Les slices en python

Complément - niveau basique

Ce support de cours reprend les notions de "slicing" vues dans la vidéo.

Nous allons illustrer les slices sur la chaine suivante; rappelez-vous toutefois que ce mécanisme fonctionne avec toutes les séquences comme les listes ou les tuples.

```
chaine = "abcdefghijklmnopqrstuvwxyz" ; print chaine
```

Slice sans pas

On a vu en cours qu'une slice permet de désigner toute une plage d'éléments d'une séquence. Ainsi on peut écrire

```
chaine[2:6]
```

Les débutants ont parfois du mal avec les bornes. Il faut se souvenir que

- les indices commencent comme toujours à zéro
- le premier indice `debut` est inclus
- le second indice `fin` est exclu
- on obtient en tout `fin-debut` items dans le résultat

Ainsi ci-dessus le résultat contient 4 éléments = 6-2

On peut omettre une borne

```
chaine[:6]
```

et utiliser des indices négatifs pour compter à partir de la fin,

```
chaine[3:-3]
```

```
chaine[-3:]
```

Slice avec pas

Il est également possible de préciser un 'pas', de façon à ne choisir, dans la plage donnée, que par exemple un élément sur deux. Par exemple:

```
chaine[3:-3:2]
```

Comme on le devine, le troisième élément de la slice, ici '2', détermine le pas. On ne retient donc, dans la chaine "def..." que "d", puis "f", et ainsi de suite.

On peut préciser du coup la borne de fin (ici -3) avec un peu de liberté puisqu'on obtiendrait un résultat identique avec -4:

```
chaine[3:-4:2]
```

Pas négatif

Il est même possible de spécifier un pas négatif. Dans ce cas, de manière un peu contre-intuitive, il faut préciser un début (le premier indice de la slice) qui soit "plus à droite" que la fin (le second indice).

Pour prendre un exemple, comme l'élément "-3" i.e. 'x' est plus à droite que l'élément "3" i.e. 'd', évidemment on obtiendrait une liste vide si on ne précisait pas le pas

```
chaine[-3:3]
```

Si maintenant on précise un pas négatif, on obtient cette fois

```
chaine[-3:3:-2]
```