

W4-S6-C1-scope-builtin

December 15, 2014

1 Le scope builtin

1.1 Complément - niveau avancé

1.1.1 Ces noms qui viennent de nulle part

Nous avons vu déjà un certain nombre de **fonctions** *built-in* comme par exemple

```
In []: open, len, zip
```

Ces noms font partie du **module** `__builtin__`. Il est cependant particulier puisque tout se passe **comme** si on avait fait avant toute chose:

```
from __builtin__ import *
```

sauf que cet import est implicite.

1.1.2 On peut réaffecter un nom *builtin*

Quoique ce soit une pratique déconseillée, il est tout à fait possible de redéfinir ces noms; on peut faire par exemple

```
In []: def open():  
      pass
```

Ceci est déconseillé car le code en aval de cette définition risque de ne pas faire ce qui est attendu; imaginez par exemple qu'on essaie maintenant de faire

```
In []: # on a modifié le nom 'open', attention aux surprises  
      with open("builtin.txt", "w") as f:  
          f.write("quelque chose")
```

1.1.3 On ne peut pas réaffecter un mot clé

À titre de digression, rappelons que les noms prédéfinis sont, à cet égard aussi, très différents des mots-clés comme `if`, `def`, `with` et autres `for` qui eux, ne peuvent pas être modifiés en aucune manière.

1.1.4 Que faire alors ?

Même quand on sait qu'il faut éviter cette pratique, on n'a pas toujours en tête la liste exhaustive des noms prédéfinis, aussi il arrive qu'on le fasse involontairement. Rappelons, cependant, qu'un bon éditeur de texte vous signalera les fonctions built-in avec une coloration syntaxique spécifique.

Sachez que vous pouvez toujours "retrouver" alors la fonction prédéfinie en l'important explicitement du module `__builtin__`. Par exemple pour écrire notre ouverture de fichier nous pouvons toujours faire

```
In []: # pour être sûr d'utiliser la bonne fonction open

import __builtin__

with __builtin__.open("builtin.txt", "w") as f:
    f.write("quelque chose")
```

1.1.5 Liste des fonctions prédéfinies

Vous pouvez trouver la liste des fonctions prédéfinies avec la fonction `dir` sur le module `__builtin__` comme ci-dessous (qui vous montre aussi les exceptions prédéfinies, qui commencent par une majuscule), ou dans la documentation sur [les fonctions prédéfinies](#)

```
In []: dir(__builtin__)
```