




Python入門 第1回

総合科学研究会

2024.4.11



このゼミでやること・目標

- プログラミング言語のひとつであるPython(パイソン)に触れる。
- データ型、条件分岐、繰り返し、関数、オブジェクト(配列)といった基本的な概念を習得する。
- プログラミング言語による開発(の真似事)を体験する。

テキスト

- ゼロからのPython入門講座

<https://www.python.jp/train/index.html>

基本的にはこのサイトを説明に用います。

予定

- 第1回

準備・Python基礎

- 第2回

文字列・入出力

- 第3回

条件分岐①

- 第4回

条件分岐②・論理演算子

- 第5回

繰り返し①

予定

- 第6回

繰り返し②・関数①

- 第7回

関数②・オブジェクト①

- 第8回

オブジェクト②

- 第9回

タプル・コレクション

- 第10回

総合演習

プログラミング言語とは

- 何らかの目的を達成するための、コンピュータを操作する言語。
- 英語や日本語といった自然言語とは異なり、決まった用法がある。
- 現在、多くのプログラミング言語が存在するが、基本的にはどの言語も同じような事が出来る。ただし、得意不得意がある。

例えば、WebアプリケーションならJavaScriptやPHP、ローカルソフトウェアならC++やJava、機械学習ならPython等...

プログラミング言語を学ぶ事で出来る事

- ソフトウェア・アプリケーションの開発

Webアプリ・ゲーム

- AIの開発

機械学習(画像や音声認識、生成モデル)

- 科学技術計算への応用

物理シミュレーション

プログラミング言語Python

- 初心者でも十分に使うことが出来るプログラミング言語
- Pythonには便利なツール(ライブラリやフレームワーク)が存在するので、近年では様々な場面で用いられる。
- ただし、他の言語にはない記法や、不自然な挙動をする場合が少なからず存在するので、経験者の中にはPythonの使用を好まない人も存在する。

第1回でやること

- 実行環境を整える
- とりあえずPythonに触れてみる

インタプリタについて

普通プログラムのコード(ソースコード)を書く時は

- 何らかのテキストエディタを使って書いて、それをファイルに保存
- コンパイラという特別なソフトウェアにコードを読み込ませて実行

という手順を踏むが、インタプリタではコードを書いたらすぐに実行され、結果が返ってくる。

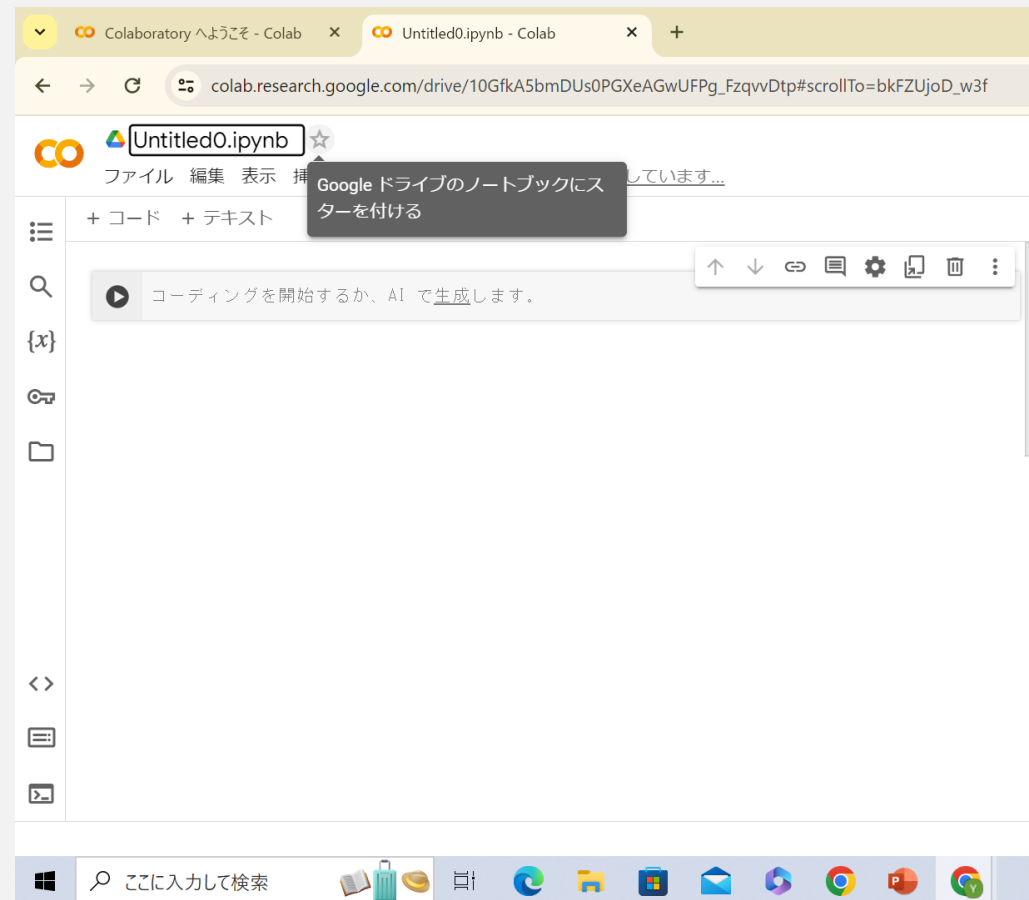
このゼミでは、Google colabというPythonインタプリタを使うことが出来る環境で演習を行います。

実行環境を整える

- Googleアカウントを作成する。(持っている場合はスキップ)
- <https://colab.research.google.com/>にアクセスする。

- 画面左上の“ファイル”から「ノートブックを新規作成」を選択

ノートブックに、Pythonのコードを書いていくイメージ



テスト

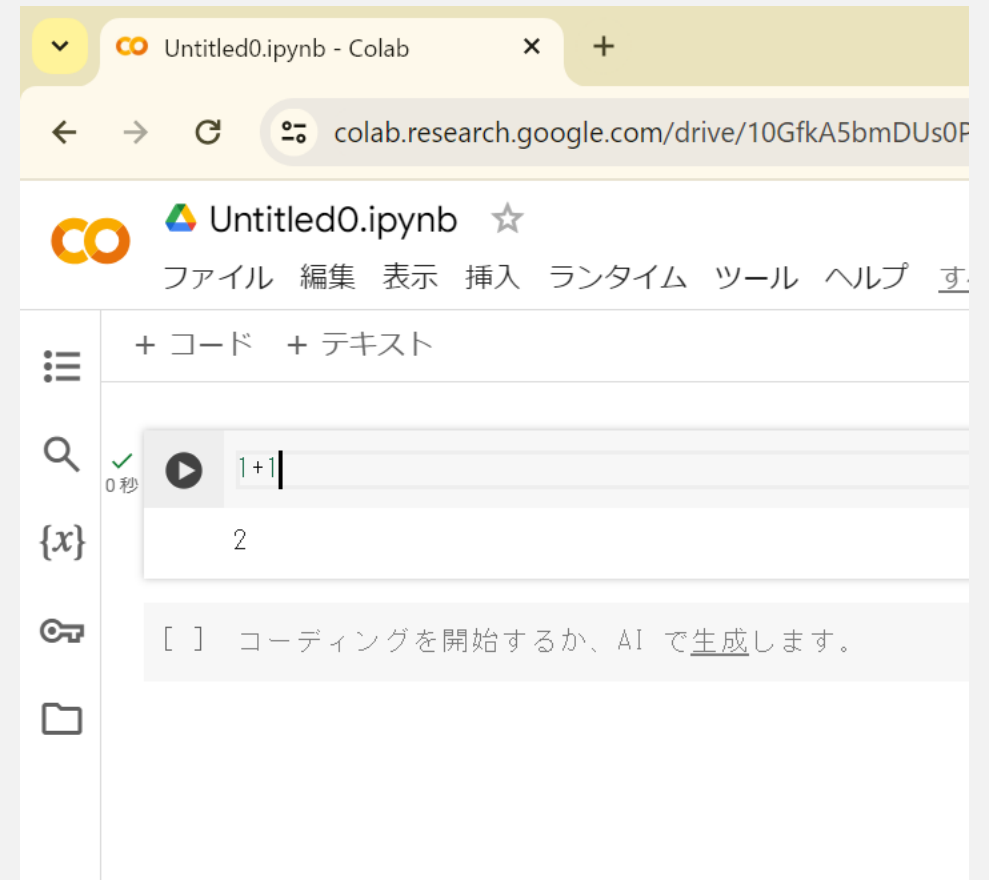
- ノートブックを作成したら白枠で囲われた部分(セルという)に

1+1

と入力して実行(▶を押すかCtrl+Enter)する。

しばらくして2と表示されたら成功。

失敗した場合→半角英数で入力しているから
どうか確認



セルの追加など

左上の「+コード」、もしくは先ほどと同じく白枠を押すと新しいセルが追加される。

右クリックを押すと出てくるメニューからセルを削除できる。

「+コード」の横にある「+テキスト」を押すとメモ用のテキストセルが今現在選択しているセルの下に表示される。



テスト②

- もう少し面白い実行例を試す
- <https://www.python.jp/train/experience/next-sample2.html>

ここから左のようなコードをコピーしてセルに貼ってみる。

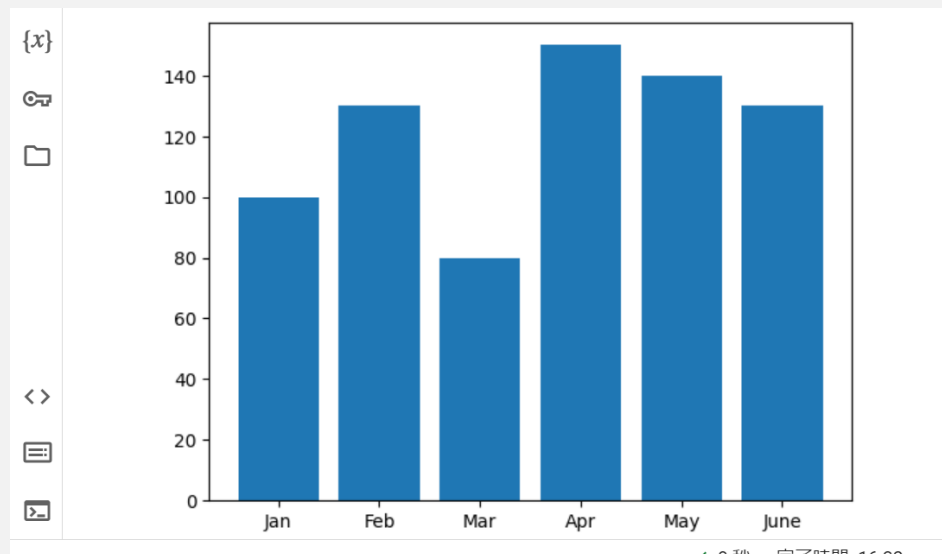
下のような実行結果になるはず。

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

```
x_values = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'June']  
y_values = [100, 130, 80, 150, 140, 130]
```

```
plt.bar(x_values, y_values)  
plt.plot()
```

```
plt.show()
```





四則計算を実行させる



算術演算

- 以下のコードをセルに書いてみる

```
▶ 20 + 24 * 4 - 11|
```

結果が105であれば正しい

Pythonでは右の図のように、算術演算が定義されている。

べき乗 $a ** b$ は a^b の事

演算	式
足し算	$a + b$
引き算	$a - b$
かけ算	$a * b$
割り算	a / b
余り	$a \% b$
べき乗	$a ** b$

演習①

- 以下の計算を実行してみてください。

(1) $100 / 20$

(2) $255 \% 12$

(3) $10 * (-7 + 4)$

(4) $25 ** 3$

整数と浮動小数点

- 先ほどの計算では登場する数字が全て整数だったが、小数点についても扱うことが出来る。

例えばコンピュータは“10”と書かれた数値を整数(integer)として扱う。一方で小数点を付けた“10.0”については値としては同じでも、浮動小数点型(floating point)と呼ばれる別のデータ型として扱う。

Pythonにおいて、整数と実数同士の四則演算の結果は以下ようになる。

整数同士の計算→整数

浮動小数点絡む計算→浮動小数点

変数

変数とは値を保持する記号のようなものである。

例えば

```
num = 23
```

とすると、変数numに23という値を代入することが出来る。

以上の操作を変数を定義するという。

定義した変数はその行より下において使うことが出来る。

```
num = 10
```

```
num2 = num + 2
```

とすると、num2には $\text{num}(10)+2=12$ が代入される。

代入操作の注意

- プログラミング言語において等号" $=$ "は右辺の値を左辺の変数に代入するという意味である。

例えば

```
num = 12
```

```
num = 2*num + 5
```

とすると、2行目の`num = 2*num + 5`は1行目で定義した`num=12`に2を掛けて5を足したものを改めて変数`num`として定義するという意味である。

数学の方程式とは全く異なるので注意！