
TP n° 3 : modularité

1. Vrai / Faux

1. Pour importer deux fonctions *f1* et *f2* d'un module *mod*, on écrit *from mod import f1 and f2*.
2. Dans l'écriture *mystere.myst*, *mystere* est une constante et *myst* est un module.
3. Une bibliothèque peut contenir de nombreux modules.
4. *API* signifie « Application Pour Internet ».

2. Documentation

Il est possible d'obtenir des informations sur les modules et leurs fonctions, par exemple avec les fonction *help* et *dir*.

Exemple : écrire dans la console de Thonny les instructions suivantes.

- `import math`
- `help(math)`
- `dir(math)`
- `math.sqrt(2)`
- `help(math.sqrt)`
- `dir(math.sqrt)`

1. À quoi sert la fonction *help* ?
2. À quoi sert la fonction *dir* ?
3. Que remarque-t-on pour les fonctions contenues dans *math.sqrt* ?

3. Rendre un programme modulaire

Ouvrir le programme *TP03_programme_brut.py*.

1. Pourquoi est-il peu lisible ? Comment pourrait-on l'améliorer ?

2. Quel est le rôle de la fonction *fonction1* ? La renommer, et la documenter.
3. Quel est le rôle de la fonction *fonction2* ? La renommer, et la documenter.
4. Quel est le rôle de la fonction *fonction3* ? La renommer, et la documenter.
5. Quelles sont les deux parties de ce programme, qui pourraient être séparées dans deux modules différents ?

4. Création d'un module

- Renommer le module *statistiques.py*, et sauvegarder le dans un dossier nommé *mon_module*, placé dans le dossier *TP03_modularité*, lui-même placé dans le dossier *nsi*, sur votre disque travail.
- Créer dans le même dossier un module nommé *test_statistiques.py*. Ne garder dans le module *statistiques* que les fonctions, et placer les tests des fonctions dans le module *test_statistiques*.

1. Quelle instruction permet d'importer le module *statistiques* dans le module *test_statistiques*, en créant l'alias *stat* ?
2. En utilisant le tableau *tab* fourni, tester chacune des fonctions dans le module. Compléter pour cela l'affichage du résultat à l'aide d'instructions du type :

```
1 print("La somme des éléments est égale à ", stat.somme(tab))
```

Quels sont les résultats obtenus aux trois tests ?

5. Création d'un package

- Créer un dossier appelé *mon_package*.
- Copier dans ce dossier le fichier d'initialisation de package, optionnel (depuis la version Python 3.3) nommé *__init__.py*.
- Créer un sous dossier *stat* dans lequel vous y copierez le fichier *statistiques.py*.
- Créer un sous dossier *fibonacci* dans lequel vous y copierez le fichier *fibonacci.py*.

- Ouvrir Thonny et créer un nouveau programme nommé *test_package* dans le dossier *TP03_modularité*.
- Copier les lignes suivantes :

```
1 import mon_package.stat.statistique as st
2 tab = [2,4,3,5,9,6,8,9,9]
3 print(st.mediane(tab))
```

- Compléter le code pour afficher la moyenne et la médiane des éléments du tableau.
- Compléter le code pour afficher la valeur du 9^e élément de la suite de fibonacci.

6. Pour aller plus loin

Le fichier `__init__.py` ne sert pas qu'à déclarer un dossier comme un package importable (facultatif depuis version Python 3.3). Il peut contenir du code qui est exécuté automatiquement, une seule fois, quand on importe le module.

- Ouvrir le fichier `__init__.py`, et copier le code ci-dessous et sauvegarder.

```
1 print("Bienvenue dans mon package")
```

- Exécuter à nouveau *test_package.py*.
- Ouvrir le fichier `__init__.py`, et copier le code ci-dessous et sauvegarder.

```
1 __version__ = "1.0.NSI"
```

- Dans la console de Thonny, exécuter les instructions suivantes :

```
>>> import mon_package
>>> mon_package.__version__
```

Capacités exigibles

- Créer des modules simples et les documenter.