

Fondements informatiques I

Cours 8: Modules et fichiers

Sorina Ionica sorina.ionica@uvsq.fr

Sandrine Vial sandrine.vial@uvsq.fr

Modules

On veut programmer une application calculatrice. Comment structurer notre programme ?

- Un fichier contenant tous les fonctions permettant de faire des calculs : racines carrées, puissance, log etc.
- Un fichier contenant toutes les fonctions permettant de dessiner l'interface graphique : dessiner des buttons, des zones de texte etc.
- Un programme principale depuis lequel on appelle toutes ces fonctions

Solution

- On range les définitions de fonctions se rapportant à une même application au sein d'un fichier (script) commun baptisé **module**.
- Un module est sauvegardé sous forme d'un fichier <nom du module>.py.

Modules - Première méthode d'importation

Utiliser un module

- Pour utiliser un module, on se sert de l'instruction

```
import <nom du module>
```

- L'exécution de cette instruction consiste à exécuter le script définissant le module (ce script peut contenir des instructions autres que des définitions de fonctions).
- Pour importer un module, Python a besoin de connaître le chemin qui permet d'accéder au fichier correspondant. Ce chemin doit apparaître dans la liste des chemins possibles stockés dans la variable path du module sys.

Exemple

```
>>> import random  
>>> random.randint(0,10)  
9
```

Modules - importation d'une fonction

Dans certains cas, on a besoin de disposer d'une seule fonction du module.

```
from [module] import [fonction]
```

Exemple

```
from math import sqrt  
racine = sqrt(49)
```

Modules - importation de toutes les fonctions

Pour disposer de toutes les fonctions d'un module :

```
from [module] import *
```

Exemple

```
from math import *
racine = sqrt(49)
angle = pi/6
print sin(angle)
```

Cette méthode est à éviter, car on peut importer sans le savoir des noms de fonctions qu'on a déjà défini par ailleurs.

Modules courants

- `math` : fonctions et constantes mathématiques de base (`sin`, `cos`, `exp`, `pi`...).
- `sys` : passage d'arguments, gestion de l'entrée/sortie standard etc...
- `os` : dialogue avec le système d'exploitation.
- `random` : génération de nombres aléatoires.
- `time` : permet d'accéder aux fonctions gérant le temps.
- `calendar` : fonctions de calendrier.
- `urllib` : permet de récupérer des données sur internet depuis python.
- `Tkinter` : interface python avec Tk (permet de créer des objets graphiques ; nécessite d'installer Tk).
- `re` : gestion des expressions régulières.
- `pickle` : écriture et lecture de structures python (comme les dictionnaires par exemple).
- ...

Obtenir de l'aide sur les modules importés

- Pour obtenir de l'aide sur un module rien de plus simple, il suffit d'invoquer la commande help :

```
import random  
help(random)
```

- Il est aussi possible d'invoquer de l'aide sur une fonction particulière d'un module en la précisant de la manière suivante :

```
help(random.randint)
```