第7回　モジュール

matplotlib, pandas, NumPy, SciPy等

目次

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

  - 演習：math（数学）モジュールのインポート
    - Spyderで[ファイル]-[新規ファイル]として、下記のコードを試しましょう。

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

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

- 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

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)
```

mathモジュールをmとしてインポート

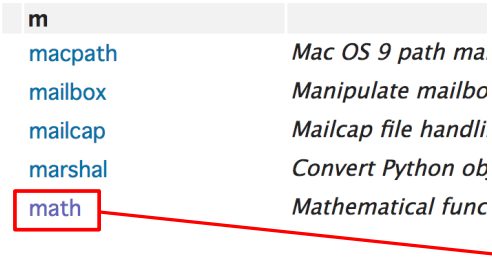
  - 例

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

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



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
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
外が発生します。その結果、どういった理由で例外が送出されたかに

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

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
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

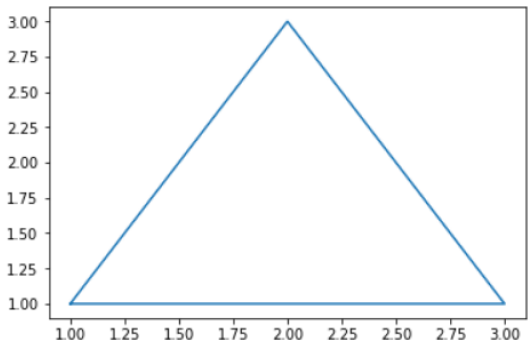
- 本章ではmatplotlibについて、次章以降で pandas, NumPy, SciPyについて演習を行います。

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
  - ・・・

```
import matplotlib.pyplot as plt
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)

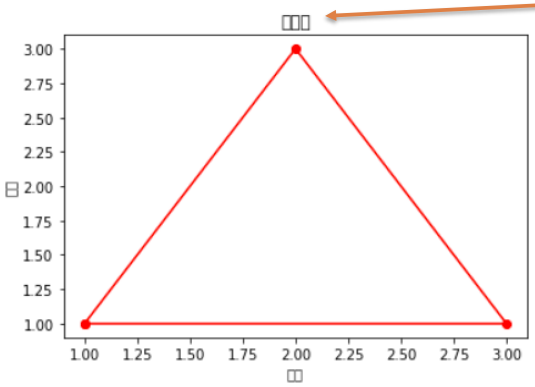
- マーカーや軸ラベルを設定してみます。

線の色　　マーカー

```
import matplotlib.pyplot as plt
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")
```

タイトルや軸ラベルの設定

文字化け！



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)
  - 実習室では、あらかじめ追加されています
  - 追加されていない環境での手順:
    - http://pana4405.u-shizuoka-ken.ac.jp/pp202102 からIPAフォント(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。
    - spyderを終了する。
    - [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 の .matplotlibフォルダ を削除する
    - spyderを再起動する

※正式な配布サイトはhttps://ipafont.ipa.go.jp/node26

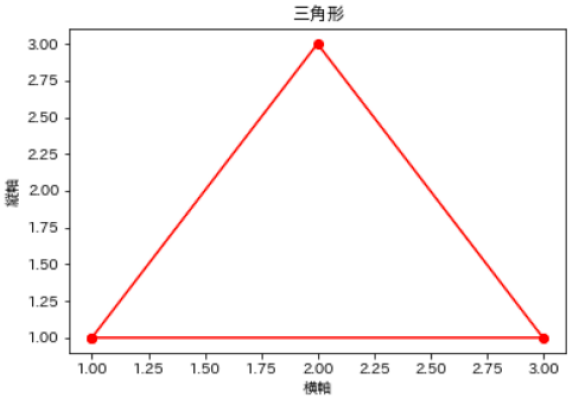
matplotlibを使ってみる(4)

```
import matplotlib.pyplot as plt

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

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")
```

IPAフォントの指定



課題7

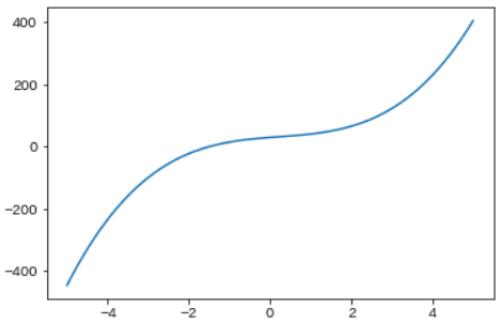
- matplotlibを用いて、

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

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

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

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

（補足）自作のモジュールをインポートする1

- 自作のモジュールから関数やクラスをインポート
  - モジュールの実態は複数のクラスや関数をまとめたファイル
    - 構文　from [モジュールファイル名] import [クラス／関数名]
    - 例：第6回で記述したPersonクラスをインポートする

```
from mymodule import Person
a = Person()
print(a.weight)
```

第6回で記述したmymodule.pyの場合、インポートした時点で「異常な体重です。」と表示されますが、これはmymodule.pyのクラス定義以外のコードが実行されるためです

  - モジュールのクラス・関数定義以外のコードがモジュール読み込み時に実行されないようにするには
    - モジュールのクラス定義や関数定義以外のコードの前に

```
if __name__ == "__main__":
    #クラスや関数以外のコード
```

というif文を入れ、クラスや関数以外のコードはそのブロック内に記述します

    - pyファイルをモジュールでなく直接実行すると__name__環境変数に”__main__”がセットされるので、if以下のコードは問題なく実行されます
    - モジュールとして読み込み時は、if文以下のコードは実行されません

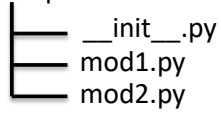
（補足）自作のモジュールをインポートする2

- 複数のモジュールをまとめたフォルダをパッケージと呼びます
  - パッケージ内の複数のモジュールを一度に認識させるためには、フォルダ内に__init__.pyファイルを用意し、

```
from . import mod1
from . import mod2
```

などと記述しておきます

例：sampleパッケージ



  - sampleパッケージを読み込んで関数を呼び出すには次のようにします

```
import sample
sample.mod1.hoge()
```

・・・mod1にhoge関数が定義されていると仮定