

第6回

モジュールとimport

目次

- モジュールとimport文
 - 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の一例
 - matplotlib
 - マーカーや軸ラベルの設定
 - 棒グラフ
 - 円グラフ
 - xlwings
 - Rangeメソッドによるセル操作
 - Excelのグラフ機能进行操作する
 - 主なグラフの種類と設定値
 - (補足)グラフにおける日本語の文字化け
- 課題6-1

モジュールとimport文

- 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          ← インポート
a = math.sqrt(2)    ← 平方根を計算する関数
print(a)
```

- 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。

- 3 -

from文

- import文に合わせてfrom文を使うと、関数やクラス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・クラス名
```

- 例

```
from math import sqrt
a = sqrt(2)          ← sqrt関数をモジュール名を省略して呼び出し
print(a)
```

- 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・クラスを呼び出せるようになります。

- 例

```
from math import *  ← *はワイルドカードとして機能
a = sqrt(2)
print(a)
```

- 4 -

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・クラス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・クラス名 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>

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

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
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

`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の一例

- 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	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(Excelファイルの読み書きが可能)
openpyxl	Excelファイルの読み書きに特化したモジュール
xlwings	Excelアプリを直接制御できるモジュール
scipy	NumPyを利用した数値解析モジュール
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

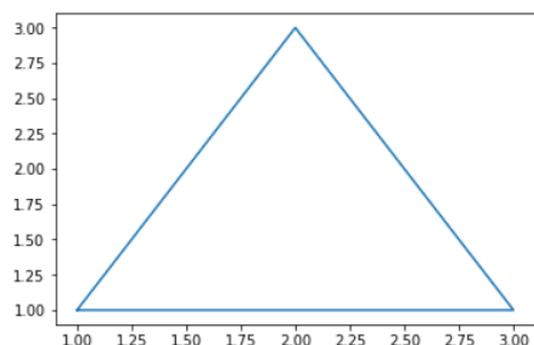
- 7 -

matplotlib

- 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
 - ...
- ファイルを新規作成し、折線グラフを描画するmatplotlib.pyplotのplot関数を使ってみる

```
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



- 8 -

マーカーや軸ラベルの設定

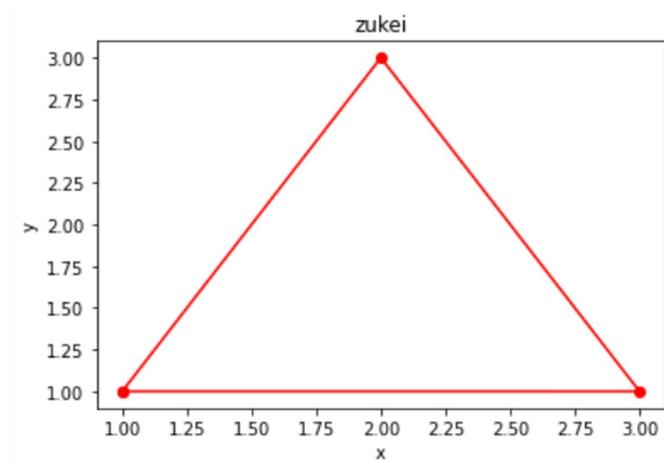
- マーカーや軸ラベルを設定してみます

```
import matplotlib.pyplot as plt  
plt.plot([1,2,3,1], [1,3,1,1], color="red", marker="o")  
plt.title("zukei")  
plt.xlabel("x")  
plt.ylabel("y")  
plt.show()
```

線の色 マーカー

タイトルや軸ラベルの設定

描画を確定させる

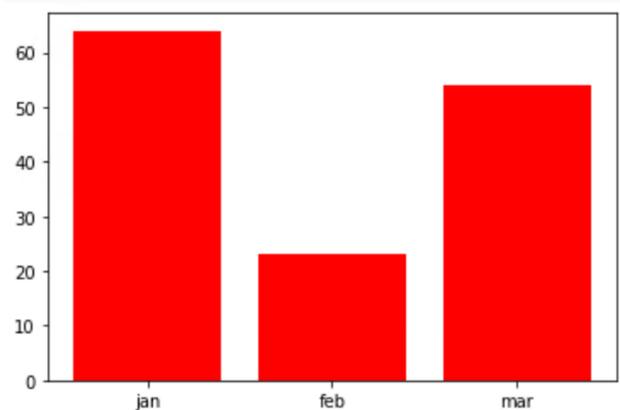


- 9 -

棒グラフ

- matplotlib.pyplotのbar関数で縦棒グラフが描画できます

```
x = ["jan", "feb", "mar"]  
y = [64, 23, 54]  
plt.bar(x, y, color="red")  
plt.show()
```



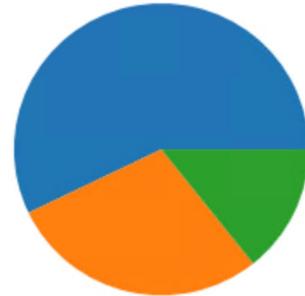
- bar関数を、barh関数に変更することで、横棒グラフが描画できます

- 10 -

円グラフ

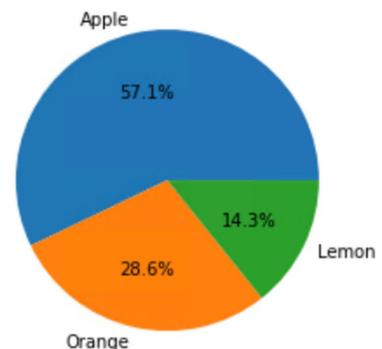
- matplotlib.pyplotのpie関数で円グラフが描画できます

```
x = ["Apple", "Orange", "Lemon"]
y = [200, 100, 50]
plt.pie(y)
plt.show()
```



- labels引数でラベルを表示できます
- autopct引数で%を表示できます

```
x = ["Apple", "Orange", "Lemon"]
y = [200, 100, 50]
plt.pie(y, labels=x, autopct="%1.1f%%")
plt.show()
```



.1fは小数点以下1桁を表している

- 11 -

xlwings

- xlwingsは、表計算ソフトExcelと連携するモジュールで、Excelを開きながらPythonでExcelを操作できます。

- Excelには、VBA (Visual Basic for Applications) 言語によるプログラミング機能が備わっており、定型的な作業の自動化に活用できましたが、xlwingsを使用することでVBAの代わりにPythonが使用できます。
- 2024年9月には、Microsoft社公式の「Python in Excel」の一般提供が開始されましたが、利用可能なライセンスがBusinessまたはEnterpriseサブスクリプションに限定されています。
- Anacondaでは、あらかじめxlwingsモジュールが同梱されているので、Excelと組み合わせてすぐに試すことができます。

- 12 -

Rangeメソッドによるセル操作(1)

- あらかじめ、Excelを起動して、データを入力しておきます

	A	B	C
1	料理	得票	
2	ハンバーグ	63	
3	カレー	54	
4	ラーメン	93	
F			

- Spyderにて、ファイルを新規作成し、下記コードを試してみましよう

- xlwingsを使用すると、セルの値がRangeメソッドで読み出せます

```
import xlwings as xw #モジュールのインポート
```

```
print(xw.Range("A1").value)  
print(xw.Range("A1:B4").value)
```



料理
[['料理', '得票'], ['ハン
バーグ', 63.0], ['カレー',
54.0], ['ラーメン', 93.0]]

- 13 -

Rangeメソッドによるセル操作(2)

- Rangeメソッドのvalueでセルの値を上書きできます

```
xw.Range("B4").value = 123
```



	A	B
1	料理	得票
2	ハンバーグ	63
3	カレー	54
4	ラーメン	123

- Rangeメソッドのformulaでセルに数式が設定できます

```
xw.Range("B5").formula = "=sum(B2:B4)"
```



	A	B
1	料理	得票
2	ハンバーグ	63
3	カレー	54
4	ラーメン	123
5		240

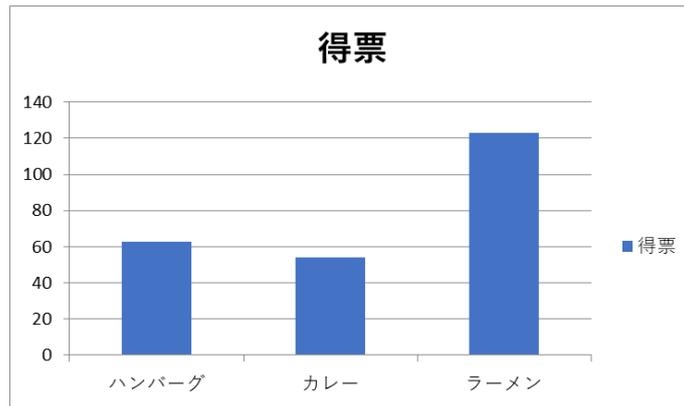
- 14 -

Excelのグラフ機能进行操作する(1)

- Chartクラスのオブジェクトを生成することで、Excelのグラフが操作できます

```
chart = xw.Chart()
chart.chart_type = "column_clustered"
chart.set_source_data(xw.Range("A1:B4"))
```

Chartオブジェクトの生成
グラフの種類(集合縦棒)
データの範囲



- 15 -

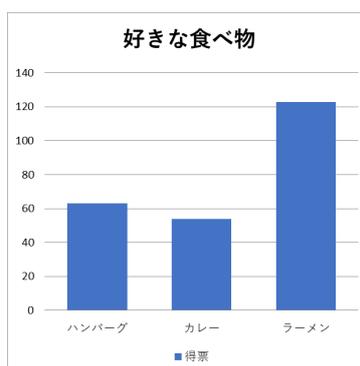
Excelのグラフ機能进行操作する(2)

- 配置やレイアウトも制御できます

```
chart = xw.Chart()
chart.chart_type = "column_clustered"
chart.set_source_data(xw.Range("A1:B4"))
```

```
chart.left = 100 # グラフの左座標
chart.top = 100 # グラフの上座標
chart.width = 300 # グラフの幅
chart.height = 300 # グラフの高さ
```

```
chart.api[1].ApplyLayout(3) # クイックレイアウト
chart.api[1].ChartTitle.Text = "好きな食べ物" # グラフのタイトル
```



- 16 -

主なグラフの種類と設定値

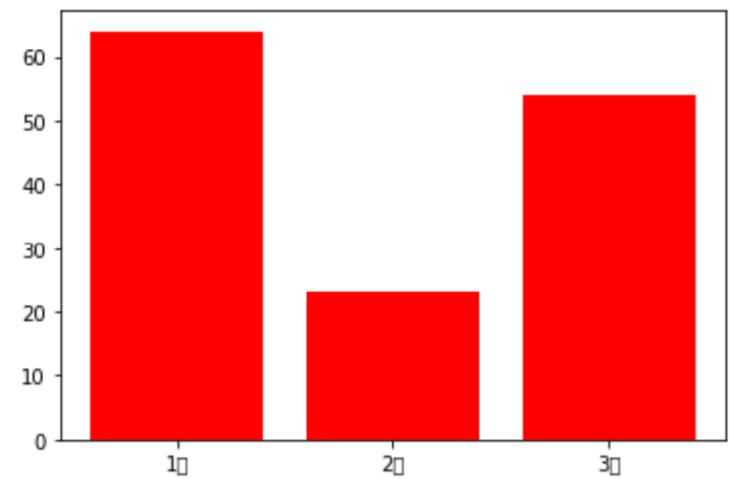
- 折れ線グラフ (line)
- 棒グラフ (column_clustered)
- 積み上げ棒グラフ (column_stacked)
- 横棒グラフ (bar_clustered)
- 積み上げ横棒グラフ (bar_stacked)
- 円グラフ (pie)
- ドーナツグラフ (doughnut)
- 散布図 (xy_scatter)
- エリアグラフ (area)
- レーダーチャート (radar)

- 17 -

(補足) グラフにおける日本語の文字化け

- Spyderでは、matplotlibで描くグラフのラベルなどに日本語が含まれると日本語フォントの設定が無い場合文字化けしてしまいます。

```
import matplotlib.pyplot as plt  
  
x = ["1月", "2月", "3月"]  
y = [64, 23, 54]  
plt.bar(x, y, color="red")  
plt.show()
```



文字化け

- 18 -

(補足)グラフにおける日本語の文字化け

- 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)
 - <https://moji.or.jp/ipafont/ipaex00301/>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をエクスプローラを使ってC:\ProgramData\Anaconda3\Lib\site-packages\matplotlib\mpl-data\fonts\ttfに入れる。上記フォルダが無い場合は、C:\Users\自分のユーザアカウント名\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。

- 19 -

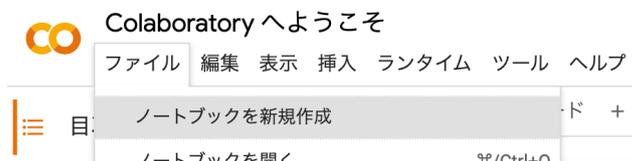
インストールレスのPython環境の紹介 (Google Colaboratory)

- Webブラウザを使ってPythonプログラムの作成と実行ができます
 - 要: Googleアカウント
- Edge/Chrome/Safariを起動する
 - google colaboratory で検索する
 - 下記リンクを開く

<https://colab.research.google.com> > notebooks > intro ▾

[Colaboratory - Google Colab](#)

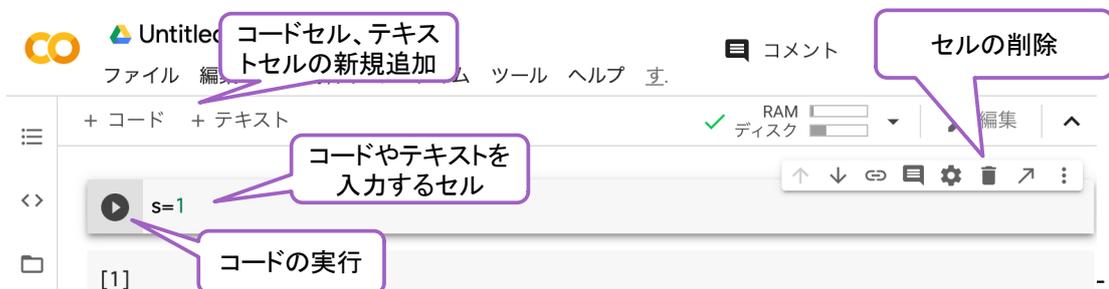
- 「ファイル」|「ノートブックを新規作成」



- 20 -

ノートブック

- Google Colaboratoryでは、ノートブックという単位でプログラムを記述します。
- セルという枠にコードを入力します。
 - コード以外にもテキスト(メモ書き)を入力するセルを追加できます。
- コードを実行するには、セル左の ▶ ボタンか、下記のキーを押します。
 - 実行: Ctrl+Enter
 - 実行と新たなセルの追加: Shift+Enter



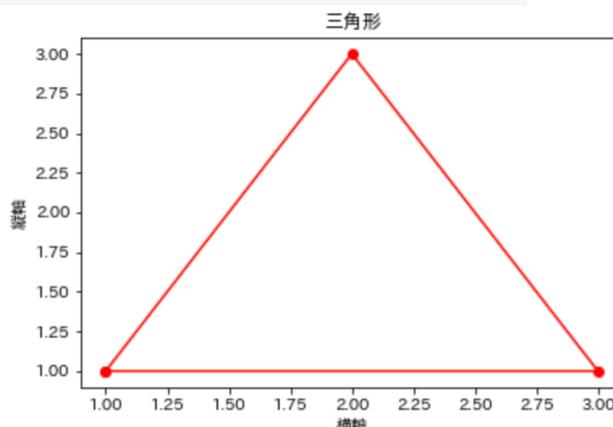
Google Colaboratoryの文字化け対策

- matplotlibで日本語を使えるようにするモジュール「japanize-matplotlib」をインストールする。

```
!pip install japanize-matplotlib
import matplotlib.pyplot as plt
import japanize_matplotlib
```

japanize-matplotlibのインストール

```
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")
```



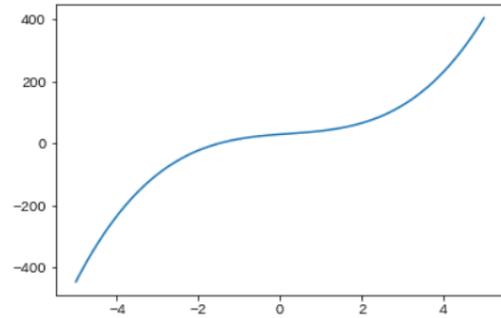
課題6-1

- 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
xp = []
yp = []
for x in range(-50, 50):
    xp.append(x * 0.1)
    yp.append(ここに考える)

plt.plot(xp, yp)
plt.show()
```