

Quelques bibliothèques

1 De quoi s'agit-il?

PYTHON dispose de d'un certain nombre de fonctions toutes faites, que l'on peut utiliser directement et qu'il n'est donc pas utile de programmer nous-mêmes. Ces fonctions sont rassemblées de façon thématique dans ce que l'on appelle des **bibliothèques** ou **modules**.

On a déjà rencontré la bibliothèque **math**. On indique ci-dessous quelques bibliothèques qu'il peut être utile de connaître:

nom de la bibliothèque	description brève
math	fonctions usuelles
random	nombres pseudo-aléatoires
datetime	dates et heures
decimal	nombres décimaux
numpy	tableaux
matplotlib	outils graphiques
pylab	

- le module **math** contient les fonctions les plus couramment utilisées en analyse.
- le module **random** contient des fonctions permettant de simuler des expériences aléatoires.
- le module **numpy** contient des fonctions permettant de manipuler des vecteurs et des matrices.
- le module **matplotlib** ou **pylab** permet de créer des représentations graphiques.

2 Importation

Certaines bibliothèques sont automatiquement chargées à l'ouverture de Pyzo. On peut alors utiliser directement les fonctions présentes dans ces bibliothèques.

D'autres ne le sont pas, de façon à conserver un temps de chargement correct du logiciel. La liste des bibliothèques chargées à l'ouverture peut aussi varier d'une version à l'autre. Dans ce cas, il faut D'ABORD charger la bibliothèque avant d'utiliser les fonctions qu'elle contient.

L'importation se fait avec l'instruction **import** qui peut être utilisée de différentes façons:

- `from nombibliothèque import *`

permet de charger **toutes** les fonctions présentes dans la bibliothèque.

Toutefois, cette syntaxe n'est pas recommandée car elle peut mener à des confusions si deux fonctions issues de bibliothèques différentes portent le même nom.

Par exemple,

la fonction **sin** est définie dans les modules **math** et **numpy**

la fonction **randint** est définie dans les modules **random** et **numpy**. ET ATTENTION !! Dans **random**, la borne de droite est incluse, mais dans **numpy**, elle est exclue ...

Donc pour maîtriser l'origine de l'utilisation d'une fonction, il est préférable de l'appeler en spécifiant sa bibliothèque, donc de la manière suivante :

`math.sin(x)` ou bien `numpy.sin(x)`

- `import nombibliothèque as alias`

permet d'importer la bibliothèque sous un certain alias. Si le choix de l'alias est libre, certains noms se sont imposés par l'usage: `import numpy as np` ; `import random as rd` ; etc ... Voir le mémento Python distribué au concours.

On appellera alors la fonction de cette façon pour l'utiliser : *alias.fonction*. Par exemple :

```
import numpy as np
np.sin(x)
```

- `from nombibliothèque import fonction`

permet d'importer uniquement la fonction désirée de la bibliothèque (et pas les autres). L'appel de la fonction se fait alors avec la syntaxe : `bibliothèque.fonction`. Par exemple :

```
from random import randint
random.randint(1,6)
```

3 Aide sur une bibliothèque

Pour obtenir de l'aide à propos d'une bibliothèque dont on connaît le nom, on peut utiliser la commande `help(nombibliothèque)`. Afin d'utiliser une fonction toute faite répondant aux besoins et à l'utiliser correctement, on peut écrire:

- `dir(module)`
permet d'afficher la liste des fonctions présentes dans la bibliothèque `module`
- `help(fonction)`
permet d'obtenir de l'aide sur la `fonction` (utilité, nombre de paramètres, exemples d'utilisation, ...)
- Si ces deux commandes ne vous permettent toujours pas de trouver l'information recherchée, il reste encore internet...