

## モジュールの使い方概論

---



こちらへんから、モジュールを使うことについてちゃんと説明してみたいと思います。たぶん新展開ですよ。

イメージとしては、今までが「素のpython」だったのに対して、モジュールを使い始めると「パワーアップしたpython」になるといったところでしょうか。

今までのテキスト内では、一度、お遊びでrandomモジュールを使ってみたことがありました。importを使って、randomモジュールで準備されている便利な関数をひとつ借りてきました。借りるといっても別に遠慮する必要はなく、必要に応じてじゃんじゃん借りまくればいいです。

あのときは確か、下のように、randomモジュールからrandint関数を借りてきました。

```
>>> from random import randint
```

これは、randomというモジュールから、randintという名前のついた何か（ここでは関数）を今から使いたいです、という意味です。こうすると、今までの「素のpython」君は、randomという名前のついた装備のようなものを我が身にまとい、今までは持っていなかった機能であるrandintという関数が見えるようになるわけです。

```
>>> randint(1, 6)
2
（実行結果はそのつど異なります）
```

普通にインストールされたpythonでは、標準的なモジュールが最初からたくさん付属しています。（標準配布モジュールと呼ぶときがあります。）やりたいことはほとんどこのモジュールで間に合ってしまうくらいの充実ぶりです。特殊な用途のときには、外部で配布されているモジュールをもらってきてインストールすることもありますけどね。

randomモジュールは、乱数にかかわる色々な関数を準備してくれています。必要なたびにリファレンスを見て勉強すれば充分なのですが、さっき使ったrandintのほかに、例えばchoiceとかshuffleというこれも便利な関数があります。

下のような書き方で、choice関数とshuffle関数が使えるようになります。

```
>>> from random import choice, shuffle
```

choiceは、リストの中から適当にひとつ値を選んでくれる関数。使い方のイメージは、下のような感じ。

```
>>> a = ['Stone', 'Scissors', 'Paper']
>>> choice(a)
'Paper'
```

上の例は、ジャンケン手出し支援システムです。

shuffleは、リストの中身を適当にシャッフルしてくれる関数です。sortの反対みたいなもんですね。使ってる例は下のような感じ。

```
>>> b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> shuffle(b)
>>> b
[2, 7, 9, 10, 5, 3, 4, 6, 8, 1]
```

トランプゲームでも作ることがあるとしたら、それこそカードの「シャッフル」に使えますね。

randomモジュールを使うときには、下のような書き方もできます。

```
>>> import random
>>> random.choice(a)
>>> random.shuffle(b)
```

このときは、randomモジュールから何を使いたいかをあらかじめimportで宣言していません。代わりに、choiceとかshuffleを使いたいときに、「randomモジュールの中にあるはずのchoice関数」とか「同モジュールの中にあるはずのshuffle関数」という書き方でこれを使うことになります。

どちらでもいいです。

目安としては、使う機能が限られているときは from 何々 import 何々 という書き方がいいのかなあ。でも、from 云々 という方法だと、他にchoiceという変数を自分で決めて使っていたときにこれを塗りつぶしてしまう危険もありますね。random.choice という書き方で呼び出すのなら、他に自前の choice 変数なんかを使っても大丈夫です。pythonには見分けがつけられるわけですから。まあ、色々と使い方を試して覚えるのがいいと思います。

importなんか待たないで、準備されているものなら最初からなんでも使えるようにしておけばいいじゃないか、と考える方もいるでしょう。まあ、その通りではあるんですけど、色々なモジュールをimportしていくうちに、pythonの占有するメモリの量はだんだん増えていきます。だから、中核的な機能だけは「素の」pythonに入れておくけど、あとは追加的に読み込みながら使ってね、というスタイルになったんでしょう。多くのモジュールに用がないうちは、pythonをスリムに保って使いたいでしょ。「素の」状態でも充分仕事がこなせることは、今までの練習問題なんかで体験してもらったとおりですし。

さて、今はrandomモジュールについて話をしていましたが、pythonで他にはどんな「標準配布の」モジュールが使えるでしょうか。主なもので、使い道がわかりやすそうなものを下にざっと挙げてみます。

- datetime ... 年月日、時分秒を扱うときに非常に便利な機能をたくさん提供します。
- calendar ... ひと月分のカレンダー用データを簡単に作れたりする機能などを提供します。
- math ... 三角関数や円周率など、少し高度な数学的处理をする機能を色々と提供します。
- random ... 乱数にかかわる色々な機能を提供します。
- re ... 「正規表現」と呼ばれる表記法をサポートし、文字列の込み入った検索や置換を行わせる機能を提供します。
- xml ... XMLというデータ形式を扱うために便利な機能を提供します。
- urllib ... これはテキストで少しだけ触れました。Web上のデータを取得するための機能を提供します。
- email ... 電子メールのデータを解析したり、新たに電子メール用データを作るための機能を提供します。
- smtplib ... 電子メールを送信するための機能を提供します。
- zipfile ... ZIP形式の圧縮ファイルを扱うための機能を提供します。
- glob ... ファイルの一覧を取得するための機能を提供します。
- sqlite3 ... SQLの使えるデータベース機能を提供します。
- Tkinter ... ボタンや入力エリアなどを持ってマウス操作に反応する、いわゆるGUIの機能を提供します。

実際はもっともっと、目がくらむほどたくさんあります。ほとんど使わないようなモジュールも結構ありますけど。

また、この一覧の中でもすでに、わからん用語がある人がいるでしょう。「正規表現」「SQL」「XML」とか。これはpythonだけが使うわけじゃない、プログラミング一般に関係する事柄です。説明すべき時が来たら、そのとき説明しますよ。

Tkinterは面白いんですけど、普通の人が仕事で使うか、といわれると、まあ使わないといっているんじゃないかな。イベント駆動とかコールバックとか、まだまだ新しい概念を呑みこむ必要がありますしね。デモくらいやってみせるかもしれないけど、このテキストで習得する予定は、今のところないとしておきましょう。

上に列挙したモジュールのうちその他は、今後ちょっとづつ時間をかけて説明していきたいと思います。どこまでやりきれるかな。（筆者がね。）

モジュールの機能を全部暗記して使いこなしたりすることはあまりありません。「こんな機能があるはずだけど...」といった感じでリファレンスマニュアルを読んで使い方が確認できれば充分です。Windowsに普通にpythonをインストールすれば、ヘルプファイルも一緒にインストールされているはずですから、そいつを探すのが普通です。でも英語なんですよ... まあ、Googleなんかで「python ライブラリ リファレンス」くらいに検索すれば、ある程度の情報は日本語で見られます。

あと、念のため。モジュールをimportしても、pythonをいったん終了したら全部御破算です。次にpythonを動かしたら、もちろんimportしなおしですよ。