

1A_1_python_module_and_package

November 27, 2023

1 Python Module and Packages

1.1 Python - Modules

Plus notre application va grossir, plus il va être difficile de maintenir le code. Pour cela, il est possible de découper le code en plusieurs fichiers. Chaque fichier est appelé un module. Un module peut contenir des fonctions, des classes, des variables, etc. Les modules permettent de mieux organiser notre code. Il est possible de créer nos propres modules ou d'utiliser des modules existants (math, random, etc..).

1.1.1 Créer un module

Pour créer un module, il suffit de créer un fichier avec l'extension `.py`. Par exemple, nous allons créer un fichier nommé `callee.py` avec le contenu suivant:

```
def say_hello():  
    print("Hello from the callee")
```

Ici, nous avons créé un module nommé `callee` qui contient une fonction `say_hello`.

1.1.2 Utiliser un module

Pour utiliser un module, il faut l'importer. Pour cela, nous utilisons le mot clé `import` suivi du nom du module. Par exemple, pour importer le module `callee` créé précédemment, il faut écrire:

```
import mymodule
```

Note: Le module doit être dans le même répertoire que le fichier qui l'importe.

Utiliser un modules dans différents niveau d'arborescence Si le module est dans un répertoire différent, il faut préciser le chemin relatif du module dans le dossier `sys.path` de python.

```
import sys  
sys.path.append("path/to/module")
```

```
import callee
```

2 Exemples

Note: Si les fichiers sont dans le même répertoire, il n'est pas nécessaire de préciser le chemin relatif.

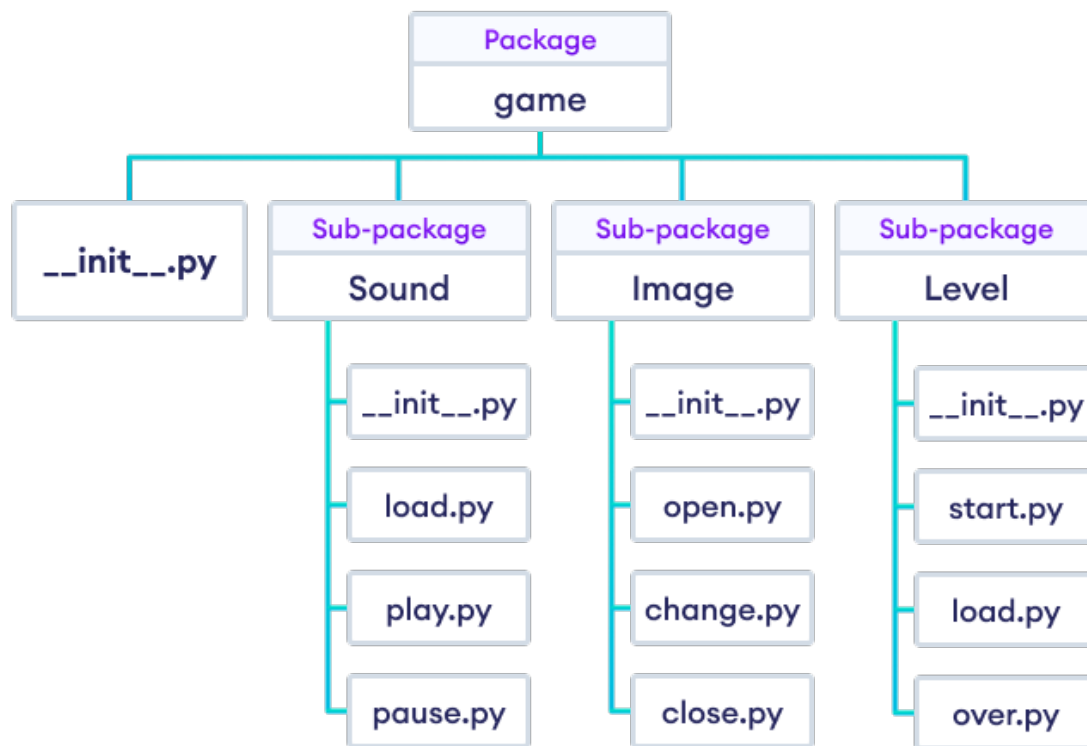
2.1 Modules - from ... import

Il est possible d'importer une fonction ou une variable d'un module. Pour cela, il faut utiliser le mot clé **from** suivi du nom du module, du mot clé **import** et du nom de la fonction ou de la variable. Ces imports sont appelés des imports explicites.

Les imports explicites permettent d'importer uniquement les fonctions ou les variables dont nous avons besoin. Cela permet d'éviter d'importer tout le module et donc de réduire la consommation de mémoire.

3 Python - packages

Les packages permettent de regrouper des modules. Un package est un répertoire qui contient un fichier nommé `__init__.py`. Ce fichier est vide, mais il est obligatoire pour que le répertoire soit considéré comme un package.



Source

3.1 Le fichier `init.py`

Le fichier `__init__.py` peut contenir du code. **Ce code est exécuté lorsque le package est importé.** Par exemple, si nous avons le package `mypackage` qui contient le fichier `__init__.py` avec le contenu suivant:

```
print("Hello from the package")
```

Lorsque nous importons le package `mypackage`, le message `Hello from the package` est affiché.

3.2 Importer un module d'un package

En python nous pouvons importer un package en utilisant l'opérateur `..`. Par exemple, dans la figure ci-dessus, si nous voulons importer le module `start` il faudra cet import :

```
import Game.Level.start
```

En admettant que le module `start` contienne une fonction `select_difficulty()`, pour appeler cette fonction, il faut écrire:

```
Game.Level.start.select_difficulty(2)
```

3.2.1 Importer un module d'un package avec `from ... import`

Pour éviter d'avoir à réécrire tout le prefix du package (chemin du package), il est possible d'utiliser la syntaxe `from ... import`. Par exemple, pour importer la fonction `select_difficulty()` du module `start`, il faut écrire:

```
from Game.Level.start import select_difficulty
```

On pourra ensuite appeler la fonction `select_difficulty()` directement:

```
select_difficulty(2)
```

3.2.2 Autre méthode

Si on veut garder un minimum de consistance dans le code et ne pas avoir une fonction sortie de nulle part. On peut utiliser la syntaxe suivante :

```
from Game.Level import start
```

On pourra ensuite appeler la fonction `select_difficulty()` directement:

```
start.select_difficulty(2)
```

Ici, on garde une trace de la provenance de la fonction `select_difficulty()`.

Note: Pour importer les package, Python regarde dans les paths contenus dans le `sys.path`. Il faut par conséquent ajouter le chemin du package dans le `sys.path` pour pouvoir l'importer.