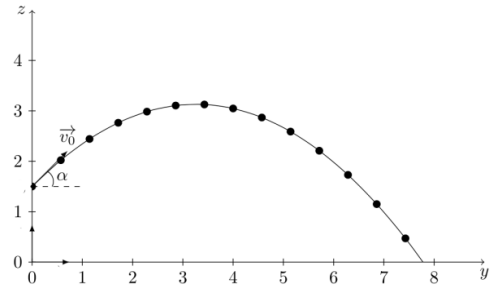


Liste à 2 dimensions

Imaginons l'enregistrement d'une trajectoire:

Les coordonnées de chacun des points peuvent être stockées dans une liste $[y, z]$.

Et l'ensemble de points dans une liste de listes: (exemple avec les 4 premiers points)



```
trajectoire = [ [0, 1.5], [0.6, 2.0], [1.2, 2.4], [1.8, 2.6] ]
```

Pour accéder au premier élément, $[0, 1.5]$, on fait:

```
trajectoire[0]
```

Instructions	sortie	Commentaires
<code>trajectoire[1]</code>		
<code>trajectoire[1][0]</code>		
<code>trajectoire[1][1]</code>		
<code>len(trajectoire)</code>		

Parcours d'une liste

Elements de liste

l'itérable est L

```
for elem in L
```

```
1. L = ['un', 'deux', 'trois']
2. for elem in L:
3.     print(elem)
4. # affiche un deux trois
```

index

L'itérable est `range(len(L))`

```
for i in range(len(L))
```

```
1. L = ['un', 'deux', 'trois']
2. for i in range(len(L)):
3.     print(i, L[i])
4. # affiche 0,un 1,deux 2,trois
```

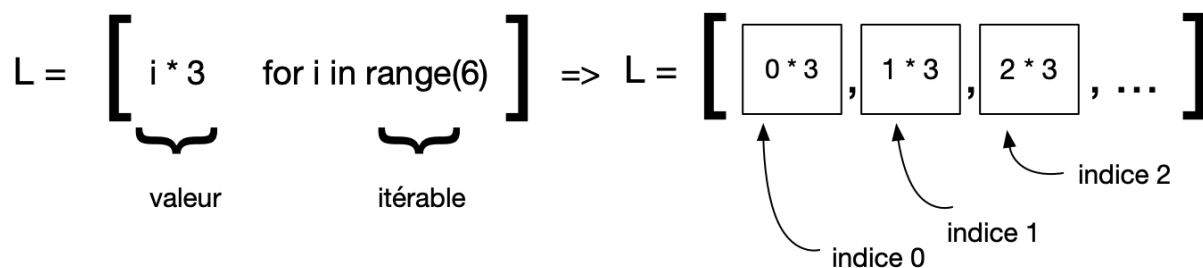
Créer une liste avec un itérable

Boucle bornée

```
1. L = []
2. for i in range(6):
3.     L.append(i * 3)
4. # L est égal à [..., ..., ..., ..., ..., ..., ...]
```

Compréhension de liste

```
1. L = [i * 3 for i in range(6)]
2. # L est égal à [..., ..., ..., ..., ..., ..., ...]
```



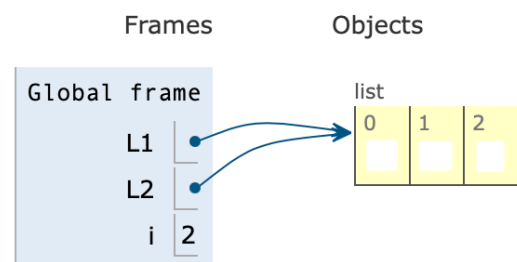
Copier une liste

Copie par référence

Lorsque l'on utilise le symbole d'affectation, la copie d'une liste se fait par REFERENCE:
 $L2 = L1$

La modification d'une liste entraîne la modification de la liste d'origine, car les 2 noms pointent vers le même objet en mémoire:

```
1. L1 = [1, 2, 3]
2. L2 = L1
3. for i in range(len(L2)):
4.     L2[i] += 1
5.
6. # L2 est égal à [..., ..., ...]
7. # L1 est égal à [..., ..., ...]
```



Copie par valeur

Pour effectuer une copie par valeur, on peut utiliser la fonction `list` :

$L2 = \text{list}(L1)$

```
1. L1 = [1, 2, 3]
2. L2 = list(L1)
3. for i in range(len(L2)):
4.     L2[i] += 1
5.
6. # L2 est égal à [..., ..., ...]
7. # L1 est égal à [..., ..., ...]
```

