

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



# python



## Objectifs de l'activité pratique :

Les conditions :

- instructions if, else et elif

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 conditions.htm

Ce document au format « pdf »

## DOCUMENT GUIDE

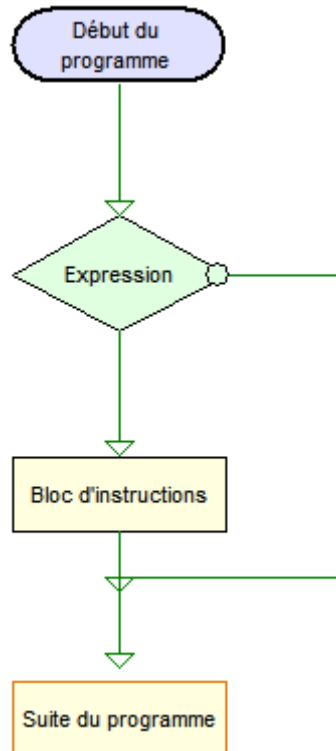
**IL VOUS APPARTIENT DE RÉDIGER VOTRE COMPTE-RENDU**

## LES CONDITIONS

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 « if »****Syntaxe :**

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

Si l'expression est vraie (True) alors le bloc d'instructions est exécuté.  
Si l'expression est fausse (False) on passe directement à la suite du programme.

**Question 1 :**

Vous allez commencer par créer le script « **Condition1.py** ».

Entrer puis tester le code source ci-dessous :

```
# script Condition1.py  
  
chaine = input("Note sur 20 : ")  
note = float(chaine)  
if note >= 10.0:  
    # ce bloc est exécuté si l'expression (note >= 10.0) est vraie  
    print("J'ai la moyenne")  
print("Fin du programme")
```

**L'INSTRUCTION « else »**

Une instruction else est toujours associée à une instruction if.

**Syntaxe :**

```
if expression:
    bloc d'instructions 1      # attention à l'indentation
else:
    bloc d'instructions 2      # else est au même niveau que if
    # suite du programme      # attention à l'indentation
```

Si l'expression est vraie (True) alors le bloc d'instructions 1 est exécuté.

Si l'expression est fausse (False) alors c'est le bloc d'instructions 2 qui est exécuté.

**Question 2 :**

*Entrer puis tester le code source ci-dessous :*

```
# script Condition2.py

chaine = input("Note sur 20 : ")
note = float(chaine)
if note >= 10.0:
    # ce bloc est exécuté si l'expression (note >= 10.0) est vraie
    print("J'ai la moyenne")
else:
    # ce bloc est exécuté si l'expression (note >= 10.0) est fausse
    print("C'est en dessous de la moyenne")
print("Fin du programme")
```

**Remarque :**

Pour traiter le cas des notes invalides (<0 ou >20), on peut imbriquer des instructions conditionnelles.

**Question 3 :**

*Modifier puis tester le code source précédent comme ci-après :*

```
# script Condition3.py

chaine = input("Note sur 20 : ")
note = float(chaine)
if note > 20.0 or note < 0.0:
    # ce bloc est exécuté si l'expression (note > 20.0 or note < 0.0) est vraie
    print("Note invalide !")
else:
    # ce bloc est exécuté si l'expression (note > 20.0 or note < 0.0) est fausse
    if note >= 10.0:
        # ce bloc est exécuté si l'expression (note >= 10.0) est vraie
```

```
print("J'ai la moyenne")
else:
    # ce bloc est exécuté si l'expression (note>=10.0) est fausse
    print("C'est en dessous de la moyenne")
print("Fin du programme")
```

#### Question 4 :

Modifier de nouveau le script précédent en ajoutant un niveau d'imbrication pour traiter les cas particuliers « 0 » et « 20 ».

Lorsque la note = 20 alors on affiche : « **C'est excellent** », lorsque la note = 0 alors on affiche : « **C'est lamentable !** »

### L'INSTRUCTION « elif »

Une instruction elif (contraction de else if) est toujours associée à une instruction if.

#### Syntaxe :

```
if expression 1:
    bloc d'instructions 1
elif expression 2:
    bloc d'instructions 2
elif expression 3:
    bloc d'instructions 3# ici deux instructions elif, mais il n'y a pas de limitation
else:
    bloc d'instructions 4
# suite du programme
```

Si l'expression 1 est vraie alors le bloc d'instructions 1 est exécuté, et on passe à la suite du programme.  
Si l'expression 1 est fausse alors on teste l'expression 2 :

si l'expression 2 est vraie on exécute le bloc d'instructions 2, et on passe à la suite du programme.  
si l'expression 2 est fausse alors on teste l'expression 3, etc...

Le bloc d'instructions 4 est donc exécuté si toutes les expressions sont fausses (c'est le bloc "par défaut").

Parfois il n'y a rien à faire.

Dans ce cas, on peut omettre l'instruction else :

```
if expression 1:
    bloc d'instructions 1
elif expression 2:
    bloc d'instructions 2
elif expression 3:
    bloc d'instructions 3
# suite du programme
```

L'instruction elif évite souvent l'utilisation de conditions imbriquées (et souvent compliquées).

#### Question 5 :

Tester le script ci-dessous :

```
# script Condition5.py
# ce script fait la même chose que Condition4.py

note = float(input("Note sur 20 : "))
if note==0.0:
    print("C'est en dessous de la moyenne")
    print(" C'est lamentable !")
elif note==20.0:
    print("J'ai la moyenne")
    print("C'est excellent !")
elif note<10.0 and note>0.0:# ou bien : elif 0.0 < note < 10.0:
    print("C'est en dessous de la moyenne")
elif note>=10.0 and note<20.0:# ou bien : elif 10.0 <= note < 20.0:
    print("J'ai la moyenne")
else:
    print("Note invalide !")
print("Fin du programme")
```

## QCM ET EXERCICES D'APPLICATION

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

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

### Question :

Compléter le formulaire puis sauvegarder votre travail sous le nom : « **AP\_Python2\_VOS NOMS.htm** » dans votre dossier personnel puis glisser une copie du fichier dans votre groupe de partage.