

Sérialisation

Modules

Packages

Classes

sérialisation

Les modules

Un programme python peut être composé de plusieurs fichiers sources appelés modules

Un module est considéré comme une bibliothèque de classes, de fonction et de variables globales. L'instruction `import` est utilisée pour importer un module.

Package

Un package est une collection de modules, concrètement un package est un dossier dont le nom est le nom du package et il doit contenir un fichier nommé `__init__.py`

Utilisation d'un module

m1.py

```
def afficher(nombre):  
    print ('Nombre:', nombre)
```

Le dossier contenant m1.py doit aussi contenir un fichier nommé `__init__.py`

modules.py (dans le même dossier que m1.py)

```
import m1  
m1.afficher (34)
```

Pour importer un module, on utilise le nom du fichier sans l'extension et on peut importer une liste de modules séparés par des virgules.

L'instruction `import` supporte la possibilité d'importer une classe ou une fonction à partir d'un module

```
from m1 import afficher  
afficher (34)
```

Si le module se trouve dans un dossier différent il faut préciser le chemin complet dans l'instruction `import`, exemple: `import libs.math` (`libs` est le nom du dossier contenant le module)

Classes

Syntaxe

Une classe python peut contenir la définition d'une ou de plusieurs méthodes et un constructeur

La définition d'une méthode est similaire à celle d'une fonction, sauf qu'elle doit avoir au moins un paramètre qui est une référence vers l'instance de la classe (self)

self est utilisé pour accéder aux attributs de la classe

Le constructeur est une méthode nommée `__init__`

Un constructeur ou une méthode sans paramètres doivent être définis respectivement comme suit `__init__(self):` et `def nom_méthode(self):`

Exemple1

```
class Routeur :  
    # Constructeur  
    #Le premier paramètre est une référence vers l'instance de la classe  
    def __init__(self,nom,nb_interfaces, marque):  
        self.nom=nom  
        self.nb_interfaces=nb_interfaces  
        self.marque=marque  
    def afficher(self):  
        print(self.nom,self.marque)
```

La classe routeur possède 3 attributs définis par son constructeur

Instanciación y acceso a los miembros de una clase

Instanciación

```
r = Routeur("ALfa AX", 4, "Alf")
```

Acceso a los miembros

```
r.nom="Alf"
```

```
r.afficher()
```

Sérialisation et persistance de données

La persistance des données et le stockage d'un objet ou d'une collection d'objets sur un support de stockage.

La sérialisation consiste à transformer un objet ou une collection en un format compatible avec le support de transport sur un réseau ou bien avec le support de stockage dans le cas d'une persistance de données, python supporte 3 formats de sérialisation:

- XML
- JSON
- Binaire (format propriétaire)

La désérialisation est l'opération inverse qui permet de récupérer un objet ou une collection à partir de sa forme sérialisée

Les fonctions de sérialisation/désérialisation sont définies dans le module pickle

Sérialisation / Désérialisation binaire

Sérialisation

La fonction `dumps` du module `pickle` permet de sérialiser un objet:

```
r=Routeur("ALfa AX", 4, "Alf")
```

```
data=pickle.dumps(r)
```

`data` est un table d'octets contenant une représentation binaire de l'objet `r`

```
print (type(data))
```

```
print(data)
```

Désérialisation

La fonction `loads` désérialise un tableau d'octets et retourne sa forme d'origine

```
routeur=pickle.loads(data)
```

```
print (routeur.nom)
```

-

Persistance des données dans un fichier binaire

Enregistrement des données dans un fichier

Pour écrire dans un fichier en mode binaire, il doit être ouvert en mode écriture binaire (wb)

```
fd=open("r.dat", "wb")
```

```
pickle.dump(r,fd)
```

Désérialisation à partir d'un fichier

Pour lire à partir d'un fichier en mode binaire, il doit être ouvert en mode lecture binaire (rb)

```
fd=open("r.dat", "rb")
```

```
r=pickle.load(fd)
```

Exercices

1. Ecrire une classe Produit (un produit possède une référence (ref), un libelle (libelle), un prix et une quantité)
2. Une liste nommée produits sera utilisée pour stocker les produits, écrire le programme qui réalise les opérations suivantes sur les produits:
 - Ajout
 - Affichage de tous les produits
 - Recherche d'un produit par référence
 - Suppression d'un produit
 - Modification