

第7回 モジュール

目次

- モジュール
- from文
- as文
- 標準モジュール
- サードパーティー製のモジュール
- matplotlibを使ってみる(1)
- matplotlibを使ってみる(2)
- matplotlibを使ってみる(3)
- matplotlibを使ってみる(4)
- 課題7

モジュール

- Pythonには、標準モジュールと呼ばれるプログラムを構成する多数の部品が備わっています。
- 各モジュールには、関連する関数やクラスが用意されており、組み込みのデータ型や関数では実現できないような複雑な機能を手軽に実行することができます。
- モジュールを利用するには、import文を使ってあらかじめモジュールを読み込んでおく必要があります。

- 構文

```
import モジュール名
```

- 例: math(数学)モジュールのインポート

```
import math
```

```
a = math.sqrt(2)
```

```
print(a)
```

```
1.4142135623730951
```

インポート

平方根を計算する関数

- モジュールに含まれる関数やクラスを呼び出すには、頭に「モジュール名.」を付けます。

from文

- import文に合わせてfrom文を使うと、関数をモジュール名を省略して呼び出すことができるようになります。

- 構文

```
from モジュール名 import 関数名
```

- 例

```
from math import sqrt
a = sqrt(2)
print(a)
```

sqrt関数をモジュール名を省略して呼び出し

```
1.4142135623730951
```

- インポートする関数名として*を指定すると、モジュール内のすべての関数を呼び出せるようになります。

- 例

```
from math import *
a = sqrt(2)
print(a)
```

*はワイルドカードとして機能

```
1.4142135623730951
```

as文

- import文に合わせてas文を使うと、読み込んだモジュールや関数に別名を割り当てることができます。

- 構文

```
import モジュール名 as 別名  
from モジュール名 import 関数名 as 別名
```

- 例

```
import math as m  
a = m.sqrt(2)  
print(a)
```

```
1.4142135623730951
```

mathモジュールをmとしてインポート

```
from math import sqrt as sq  
a = sq(2)  
print(a)
```

```
1.4142135623730951
```

sqrt関数をsqとしてインポート

標準モジュール

● 標準モジュールの一覧

<https://docs.python.jp/3/py-modindex.html>

m	
macpath	<i>Mac OS 9 path ma</i>
mailbox	<i>Manipulate mailbo</i>
mailcap	<i>Mailcap file handli</i>
marshal	<i>Convert Python ob</i>
math	<i>Mathematical func</i>

9.2. [math](#) — 数学関数 ¶ (原文)

このモジュールはいつでも利用できます。標準 C で定義されている数

これらの関数で複素数を使うことはできません。複素数に対応する必
ください。ほとんどのユーザーは複素数を理解するのに必要なだけの数
関数の区別がされています。これらの関数では複素数が利用できないけ
外が発生します。その結果、こういった理由で例外が送出されたかに早

このモジュールでは次の関数を提供しています。明示的な注記のない関

9.2.1. 数論および数表現の関数 (原文)

`math.ceil(x)` (原文)

x の「天井」 (x 以上の最小の整数) を返します。 x が浮動小数点数
されます。

`math.copysign(x, y)` (原文)

x の大きさ (絶対値) で y と同じ符号の浮動小数点数を返しま
`copysign(1.0, -0.0)` は `-1.0` を返します。

サードパーティー製のモジュール

- 標準モジュール以外にも、Pythonで利用できる様々なモジュールがインターネット上で公開されています。

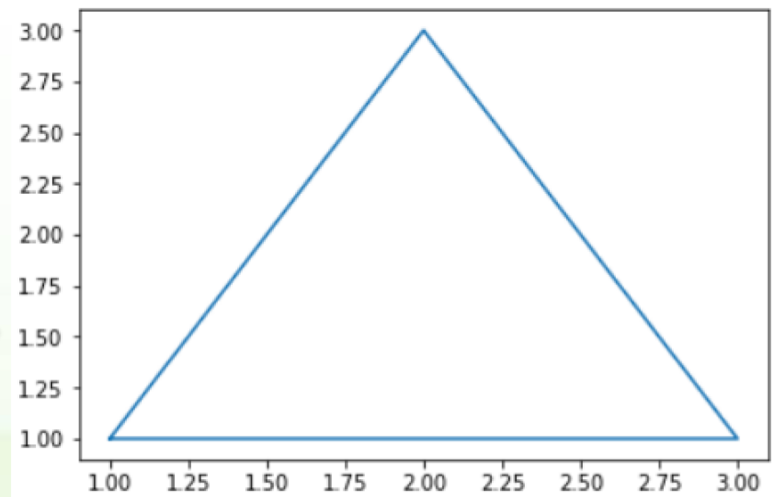
モジュール名	概要
matplotlib	グラフなどの可視化モジュール
seaborn	matplotlibの見栄えをより綺麗にするモジュール
NumPy	数値計算を効率的に行うためのモジュール
SymPy	数式・記号計算用モジュール
pandas	データ解析支援モジュール
scipy	NumPyを利用した数値解析モジュールで、統計、最適化、積分、線形代数、フーリエ変換、信号処理、遺伝的アルゴリズムなどの関数を提供する
scikit-learn	機械学習モジュール
NetworkX	グラフ・ネットワーク計算と可視化モジュール
Basemap	地図描画モジュール

matplotlibを使ってみる(1)

- matplotlibはPythonでグラフを描画するための可視化モジュールで、複数のモジュールから構成されています。
 - matplotlib.pyplot・・・グラフ描画モジュール
 - matplotlib.patches・・・図形描画モジュール
 - matplotlib.animation・・・アニメーション描画モジュール
 - ...
- Jupyterでmatplotlibを使用するためには、ブラウザにグラフを埋め込む必要があります。
%matplotlib inline
宣言が必要になります。

```
import matplotlib.pyplot as plt
%matplotlib inline

plt.plot([1, 2, 3, 1],[1, 3, 1, 1])
```



折れ線グラフを描画するplot関数 x座標 y座標

matplotlibを使ってみる(2)

- マーカーや軸ラベルを設定してみます。

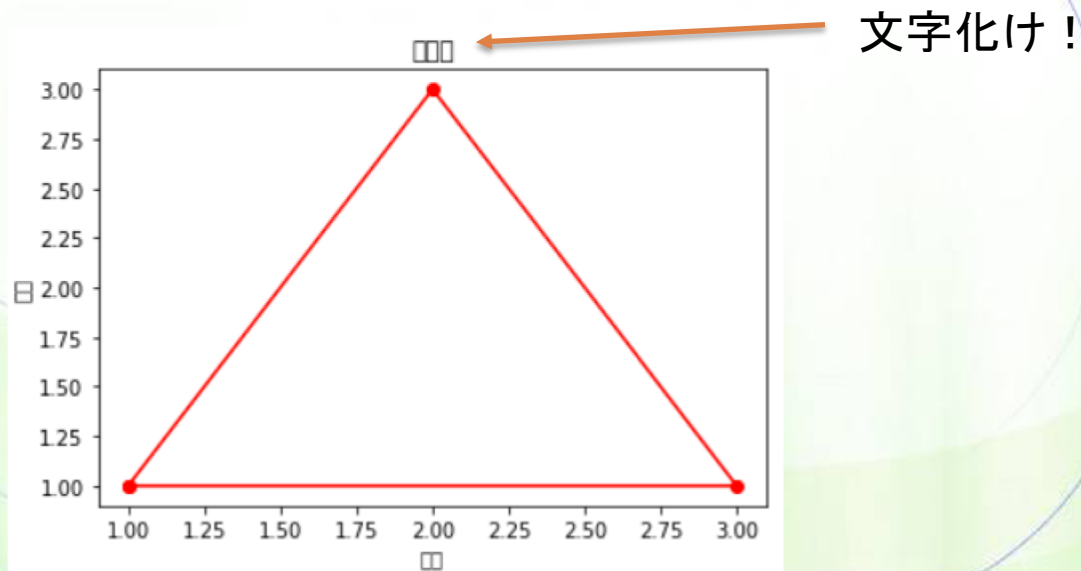
```
import matplotlib.pyplot as plt
%matplotlib inline

plt.plot([1, 2, 3, 1],[1, 3, 1, 1], color="red", marker="o")
plt.title("三角形")
plt.xlabel("横軸")
plt.ylabel("縦軸")
```

線の色 マーカー

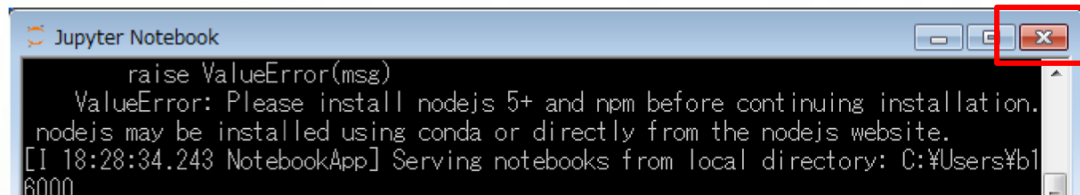
↑ ↑

← タイトルやマーカーの設定



matplotlibを使ってみる(3)

- matplotlibの日本語フォントの文字化け対策
 - 独立行政法人情報処理推進機構が配布している日本語フォントをインストールする(IPAフォント)
 - 手順:
 - <http://pana4405.u-shizuoka-ken.ac.jp/python201803>からipaexg00301.zipをダウンロードする。
 - ダウンロードしたファイルを展開し、ipaexg.ttfをC:\ProgramData\Anaconda3\Lib\site-packages\matplotlib\mpl-data\fonts\ttfに入れる。
 - Jupyter Notebookを終了し、ブラウザも閉じる



```
Jupyter Notebook
raise ValueError(msg)
ValueError: Please install nodejs 5+ and npm before continuing installation.
nodejs may be installed using conda or directly from the nodejs website.
[I 18:28:34.243 NotebookApp] Serving notebooks from local directory: C:\Users\pykousa\6000
```

- [スタート]-[コンピュータ]を開き、C:\Users\自分のユーザアカウント名(今回は、pykousa)のmatplotlibフォルダを削除する
- Jupyter Notebookを起動する

matplotlibを使ってみる(4)

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
plt.rcParams["font.family"] = "IPAexGothic"
```

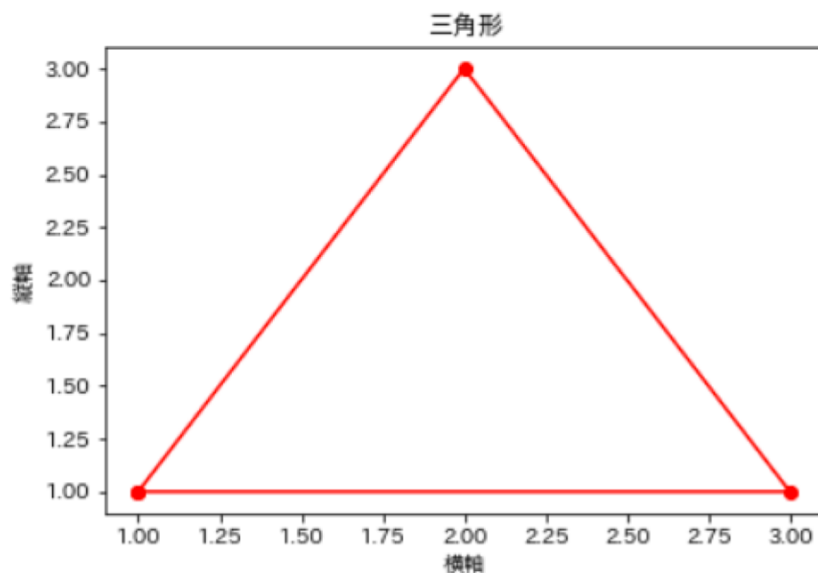
IPAフォントの指定

```
plt.plot([1, 2, 3, 1],[1, 3, 1, 1], color="red", marker="o")
```

```
plt.title("三角形")
```

```
plt.xlabel("横軸")
```

```
plt.ylabel("縦軸")
```



課題7

- matplotlibを用いて、

$$-5 \leq x \leq 5$$

において、

$$y = 3x^3 - 2x^2 + 10x + 30$$

のグラフを描いてください。

```
import matplotlib.pyplot as plt
%matplotlib inline

xp = [x * 0.1 for x in range(-50, 51)]
plt.plot(xp, ここを考える)
```

