

## #16の 解説

### 1. Python、numpy、TensorFlow について

Python はプログラミング言語として特殊なものではありません。Python で記述可能なプログラムは、C や C++、LISP、ないし Java といった一般的に用いられているプログラミング言語でも記述可能です。

Python が最近流行しているディープラーニングで使用されるのはなぜでしょうか？

それは、マトリックス演算（行列および行列に関する計算）を、numpy や TensorFlow というライブラリ等を用いて、GPU というグラフィックボードに搭載されている装置を駆使し、高速に扱えるということが主な理由となっています。

今回の演習では、2次元の回転操作を行うプログラムを実行させました。ここでは2次元の座標を「base\_point」等という記号で扱い、回転操作を単に「rot」等という記号で扱っています。1学期の途中、エクセルシートで2次元・3次元のアフィン変換操作を行いました。非常に複雑な工程が必要でした。しかしながら、Python を用いれば、このように非常に単純な記述にて扱うことが可能です。

日本では、平成30年3月に公示された新高等学校学習指導要領に基づき、共通必修科目「情報Ⅰ」にてPythonを使った講義が取り入れられています。

※「共通必須」科目です。「選択科目」は情報Ⅱです。

[日本語サイト]

高等学校情報科（各学科に共通する教科）

[https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1416746.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416746.htm)

高等学校情報科「情報Ⅰ」教員研修用教材（本編）

[https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1416756.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm)

第3章 コンピュータとプログラミング (PDF:9.3MB)

[https://www.mext.go.jp/content/20200722-mxt\\_jogai02-100013300\\_005.pdf](https://www.mext.go.jp/content/20200722-mxt_jogai02-100013300_005.pdf)

上記内部に記載されているPDF3ページ(オリジナル102ページ)のプログラムをjupyter notebookにて実行した結果を図1に示します。

```
File Edit View Insert Cell Kernel Widgets Help
+ < > Run [ ] Code [ ]
```

```
In [1]: import sys # 変数や関数のライブラリを呼び出し
x=sys.float_info.max #float 型で表せる最大値を x に代入
print(x) # 画面に x の値を表示
x=1.797893134862315799999e+308 #float 型の小数点以下に 5 桁「9」の追加した値を代入
print(x) # 画面に x の値を表示
x=1.8e+308 #float 型の最大値より大きな値を代入
print(x) # 画面に x の値を表示

1.7978931348623157e+308
1.7978931348623157e+308
inf

In [2]: x = 28-27 #x に 28-27 の計算結果を入れる
print(x) # 画面に x の値を表示
y = 0.28-0.27 #y に 0.28-0.27 の計算結果を入れる
print(y) # 画面 y の値を表示

1
0.010000000000000009
```

関連して都道府県の教育委員会レベルでも下記のような資料が出ています。

<http://www.edu.pref.kagoshima.jp/research/result/siryou/hyoudai/jyouhoukyouiku/pdf/2036-jyoho153.pdf>

下記は高校生用のテキストの一例です。

<https://www.jikkyo.co.jp/book/detail/22023322>

## 2. Jupyter と notebook について

木星 (Jupiter) とは「j」と「y」とで異なる綴りです。由来は、Julia + Python + R という言語の組み合わせから来ているとのこと。

Jupyter はブラウザで対話的に実行でき、また結果をグラフで視覚化できるため、主に入門用や確認用としてよく使用されます。

プログラムのソースコードは拡張子が「.ipynb」の notebook という形式 (ipynb の「ip」は「IPython」という Jupyter の核となる Python の一種、「nb」は「notebook」の意味) で扱われ、セルという単位にプログラムを記入し、セル単位で実行しながら動作確認可能となっています。

セルの実行に伴い変化する環境 (変数の状態を含む) が保持されます。セルを追加したり変更したりすることで、割り込み的に状態を解析することもできます。

今回、最後までセルを順に実行後、途中から再度繰り返すと、データが増えていく動作を確認しました。すなわち、途中経過が保存される状態なので、任意の場所のセルを実行させると、思わぬ結果をもたらす危険性も認識できたと思います。

## 3. ネットでの検索は重要です

Python の文法をはじめ、numpy や TensorFlow ライブラリに含まれる多くの関数について、詳しい説明を得ることができます。

たとえば、「TensorFlow」と「reduce\_sum」で検索すると、下記のような URL がヒットし

できます。

[日本語サイト] <https://dev.classmethod.jp/machine-learning/tensorflow-math/>

[英語サイト] [https://docs.w3cub.com/tensorflow~python/tf/reduce\\_sum/](https://docs.w3cub.com/tensorflow~python/tf/reduce_sum/)