Les boucles en python

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

voyons cela par l'exemple

La boucle for

In [1]:
```
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 laché 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 [3]:

```
# 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 parcours 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 sur combiner cette boucle avec un test permettant de n'imprimer que les éléments de numéro atomique inférieur à une certaine valeur:

In [4]:

```
def affichage_premiers_elements(Zmax):
    """
    Affiche les éléments de numéro atomique inférieur ou égal à Z_max

    :param Z_max: numéro atomique maximal (inférieur à 18)
    """
    # 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"]

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

```
    else:
        print("tsss...lisez la doc : le numéro atomique doit être inférieur à 18 !!")

affichage_premiers_elements(4)
print("")
affichage_premiers_elements(33)
```

```
les  4 premiers éléments de la classification sont :
 -  Hydrogène H
 -  Hélium He
 -  Lithium Li
 -  Béryllium Be

tsss...lisez la doc : le numéro atomique doit être inférieur à 18 !!
```

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 [5]:

```
def affichage_premiers_elements_avec_while(Zmax):
    """
    Affiche les éléments de numéro atomique inférieur ou égal à Z_max

    :param Z_max: numéro atomique maximal (inférieur à 18)
    """
    # 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"]

    # test pour vérifier que Z est inférieur ou égal à 18
    if Zmax <=18:
        print("les ", Zmax, "premiers éléments de la classification sont :")
        numero = 1
        while numero <=Zmax:
            print (" - ",elements[numero-1])
            numero = numero + 1
    else:
        print("nan, j'affiche toujours pas : le numéro atomique est supérieur à 18")

affichage_premiers_elements_avec_while(4)
print("")
affichage_premiers_elements_avec_while(32)
```

```
les  4 premiers éléments de la classification sont :
 -  Hydrogène H
 -  Hélium He
 -  Lithium Li
```

```
 -  Béryllium Be

nan, j'affiche toujours pas : le numéro atomique est supérieur à 18
```

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 [6]:

```
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 [7]:

```
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.