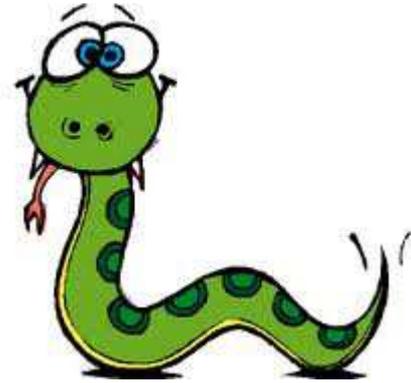


PROGRAMMER EN LANGAGE INTERPRÉTÉ ORIENTÉ OBJET AVEC « PYTHON » (LES BOUCLES)



python



Objectifs de l'activité pratique :

Les boucles :

- instructions : while, for, break
- fonction : range()

QCM

Exercices d'application

Support d'activité :

Logiciels : Portable python 2.7, Microsoft Office 2003, Libre Office et Firefox

Fichier : QCM et exo sur Python-Les boucles.htm

Ce document au format « pdf »

DOCUMENT GUIDE

IL VOUS APPARTIENT DE RÉDIGER VOTRE COMPTE-RENDU

LES BOUCLES

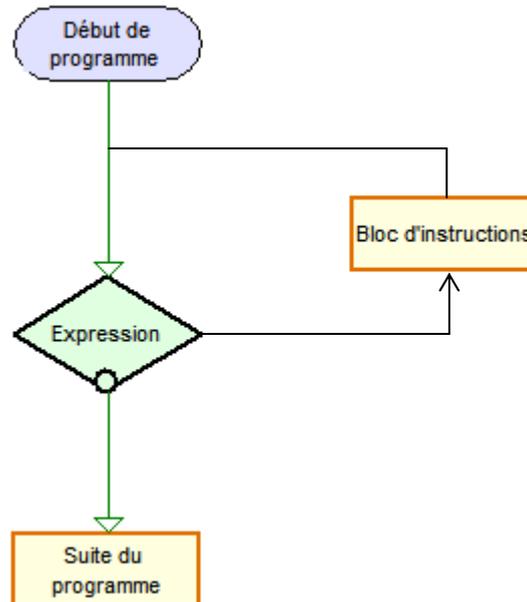
Lancer Pyscripter et ouvrir un nouveau script (**File/New Window** ou **CTRL+N**)

Remarque :

Penser à sauvegarder vos scripts dans votre dossier personnel et sur votre clé USB.

L'INSTRUCTION « while »

Une boucle permet d'exécuter une portion de code plusieurs fois de suite.

**Syntaxe :**

```
while expression:           # ne pas oublier le signe de ponctuation ':'  
    bloc d'instructions     # attention à l'indentation  
  
# suite du programme
```

Si l'expression est vraie (True) le bloc d'instructions est exécuté, puis l'expression est à nouveau évaluée.

Le cycle continu jusqu'à ce que l'expression soit fausse (False) : on passe alors à la suite du programme.

Question 1 :

Vous allez commencer par créer le script « **Boucle1.py** ».

Entrer puis tester le code source ci-dessous :

```
# script Boucle1.py  
  
# initialisation de la variable de comptage  
compteur = 1  
while compteur<5:  
    # ce bloc est exécuté tant que la condition (compteur<5) est vraie  
    print(compteur, compteur<5)  
    compteur += 1 # incrémentation du compteur, compteur = compteur + 1
```

```
print(compteur<5)
print("Fin de la boucle")
```

Question 2 :

Réaliser le script permettant l'affichage de la table de multiplication par 8.

Exemple de résultat :

```
>>>
1 * 8 = 8
2 * 8 = 16
3 * 8 = 24
4 * 8 = 32
5 * 8 = 40
6 * 8 = 48
7 * 8 = 56
8 * 8 = 64
9 * 8 = 72
10 * 8 = 80
Et voilà !
```

Question 3 :

Entrer puis tester le code source ci-dessous permettant l'affichage de l'heure courante :

```
# script Boucle3.py

import time    # importation du module time
quitter = 'n'  # initialisation
while quitter != 'o':
    # ce bloc est exécuté tant que la condition est vraie
    # strftime() est une fonction du module time
    print 'Heure courante ',time.strftime('%H:%M:%S')
    quitter = raw_input("Voulez-vous quitter le programme (o/n) ? ")
print "A bientôt"
```

L'INSTRUCTION « for »**Syntaxe :**

```
for élément in séquence:
    bloc d'instructions
# suite du programme
```

Les éléments de la séquence sont issus d'une chaîne de caractères ou bien d'une liste.

Question 4 :

Tester le script ci-dessous (séquence de caractères) :

```
# script Boucle4.py
```

```
chaine = 'Bonsoir'
for lettre in chaine:
    print lettre
print "Fin de la boucle"
```

lettre est la variable d'itération

La variable lettre est initialisée avec le premier élément de la séquence ('B').

Le bloc d'instructions est alors exécuté.

Puis la variable lettre est mise à jour avec le second élément de la séquence ('o') et le bloc d'instructions à nouveau exécuté...

Le bloc d'instructions est exécuté une dernière fois lorsqu'on arrive au dernier élément de la séquence ('r').

Question 5 :

Tester le script ci-dessous (éléments d'une liste) :

```
# script Boucle5.py
```

```
maliste = ['Pierre',67.5,18]
for element in maliste:
    print element
print "Fin de la boucle"
```

Ici, on affiche dans l'ordre les éléments de la liste.

FONCTION « range() »

L'association avec la fonction range() est très utile pour créer des séquences automatiques de nombres entiers.

Question 6 :

Tester le script ci-dessous :

```
# script Boucle6.py
print range(1,5)

for i in range(1,5):
    print i
print "Fin de la boucle"
```

Question 7 :

Réaliser le script permettant l'affichage de la table de multiplication par 9 avec une boucle for.

Exemple de résultat :

```
>>>  
1 * 9 = 9  
2 * 9 = 18  
.....  
.....  
10 * 9 = 90  
Et voilà !
```

L'INSTRUCTION « break »

L'instruction break provoque une sortie immédiate d'une boucle while ou d'une boucle for.

Dans l'exemple suivant, l'expression True est toujours ... vraie : on a une boucle sans fin.
L'instruction break est donc le seul moyen de sortir de la boucle.

Question 8 :

Tester le script ci-dessous :

```
# script Boucle8.py  
  
import time # importation du module time  
while True:  
    # strftime() est une fonction du module time  
    print 'Heure courante ',time.strftime('%H:%M:%S')  
    quitter = raw_input("Voulez-vous quitter le programme (o/n) ? ")  
    if quitter == 'o':  
        break  
print "À bientôt"
```

Remarque :

Si vous connaissez le nombre de boucles à effectuer, utiliser une boucle for.
Autrement, utiliser une boucle while (notamment pour faire des boucles sans fin).

QCM ET EXERCICES D'APPLICATION

Copier puis coller le fichier « **QCM et exo sur Python-Les boucles.htm** » dans votre dossier personnel.

Ouvrir le fichier « **QCM et exo sur Python-Les boucles.htm** » à l'aide de « **FireFox** ».

Question :

Compléter le formulaire puis sauvegarder votre travail sous le nom : « **AP_Python3_VOS NOMS.htm** » dans votre dossier personnel puis glisser une copie du fichier dans votre groupe de partage.