➢ Tests

if … : Permet de tester une condition (ou plusieurs avec « *and* », « *or* » ou « *not* »)
si la condition est vérifiée (*True*), on traite les instructions situées après la tabulation

Exemple :
```
if n != 0 :
    print(n, "est non nul")
```

else … : Permet de traiter le cas où la condition précédentes (ou les conditions) n'est pas vérifiée
on exécute alors les instructions situées après la tabulation

Exemple :
```
if n%2 == 0 :
    print(n, "est pair")
else :
    print(n, "est impair")
```

elif … : Permet de condenser un « *else* » avec un « *if* »

Remarque : On peut imbriquer plusieurs tests les uns dans les autres, attention à bien identifier les blocs avec les tabulations.

Complément pour les tests :

pass Permet de passer si le résultat d'un test est vérifié mais que l'on n'a pas d'instruction.

➢ Boucles inconditionnelles

for … *in* … : Permet de répéter les instructions situées après la tabulation pendant qu'une variable décrit un ensemble de valeurs prises dans une chaîne de caractère, une liste, un tuple ou un ensemble

Exemple :
```
for i in (1, 2, 3) :
    print(i)
```

Exemple :
```
for c in "bonjour" :
    print(c)
```

Remarque : L'instruction « *range* » permet de créer une liste virtuelle de nombres à utiliser dans la boucle *range*(n) crée une liste virtuelle de n valeurs allant de 0 à n-1, *range*(a, b) une liste virtuelle de a à b-1,

Exemple :
```
for i in range(5) :
    print(i)
```

➢ Boucles conditionnelles

while … : Permet de répéter les instructions situées après la tabulation tant que la condition indiquée est vérifiée.

Exemple :
```
n = 0
while n <= 10 :
    print(n)
    n = n + 1
```

Attention : Il y a un risque de créer une boucle infinie si la condition n'est jamais réfutée !

Compléments pour les boucles

break Permet de sortir d'une boucle (« *for* » ou « *while* ») et de continuer la suite des instructions du programme situées après le bloc.

continue Permet de passer directement à la boucle suivante du « *for* » ou du « *while* » en ignorant les instructions situées après dans le bloc.