

▼ 第一回 Google Colabの紹介

いまやPythonの使い道は多岐に渡っていて、すべてを把握するのは不可能です。正直言って、「できないことがない」ような気がします。例えば、Excelのファイルにアクセスし、一部を書換えるとか、wordに自動で作文を書くようなことすらも簡単にできてしまいます。

この講義では、これらを横断的に使うための、基盤となる知識と、特に化学で使いそうな機能を紹介しようと考えています。

なぜPython?

コンピュータ言語は、Python以外にも無数にあります。また、Pythonよりも高速に処理できる言語もたくさんあります。例えば、理論化学系の研究室であれば、Fortranを使う機会があると思いますが、Fortranに比べればPythonは100倍ぐらい遅い言語です。

ただし、処理が速ければ、早く仕事(我々の場合でいえばデータ解析)が終わるというわけではありません。数値ばかりがたくさんならんだデータを扱っている限り、Fortranは速いのですが、文字列や、ネットワークや、数式などを扱おうとすると、とたんにとてつもない苦難に見舞われます。Pythonであれば、扱えるデータの幅がひろがります。「なんでも」「そこそこの速さで」「短いプログラムで」処理できる言語として、Pythonを使うメリットがあります。

機械学習の分野では、(とくに近年になって)新しいアルゴリズムはまずPythonでライブラリが提供される、ということが多くなっています。Pythonを学んでおけば、常に最新のアルゴリズムを利用できるということです。

Pythonには、大量のデータを簡潔なプログラムで高速に扱うnumpyという仕組みが提供されています。実際には、画像処理(OpenCV)も機械学習も統計処理も、ほとんどの計算関連の機能はnumpyを使って構築されています。numpyを知れば、ほかの言語ではありえない簡潔さで、データ処理を行えます。

簡潔にコードが書けると、バグを減らすことができます。例えば、numpyなら、行列 A の転置とベクトル x をかけて y を足す計算は、以下のように表記します。

```
y = A.T @ x + y
```

ちなみに、もしnumpyを知らなかったら、こんなコードを書くことになるでしょう。

```
for i in range(len(x)):
    s = 0
    for j in range(len(x)):
        s += A[j][i]*x[j]
```