

06-boucles

April 4, 2019

1 Les boucles en python

(code sous licence creative commun CC BY-NC-SA BY Alexis Dendiével)

1.1 utilité des boucles

Les boucles constituent, avec les tests, un élément indispensable de programmation. Il existe deux types de boucles: - La boucle for qui se répète quand on parcourt un itérable par exemple, - la boucle while qui se répète tant qu'une condition est remplie voyons cela par l'exemple

1.2 la boucle for

```
In [3]: for i in range(10):  
        print(i)
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Dans cet exemple, nous voyons que i prend successivement les valeurs allant de 0 à 9 L'utilité peut être de faire plusieurs fois le même calcul, comme par exemple le calcul de la distance de chute.

```
In [2]: # donnée  
g = 9.81
```

```
# calcul de l'ordonnée d'un lâché vertical sans vitesse initial dans le champ de pesanteur  
for t in range(10):  
    y = 0.5*g*t**2  
    print("à l'instant ", t, ", y vaut : ", y, " m")
```

```

à l'instant 0 , y vaut : 0.0 m
à l'instant 1 , y vaut : 4.905 m
à l'instant 2 , y vaut : 19.62 m
à l'instant 3 , y vaut : 44.145 m
à l'instant 4 , y vaut : 78.48 m
à l'instant 5 , y vaut : 122.625 m
à l'instant 6 , y vaut : 176.58 m
à l'instant 7 , y vaut : 240.345 m
à l'instant 8 , y vaut : 313.92 m
à l'instant 9 , y vaut : 397.305 m

```

la boucle for peut aussi servir à parcourir un itérable, comme le montre l'exemple suivant, qui donne les n premiers éléments chimiques avec leur symbole.

```

In [4]: # liste des 18 premiers éléments chimiques de la classification
elements = (
    "Hydrogène H",
    "Hélium He",
    "Lithium Li",
    "Béryllium Be",
    "Bore B",
    "Carbone C",
    "Azote N",
    "Oxygène O",
    "Fluor F",
    "Néon Ne",
    "Sodium Na",
    "Magnésium Mg",
    "Aluminium Al",
    "Silicium Si",
    "Phosphore P",
    "Soufre S",
    "Chlore Cl",
    "Argon Ar")

# la boucle qui parcourt la liste des éléments et les imprime
for element in elements:
    print (element)

```

```

Hydrogène H
Hélium He
Lithium Li
Béryllium Be
Bore B
Carbone C
Azote N
Oxygène O

```

Fluor F
Néon Ne
Sodium Na
Magnésium Mg
Aluminium Al
Silicium Si
Phosphore P
Soufre S
Chlore Cl
Argon Ar

On peut bien sûr combiner cette boucle avec un test permettant de n'imprimer que les éléments de numéro atomique inférieur à une certaine valeur:

```
In [5]: # liste des 18 premiers éléments chimiques de la classification
elements = (
    "Hydrogène H",
    "Hélium He",
    "Lithium Li",
    "Béryllium Be",
    "Bore B",
    "Carbone C",
    "Azote N",
    "Oxygène O",
    "Fluor F",
    "Néon Ne",
    "Sodium Na",
    "Magnésium Mg",
    "Aluminium Al",
    "Silicium Si",
    "Phosphore P",
    "Soufre S",
    "Chlore Cl",
    "Argon Ar")

# demande du numéro atome maximal
Z = int(input("entrer un numéro atomique inférieur ou égal à 18: "))

# test pour vérifier que Z est inférieur ou égal à 18
if Z <= 18:
    print("les ", Z, "premiers éléments de la classification sont :")
    for i in range(Z):
        print (elements[i])
else:
    print("le numéro atomique est supérieur à 18")
```

```
entrer un numéro atomique inférieur ou égal à 18: 13
les 13 premiers éléments de la classification sont :
```

Hydrogène H
Hélium He
Lithium Li
Béryllium Be
Bore B
Carbone C
Azote N
Oxygène O
Fluor F
Néon Ne
Sodium Na
Magnésium Mg
Aluminium Al

1.3 la boucle while

C'est une autre manière de programmer une boucle: celle-ci aura lieu tant qu'une condition sera remplie. Reprenons l'exemple précédant avec une boucle while:

```
In [8]: # liste des 18 premiers éléments chimiques de la classification
elements = (
    "Hydrogène H",
    "Hélium He",
    "Lithium Li",
    "Béryllium Be",
    "Bore B",
    "Carbone C",
    "Azote N",
    "Oxygène O",
    "Fluor F",
    "Néon Ne",
    "Sodium Na",
    "Magnésium Mg",
    "Aluminium Al",
    "Silicium Si",
    "Phosphore P",
    "Soufre S",
    "Chlore Cl",
    "Argon Ar")

# demande du numéro atome maximal
Z = int(input("entrer un numéro atomique inférieur ou égal à 18: "))

# test pour vérifier que Z est inférieur ou égal à 18
if Z <= 18:
    print("les ", Z, "premiers éléments de la classification sont :")
    numero = 1
```

```

while numero <=Z:
    print (elements[numero-1])
    numero = numero + 1
else:
    print("le numéro atomique est supérieur à 18")

```

entrer un numéro atomique inférieur ou égal à 18: 12
 les 12 premiers éléments de la classification sont :
 Hydrogène H
 Hélium He
 Lithium Li
 Béryllium Be
 Bore B
 Carbone C
 Azote N
 Oxygène O
 Fluor F
 Néon Ne
 Sodium Na
 Magnésium Mg

1.4 complément

Il peut être utile de vouloir sauter une étape à l'intérieur d'une boucle avant qu'elle ne s'achève.
 On utilise pour cela: - l'instruction continue

```

In [9]: for i in range(10):
        if i ==5:
            continue
        print(i, " ")

```

0
 1
 2
 3
 4
 6
 7
 8
 9

Nous voyons ici que l'impression du 5 n'a pas été effective. Nous pouvons aussi vouloir sortir d'une boucle, pour cela: - on utilise l'instruction break

```

In [10]: for i in range(10):
        if i ==5:
            break
        print(i, " ")

```

0
1
2
3
4

Ces deux instructions, continue et break, peuvent également s'utiliser avec la boucle while.

In []: