 à la fin du programme.

Partie 6 : Les listes

Une liste est une suite de valeurs mises entre crochets [et].

Exemple : `ma_liste = [5, 9, 12, 3, 4, 0, 78, 5]`

L'instruction `len(ma_liste)` donne le nombre de valeurs dans la liste.

Pour accéder à une valeur de la liste, on utilise son indice :

- La première valeur d'une liste est à l'indice : 0
- La dernière à l'indice d'une liste : `len(ma_liste) - 1`

Exemple : `ma_liste[0]` permet de récupérer la valeur 5

`ma_liste[6]` permet de récupérer la valeur 78

On utilise généralement une boucle `for` pour parcourir chaque élément d'une liste.

Exemple :

```
t = len(ma_liste)           <= taille de la liste
```

```
for i in range( t ) :
```

```
    print( ma_liste[i] )    <= Affichage de chaque élément
```

Exercice 6 :

- a. Effacer le programme du fichier « exercice_5_6 », puis écrire le programme suivant :

```
liste = ["Marie", "Magalie", "Morgan", "Mohammed",  
        "Marc", "Maxence"]  
  
t = len(liste)  
print(t)  
  
for i in range(t):  
    eleve = liste[i]  
    print(eleve)
```

- b. Expliquer ce qu'affiche le programme.

Il est possible de mettre des listes dans une liste, ce qui donne une matrice.

Exemple : `matrice = [ [5, 9, 7, 1] , [3, 4, 0, 2] ]`

`matrice[0]` permet de récupérer la liste `[5, 9, 7, 1]`

`matrice[1]` permet de récupérer la liste `[3, 4, 0, 2]`

`matrice[1][0]` permet de récupérer la valeur `3`

`matrice[1][3]` permet de récupérer la valeur `2`

- c. Modifier la liste comme ci-dessous :

```
liste = [{"Marie", 15, "rouge"}, {"Magalie", 16, "bleu"},  
        {"Morgan", 16, "vert"}, {"Mohammed", 15, "bleu"},  
        {"Marc", 14, "bleu"}, {"Maxence", 15, "orange"}]
```

- d. Expliquer ce qu'affiche le programme.

e. Modifier le programme comme ci-dessous :

```
liste = [["Marie", 15, "rouge"], ["Magalie", 16, "bleu"],
        ["Morgan", 16, "vert"], ["Mohammed", 15, "bleu"],
        ["Marc", 14, "bleu"], ["Maxence", 15, "orange"]]

t = len(liste)

compteurA = 0
compteurB = 0
compteurC = 0

for i in range(t):
    eleve = liste[i]
    if eleve[1] > 14 :
        compteurA = compteurA + 1
    if eleve[2] == "bleu" :
        compteurB = compteurB + 1
    else:
        compteurC = compteurC + 1

print(compteurA)
print(compteurB)
print(compteurC)
```

f. Expliquer ce qu'affiche le programme.

QCM

Répondre au questionnaire sur Ecole Directe avec vos réponses aux questions.

Attention : Quand vous validez une réponse, vous ne pouvez pas revenir en arrière !

Si tu as tout fini :

Commencer ou continuer : snt-nsi.fr/game/algopower ou snt-nsi.fr/game/cyber_enquetes