

# W2-S8-C3-la-fonction-raw\_input

December 15, 2014

## 1 Obtenir une réponse de l'utilisateur

### 1.1 Complément - niveau basique

Occasionnellement, il peut être utile de poser une question à l'utilisateur.

#### 1.1.1 La fonction `raw_input()`

C'est le propos de la fonction `raw_input`. Par exemple:

```
In []: nom_ville = raw_input("entrez le nom de la ville : ")
      print "nom_ville", nom_ville
```

Vous pouvez voir ici comment cette fonction est implémentée dans un notebook.

#### 1.1.2 Utilitaire `readline`

Nous vous conseillons d'essayer également cette fonction dans un terminal avec l'interpréteur python.

Lorsqu'elle est utilisée dans un terminal, il est utile d'associer à `raw_input` le module `readline`, ainsi l'utilisateur peut utiliser les flèches pour éditer sa réponse, pour revenir en arrière ou remonter dans l'historique de ses réponses, comme ceci

```
In []: # utile seulement dans un terminal
      import readline

      entree = range(3)
      for i in range(3):
          entree[i] = int(raw_input("entrez le nombre numéro {} ".format(i)))
      print "entree", entree
```

#### 1.1.3 Attention à bien vérifier/convertir

Notez bien que `raw_input` renvoie **une chaîne**; c'est assez évident, mais il est très facile de l'oublier et de passer cette chaîne directement à une fonction qui s'attend à recevoir, par exemple, un nombre entier, auquel cas les choses se passent mal:

```
>>> range(raw_input("nombre de lignes: "))
nombre de lignes: 4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: range() integer end argument expected, got str.
```

Dans ce cas il faut appeler la fonction `int` pour convertir le résultat en un entier:

```
In []: range(int(raw_input("nombre de lignes: ")))
```

#### 1.1.4 Limitations

Cette fonction nous sera utile pour nos premiers pas en python.

En pratique toutefois, on utilise assez peu cette fonction, car les applications “réelles” viennent avec leur propre interface utilisateur, souvent graphique, et disposent donc d’autres moyens que celui-ci pour interagir avec l’utilisateur.

Les applications destinées à fonctionner dans un terminal, quant à elle, reçoivent traditionnellement leurs données de la ligne de commande. C’est le propos du module `argparse` que nous verrons ultérieurement.