Pythonプログラミング入門

Pythonの基本文法と制御構造、リスト操作、外部ライブラリを用いた数値計算とグラフ描画

基本文法

変数、演算子、入出力処理

制御構造

条件分岐と繰り返し

リスト

複数データの格納

外部ライブラリ

NumPy、Matplotlib

前提：Webブラウザの基本操作

意義：プログラミングによる　問題解決の基礎力

開発環境

GDBonline：オンライン開発環境

インストール不要

ブラウザで開始

プログラム作成・修正 → Runボタンで実行 → Stopボタンで中断

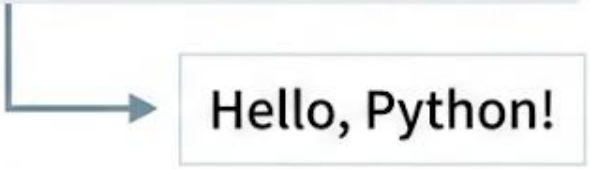
保存：拡張子 .py

プログラムの基本要素

コード

print関数: 画面に文字を表示する関数

```
print("Hello, Python!")
```

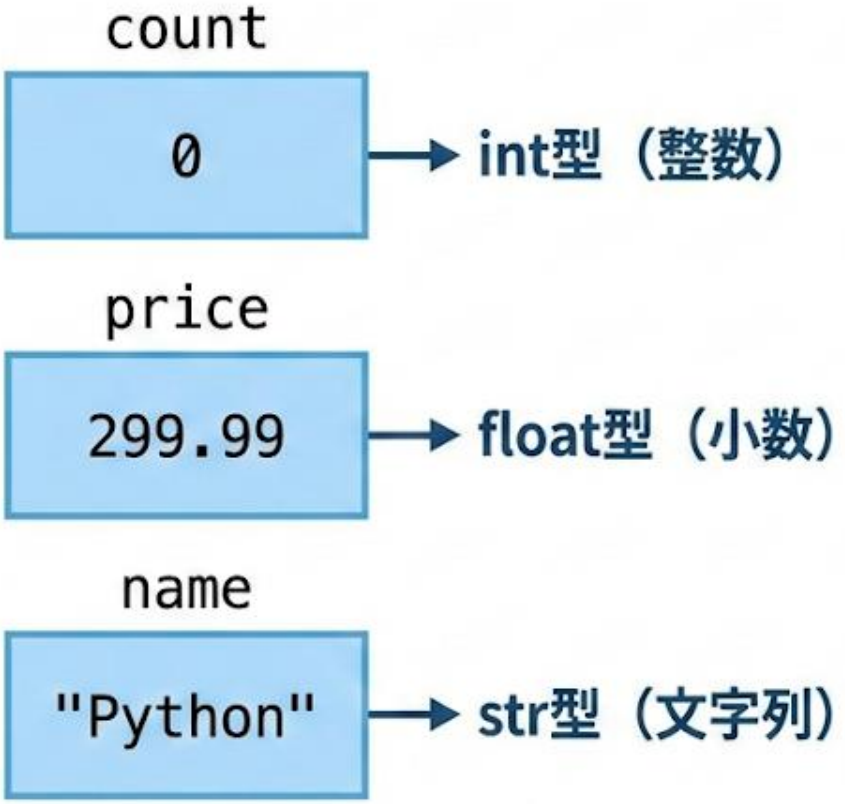


コメント

#から行末まで、実行時に無視

```
x = 10 # 変数xに10を代入
```

変数

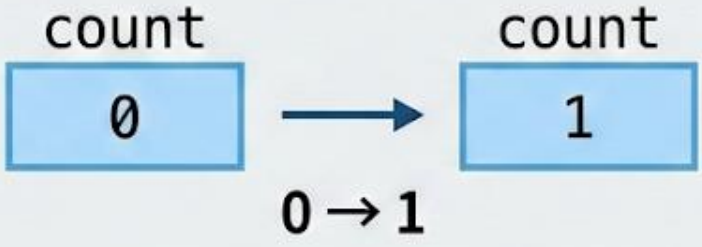


変数：データを格納するメモリ領域
型宣言不要、代入で作成

```
count = 0
price = 299.99
name = "Python"
```

値の更新

```
count = count + 1
```



新しい値を代入できる

演算子

四則演算

+	加算
-	減算
*	乗算
/	除算
//	整数除算
**	べき乗

比較演算子

<	小なり
>	大なり
==	等しい
!=	等しくない
<=	以下
>=	以上

結果：True または False

論理演算子

and：両方が真のとき真

or：いずれかが真のとき真

not：真偽を反転

出力処理

print関数

print関数：表示して改行

```
print("Hello")
```

→ Hello

str関数による変換

str関数：数値を文字列に変換

```
total = 100
print("合計：" + str(total))
```

100 → "100"

→ 合計：100

f文字列

f文字列：変数の埋め込みと書式指定

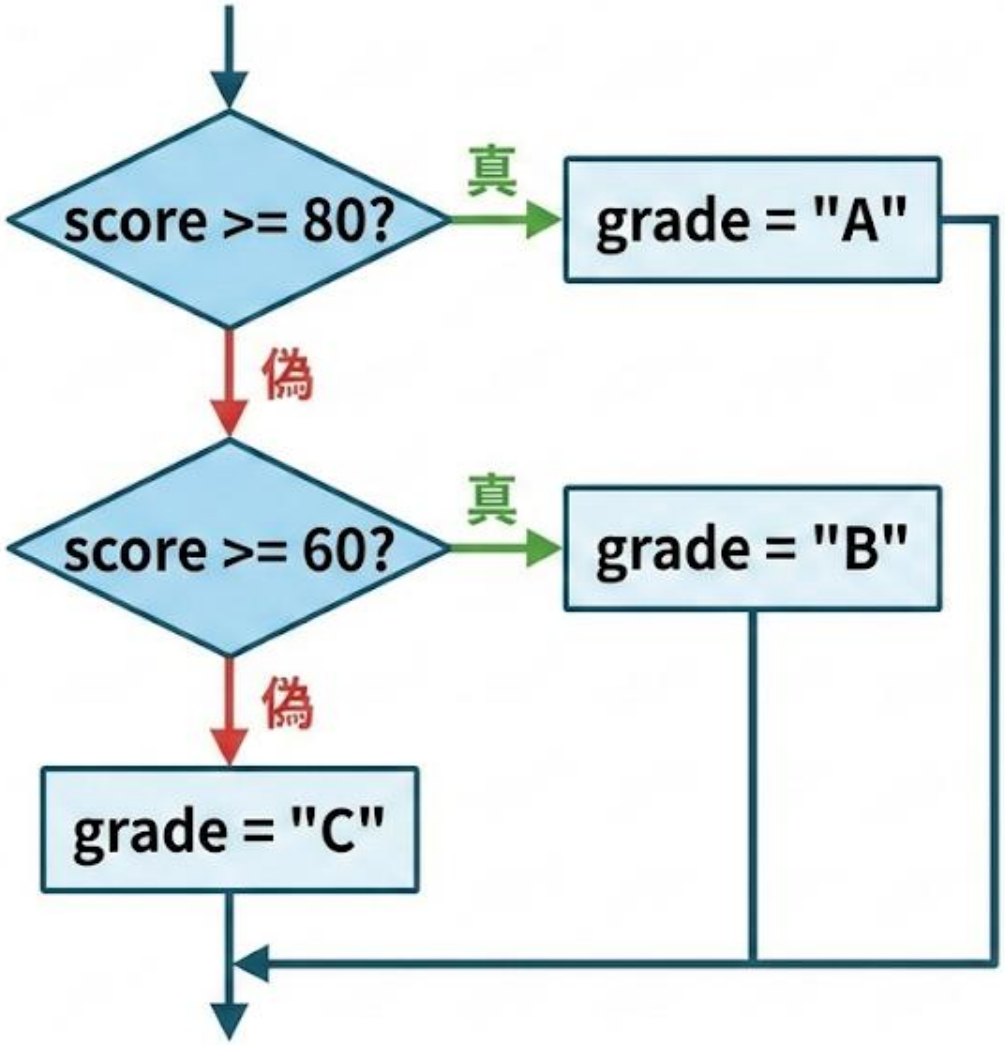
```
x = 3.14159
print(f"x = {x}")
print(f"x = {x:8.3f}")
```

→ x = 3.14159

→ x = 3.142

全体8桁、小数点以下3桁

条件分岐（if文）

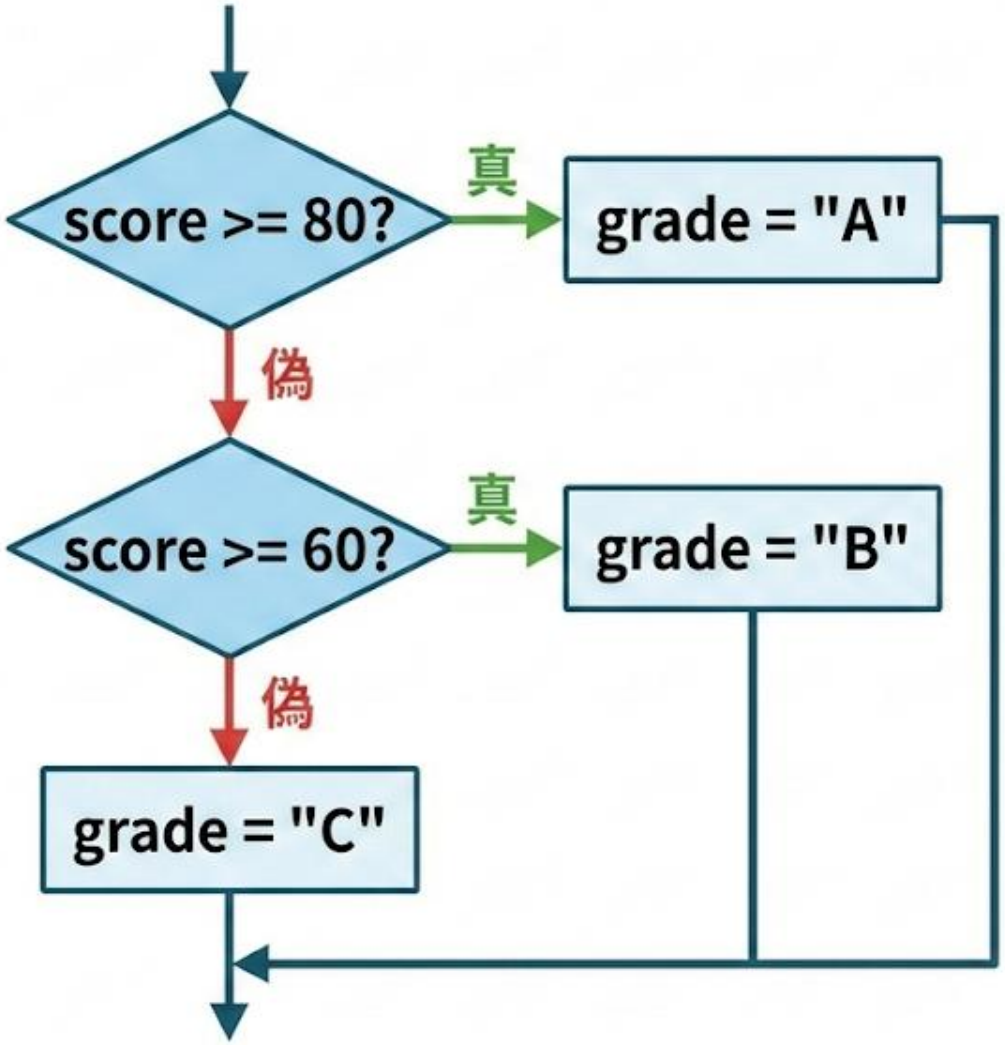



```
if score >= 80:
    grade = "A"
elif score >= 60:
    grade = "B"
else:
    grade = "C"
```

if：条件が真のとき実行
elif：複数条件の分岐
else：条件が偽のとき実行

注釈：インデント：空白4つで処理のまとまりを示す（必須）

条件分岐（if文）




```
if score >= 80:
    grade = "A"
elif score >= 60:
    grade = "B"
else:
    grade = "C"
```

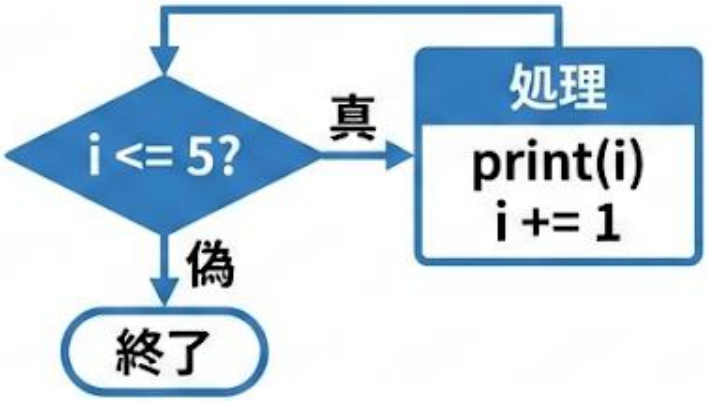
if：条件が真のとき実行
elif：複数条件の分岐
else：条件が偽のとき実行

注釈：インデント：空白4つで処理のまとまりを示す（必須）

繰り返し（while文・for文）

while文

条件が真の間、繰り返す

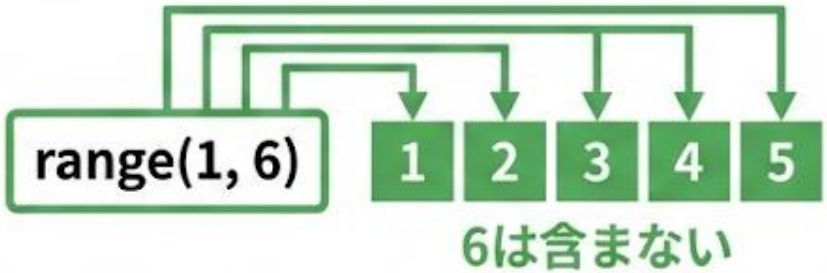



```
i = 1
while i <= 5:
    print(i)
    i += 1
```

i += 1 は i = i + 1 と同じ

for文

回数が決まっている繰り返し




```
for i in range(1, 6):
    print(i)
```

出力結果

1　2　3　4　5

リスト


```
numbers = [0,0,0,0,0]
numbers[0] = 10
```

```
for element in numbers:
    print(element)
```

```
10 0 0 0 0
```

- リスト：複数データを順序付けて格納
- インデックス：0から開始
- len関数：要素数を返す
- コード： numbers = [0] * 5; numbers[0] = 10; for element in numbers:

数学関数（mathモジュール）

```
import math
```

関数例（コード）	戻り値・説明
`math.floor(3.7)`	3（切り捨て）

`math.sin(x)`
`math.cos(x)`
`math.tan(x)`
	三角関数（ラジアン単位）

`math.exp(x)`	eのx乗（e≈2.718）
`math.log10(100)`	2.0（常用対数）

コード例

```
sin30 = math.sin(30 * math.pi / 180)
```

注釈：度→ラジアン変換：× math.pi / 180

import math：モジュールのインポート
三角関数：引数はラジアン単位
度→ラジアン：× math.pi / 180

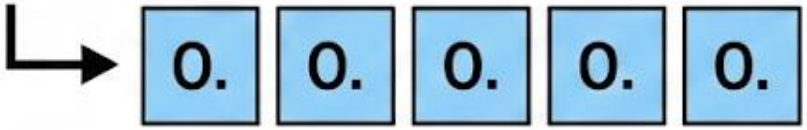
import numpy as np

NumPy

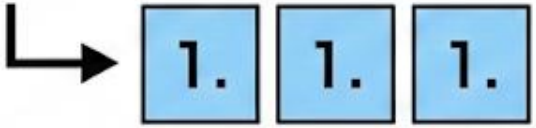
配列作成

np.array([1, 2, 3, 4, 5])

np.zeros(5)

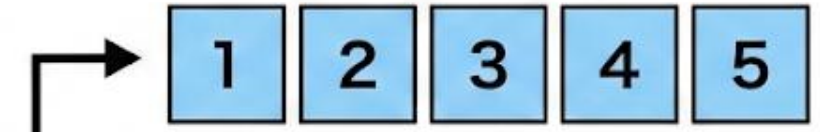


np.ones(3)



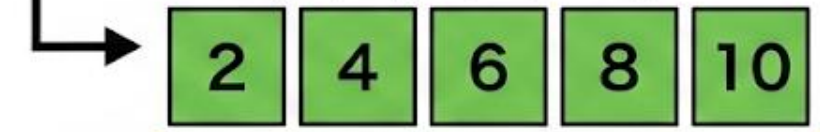
ブロードキャスト

a = [1, 2, 3, 4, 5]



×2

b = [2, 4, 6, 8, 10]



ブロードキャスト：全要素に演算が適用

- NumPy：数値計算用ライブラリ
- np.zeros：0の配列
- np.ones：1の配列
- ブロードキャスト：全要素に演算適用

コード例

```
import numpy as np
a = np.array([1, 2, 3, 4, 5])
b = a * 2
```

Matplotlib

import matplotlib.pyplot as plt

Matplotlib：グラフ描画ライブラリ

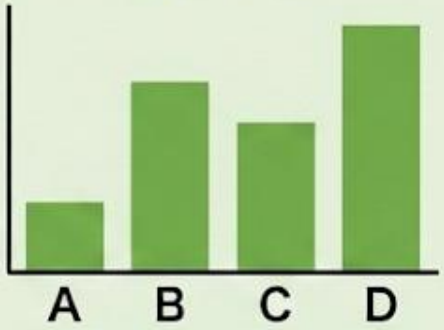
plt.plot

折れ線グラフ

plt.plot(x, y)

plt.bar

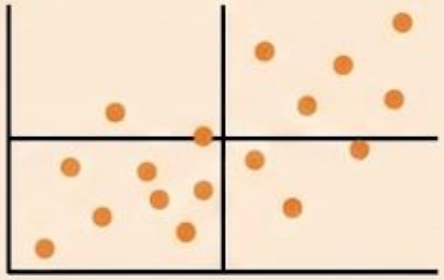
棒グラフ



plt.bar(labels, values)

plt.scatter

散布図



plt.scatter(x, y)

plt.show：画面に表示

コード

```
x = [1, 2, 3, 4]
y = [1, 4, 9, 16]
plt.plot(x, y)
plt.show()
```

装飾関数

plt.title()：タイトル
plt.xlabel()：x
plt.ylabel()：y

全体まとめ：Pythonプログラミング入門

開発環境

- GDBonlineなど

基本文法

- print関数
- コメント
- 変数
- 演算子

入出力処理

- 出力（print、f文字列）
- 入力（input、型変換）

制御構造

- 条件分岐（if-elif-else）
- 繰り返し（while、for）

データ構造

- リスト（インデックス0から）

ライブラリ

- 標準（math）
- 外部（NumPy、Matplotlib）

本資料の図解は Nano Banana Pro を用いて作成