

PROGRAMMATION PYTHON

Chapitre 6: Modules



Sommaire

1. [Introduction](#)
2. [La méthode import](#)
3. [Espace de nom spécifique](#)
4. [La méthode from ... import](#)
5. [Exemple de création de module](#)
6. [Exercices](#)

Introduction

On appelle module un groupe de fonctions stockées dans un ou plusieurs fichiers par exemple les fonctions mathématiques peuvent être placées dans un module dédié aux mathématiques.



Un module en Python identifie ses fonctions par leurs signatures qui est le nom de la fonction, donc lors de la création d'un module il faut veiller à ne pas avoir deux fonctions différentes avec le même nom.

Les modules vont nous permettre de scinder l'ensemble de nos fonctions dans plusieurs fichiers, sans les modules l'on serait obligé d'écrire tout le code dans un seul fichier ce qui devient contraignant pour un gros projet.

Python propose des modules et fonctions de base comme le module **math**, **collections...** Dans un projet on aura besoin d'écrire des modules et fonctions spécifiques selon nos besoins et notre organisation d'où l'objet de ce chapitre.

La méthode import

Lorsqu'on veut utiliser un module, il faut l'importer grâce à l'instruction **import nom_du_module**. Tous les modules internes de Python comme le module **math** ne sont pas chargés par défaut quand on démarre l'interpréteur de Python. Pour utiliser les fonctions du module **math**, il faut l'importer.

```
>>> # Essaie d'avoir la documentation de math sans l'importer
>>> help(math)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    help(math)
NameError: name 'math' is not defined

>>> # Essaie après import
>>> import math
>>> help(math)
Help on built-in module math:

NAME
    math

DESCRIPTION
    This module provides access to the mathematical functions
    defined by the C standard.

FUNCTIONS
    acos(x, /)
        Return the arc cosine (measured in radians) of x.

        The result is between 0 and pi.

    acosh(x, /)
        Return the inverse hyperbolic cosine of x.

    asin(x, /)
        Return the arc sine (measured in radians) of x.

        The result is between -pi/2 and pi/2.

    asinh(x, /)
        Return the inverse hyperbolic sine of x.

    atan(x, /)
        Return the arc tangent (measured in radians) of x.

        The result is between -pi/2 and pi/2.
```

La méthode import

Quand on importe un module, pour utiliser une fonction du module, il faut précéder la fonction du nom du module. Par exemple pour avoir le carré d'un nombre avec la fonction **pow** du module **math** on aura:

```
>>> help(math.pow)
Help on built-in function pow in module math:

pow(x, y, /)
    Return x**y (x to the power of y).

>>> # Affichage de la doc de la fonction pow
>>> import math
>>> help(math.pow)
Help on built-in function pow in module math:

pow(x, y, /)
    Return x**y (x to the power of y).

>>> # Utilisation de la fonction pow pour calculer le carré
>>> math.pow(3, 2)
9.0
>>> math.pow(2, 2)
4.0
```

Dans l'exemple précédent on affiche la documentation de la fonction **pow** avant de l'utiliser quand on connaît déjà le rôle de la fonction on peut s'en passer. Quand on veut afficher la documentation il faut éviter de mettre les parenthèses dans **help** du type **help(math.pow())** qui n'affiche pas la doc de la fonction **pow**.

Espace de nom spécifique

Quand on import un module par exemple **import math**, cela crée un espace de nom dénommé « math » qui contient les variables et les fonctions du module math.

Quand on exécute **math.pow(3, 2)**, on précise à Python que nous souhaitons exécuter la fonction pow contenue dans le module **math** d'espace de nom **math**.

Grâce aux espaces de nom, on peut avoir dans un autre module où dans l'espace de nom principal une autre fonction **pow** sans conflit car chaque fonction lors de l'appel spécifie son espace de nom.

Parfois on a besoin d'importer un module avec un nom différent pour l'espace de nom avec le mot clé **as**.

import nomModule => importe avec comme espace de nom nomModule

import nomModule as nomEspaceDeNom => import avec comme espace de nom nomEspaceDeNom

```
>>> # Import avec espace de nom par défaut
>>> import math
>>> help(math.pow)
Help on built-in function pow in module math:

pow(x, y, /)
    Return x**y (x to the power of y).

>>> math.pow(2,2)
4.0
```

```
>>> # Import avec espace de nom spécifique
>>> import math as mathematique
>>> help(mathematique.pow)
Help on built-in function pow in module math:

pow(x, y, /)
    Return x**y (x to the power of y).

>>> mathematique.pow(2,2)
4.0
```

La méthode from ... import

Page 52

Exemple de création de module

Page 49

Exercices

Exercice x

FIN CHAPITRE 6