

# W2-S3-E2-slicing

December 15, 2014

## 1 Séquences

### 1.1 Exercice - niveau basique

#### 1.1.1 Slicing

Commençons par créer une chaîne de caractères. Ne vous inquiétez pas si vous ne comprenez pas encore le code d'initialisation présenté ci-dessous.

Pour les plus curieux, l'instruction `import` permet de charger dans votre programme une boîte à outils que l'on appelle un module. Python vient avec de nombreux modules qui forment la librairie standard. Le plus difficile avec les modules de la librairie standard est de savoir qu'ils existent. En effet, il y en a un grand nombre et bien souvent il existe un module pour faire ce que vous souhaitez.

Pour vous familiariser avec les modules de la librairie standard et vous montrer qu'ils peuvent vous faire gagner du temps, nous en introduisons très tôt certains. N'hésitez pas à regarder l'aide Python qui décrit en détail tous les modules de la librairie standard, et en particulier le module `string` que l'on utilise ci-dessous.

```
In []: import string
      chaine = string.ascii_lowercase
      print chaine
```

Pour chacune des sous-chaînes ci-dessous, écrire une expression de slicing sur `chaine` qui renvoie la sous-chaîne. La cellule de code doit retourner `True`

Par exemple pour obtenir "def":

```
In []: chaine[3:6] == "def"
```

1) Pour obtenir "vwx" (n'hésitez pas à utiliser les indices négatifs)

```
In []: chaine[ <votre_code> ] == "vwx"
```

2) Pour obtenir "wxyz" (avec une seule constante)

```
In []: chaine[ <votre_code> ] == "wxyz"
```

3) Pour obtenir "dfhjlnprtvxz" (avec deux constantes)

```
In []: chaine[ <votre_code> ] == "dfhjlnprtvxz"
```

4) Pour obtenir "xurolifc" (avec deux constantes)

```
In []: chaine[ <votre_code> ] == "xurolifc"
```

## 1.2 Exercice - niveau intermédiaire

### 1.2.1 Longueur

On vous donne une chaîne `composite`, dont on sait qu'elle a été calculée à partir de deux chaînes `inconnue` et `connue` comme ceci:

```
composite = connue + inconnue + connue
```

L'exercice consiste à retrouver la valeur de `inconnue` à partir de celles de `composite` et `connue`.

```
In []: from corrections.w2_slicing import connue, composite
      print "connue=", connue
      print "composite=", composite
```

À vous d'écrire du code pour retrouver `inconnue` à partir de `composite` et `connue`.

```
In []: # vous pouvez bien sûr utiliser plusieurs lignes

      inconnue = "votre_code"
```

Le code de la correction recalcule une valeur de `composite` à partir de `connue` et de votre code pour `inconnue`, et compare le résultat avec la valeur cible pour `composite`.

```
In []: # correction
      from corrections.w2_slicing import exo_inconnue
      exo_inconnue.correction(inconnue)
```