

# 1 もくじ

- [モジュールとパッケージ](#)
  - [モジュール](#)
  - [パッケージ](#)

## 2 モジュールとパッケージ

### モジュール

ひとつのスクリプトファイルは **モジュール** として扱うことができます。

また, あるスクリプトファイルから, そのモジュールを取り込むことができます。  
**import** でそのファイルまでのパスを指定することで, 取り込むことができます。

```
# module.py, 取り込まれるモジュール
```

```
def func():  
    print("hello, world!")
```

```
# main.py
```

```
import module
```

```
def main():  
    module.func()
```

```
if __name__ == '__main__':  
    main()
```

# パッケージ

複数のモジュールをまとめてパッケージとして、扱うことができます。

`numpy` や `matplotlib` は `PyPI` にあがっているパッケージです。

## PyPIとは

PyPIはPython Package Indexの略で、誰でもパッケージを登録でき、ここにあるパッケージは `pip install` でインストールし、`import` で取り込み、利用することができるようになります。

パッケージは複数のモジュールが含まれており、ものによっては更に多くのフォルダ/ファイルから構成されている場合があります。その場合、`.` で階層構造を表すことで、どの部分を `import` するか選択することができます。

以下にmatplotlibにおけるモジュールのimportを示します。

```
import matplotlib.pyplot

fig = matplotlib.pyplot.figure()
ax = fig.add_subplot(111)
```

これは `matplotlib` という大元のパッケージから、`pyplot` を `import` しています。  
ちなみに、上記は以下のように `import` することもできます。

```
from matplotlib import pyplot

fig = pyplot.figure()
ax = fig.add_subplot(111)
```

一応、`matplotlib` 全体を `import` して、`pyplot` を呼び出すことも可能です。

```
import matplotlib

fig = matplotlib.pyplot.figure()
ax = fig.add_subplot(111)
```

また、`as` で略称をつけることができ、より簡潔に利用することが可能になります。

```
from matplotlib import pyplot as plt
```

```
fig = plt.figure()
```

```
ax = fig.add_subplot(111)
```