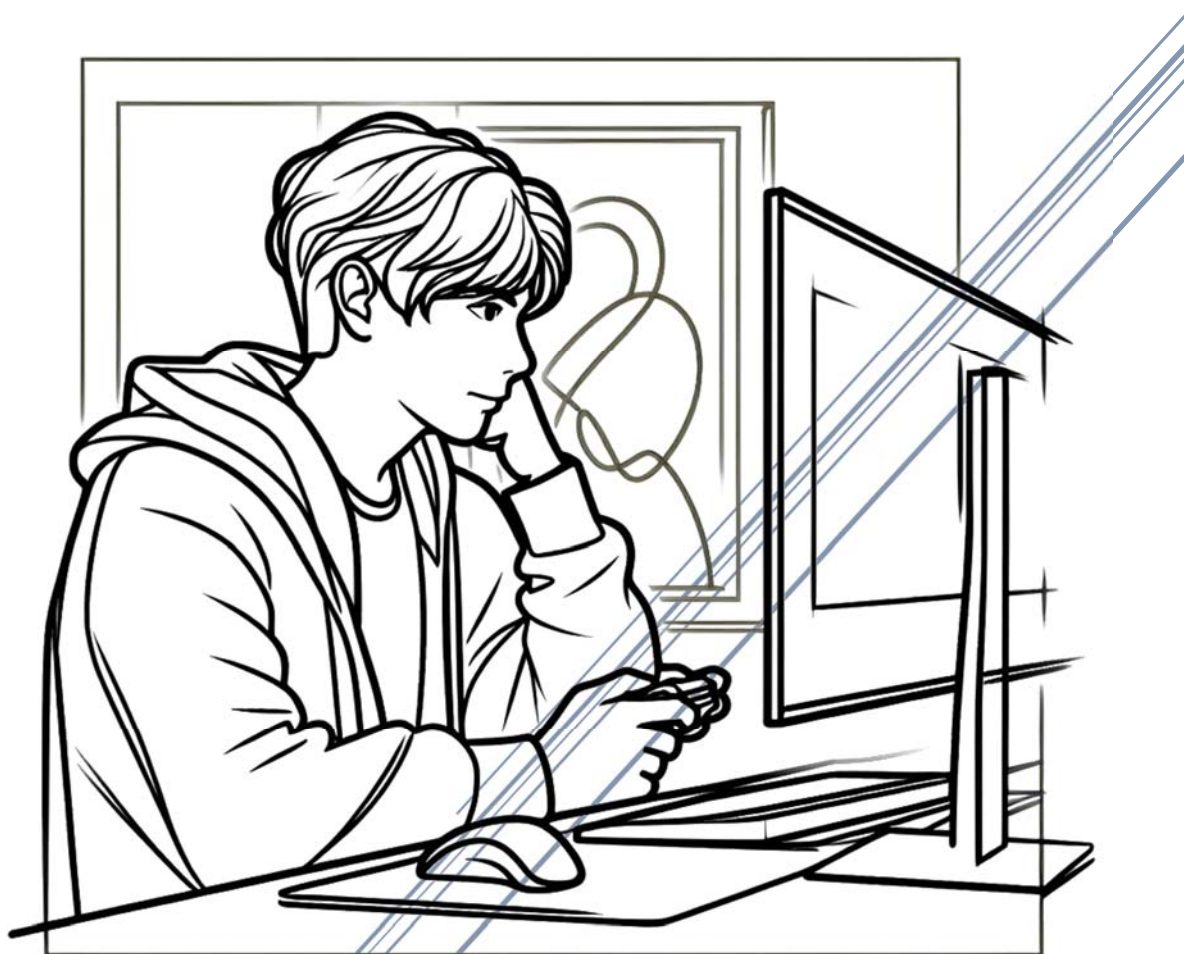


# ゼロから始めるプログラミング

プログラムをやったことのない人大集合!



東京 IT プログラミング & 会計専門学校杉並校

# 1章 プログラムについて

## 1. プログラミングとは

繰り返しや計算をたくさんやること→人間にとってはイヤ、コンピュータは得意  
作業手順をコンピュータに指示→プログラミング

そもそもプログラムとは？

→運動会のしおり→競技の順番

コンピュータ→機械にわかる文法→究極的には0 (OFF)と1 (ON) のみ  
機械と人間に近い感じで指示できる→プログラム言語

身近なプログラム

炊飯器→美味しいご飯を炊く手順

信号機や自動販売機にも



プログラムのメリット

ストーブ→人間がスイッチのオンオフ

プログラムがあれば、最適な温度、時間が来れば OFF

## 2. インタープリタとコンパイラ

インタープリタ (同時通訳者) →一行ずつ訳す (Python など)

コンパイラ (翻訳) →まとめて訳す (Java など)



## 3. いろいろなプログラミング言語

- Python→多くの Web サービスや AI の分野で使われています。
  - Java→あらゆる部分で使われている現在主流のプログラミング言語です
  - C/C#→一般のアプリやゲーム開発など
  - JavaScript→ホームページの表示などで使われています。
  - Swift→iOS (iphone など) のプログラム制作に使われています。
- 
- HTML→ホームページの文書を記述するもの→プログラム言語ではない
  - SQL→データベースを操作するための言語

## 4. Python (パイソン) について

人工知能 (AI)が実用化されてきました。最新の流行に対応しつつ、初心者でも使いやすいプログラミング言語が Python です。

特徴

- ・書き方がシンプル
- ・あらかじめプログラムに色々な機能が実装されている
- ・多くの企業が便利な機能を公開してくれている



## 5. Python の実行環境

- ・公式サイトから無料で最新版がダウンロードできます。

<https://www.python.org/>

- ・開発環境も色々あります。microsoft の Visual Studio Code がおすすめ

<https://azure.microsoft.com/ja-jp/products/visual-studio-code>

このセミナーではブラウザから実行できる [online-python.com](https://www.online-python.com/) のサイトを利用していきます。

(ダウンロードやインストールは不要です。)

<https://www.online-python.com/>

## 6. Python プログラムを作ってみよう！

### プログラム 1-1

①上の欄に書いてあるプログラムを選択して消して以下のプログラムを書いてみましょう

```
print(1+1)
```

②書き終わったら下の RUN ボタンを押してみよう！（F8 キーを押してみてもいいよ）  
すると下のウインドウに結果が出ます。

### プログラム 1-2

引き算もやってみよう！

```
print(2024-2006)
```

### プログラム 1-3

文字列も表示させてみよう

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

## 2章 プログラムの作り方について

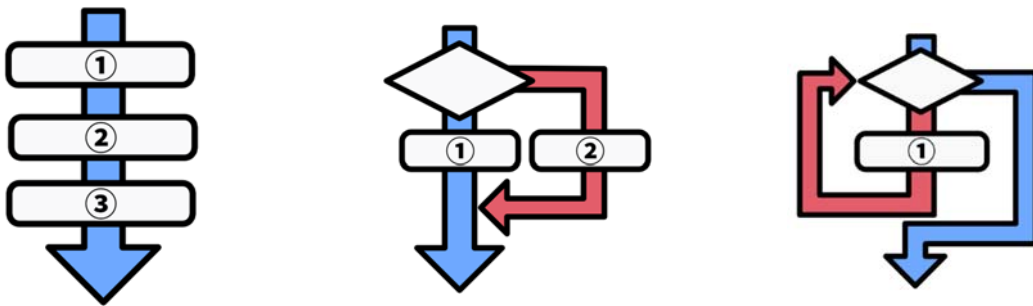
### 1. プログラムの構造は3つ

- ①順次
- ②条件分岐（判断）
- ③繰り返し（ループ）

-----

- ①順次→基本的にプログラムは全て並べた順に実行される
- ②条件→何らかの例外処理がしたいときだけ、その「条件」によって実行先を「分ける」ことができます。
- ③繰り返し→特定の回数や条件を満たすまで、一定の区間を「繰り返し」て処理を行わせることができます。

どんな複雑なプログラムのこの3つで構成されています



### 2. アルゴリズム

何らかの問題を解決するために使うものの考え方を「アルゴリズム」と言います。

目的を達成するためのロジック、ストーリーの事です。

アルゴリズムを組み立てるための使う制御方法が「順次」「条件分岐」「繰り返し」の3つです。

## 3章 変数

今まで作ったプログラムは直接、文字や計算結果を出力していましたが、何度も同じ文字を使ったり、複雑な計算をしたいときには一時的に値を格納する容器である「変数」を利用します。

変数には入れる値によって適応した「型」が存在します。（水を入れたり、お湯を入れたり、粉をいれたりする際に合った容器を使いますよね）

主な変数

- int 整数→インテージャーの頭文字
- float 浮動小数点数→フローティングポイントの略 小数点を含む数字を使うときはこれ
- str 文字列→ストリングスの略 文字を扱うときはこれ
- bool ブール→True（真）か False(偽)を判定するために、この2つしか入らない変数

### 1. 変数の使い方

a = 5 →変数 a に 5 を代入します。（型は自動で int 型にしてくれます）

b = 3.14

-----

こんな使い方もできます

c = "おはよう"

print(c)

### 2. 入力させる命令を使ってみよう！

キーボードから入力した文字や数値をプログラムで活用するための命令が「input」です

<文法>

変数 = input("画面に表示させる文字列など")

#### プログラム 3 - 1

```
a = input("名前を入力＞ ")
print(a, "さんですね。")
```

### 3. 計算式を使ってみよう！

せっかくなので、コンピューターの得意な計算をやってもらいましょう！

a に 1 を入れて 1 を足す式

```
a = 1
```

```
a = a + 1
```

-----

+の他にも演算子には下記のものがあります。

+ (足し算)、- (引き算)、\* (掛け算)、/ (割り算)

他にも //(割り算で小数点を切り捨てたもの)、% (割り算の余り)、\*\* (べき乗)

### プログラム 3 - 2

東京ドームでの換算プログラムを作ってみよう！

(東京ドームは 0.047 km<sup>2</sup>です。)

```
a = float(input("東京ドームと比較したい広さをkm2で入力して> "))
b = a / 0.047
print("東京ドーム",b,"個分の大きさです")
```

ちなみに東京ディズニーランドは 0.51 km<sup>2</sup>ですよ

## 4章 選択処理を行ってみましょう！

### 1. if 文

選択処理は「if 文」を使います。

<書式>

if 条件:

    処理 1 (条件満たしているとき)

else :

    処理 2 (条件満たしていない時)

※条件の後の : (コロ) 必要

※処理 1 の字下げ重要 (下げてある間は処理の対象)

<条件で使う比較演算子>

==(同じだったら)、!=(左辺と右辺が異なっていたら) < (右が大きかったら) <= (左が右以下)

>, > =

## プログラム 4-1

```
a = int(input("テストは何点? > "))
if (a < 40):
    print("赤点です")
else:
    print("セーフです")
```

※複数の条件を指定する elif という書き方もできます。

```
if 条件 1 :
    処理 1
elif 条件 2 :
    処理 2
```



## 5章 繰り返し処理をやってみましょう！

### 1. For 文

1 つめ：繰り返し処理は「for 文」を使います。

<書式>

```
for 変数 in range(回数):  
    繰り返す処理
```

※回数の中の：必要

※字下げしてある文を繰り返し処理します。

#### プログラム 5-1

```
for i in range(3):  
    print("Hello!")
```

※変数 i は 0 からスタートして、0,1,2 の 3 回実行されています。

### 2.while 文

2 つめ：「while 文」を使った繰り返し処理もあります。

<書式>

```
while 条件式:  
    繰り返し処理
```

※for 文のように自動的にカウントアップはしません。

(繰り返し処理の中に書く必要があります。)

※条件式に True と書くと永久に繰り返します。(抜け出す時には break 文を書きます。)

#### プログラム 5-2

```
print("アノ人のような返し、おわる時は「さようなら」と入力")  
while True:  
    a = input("あなた : ")  
    if a == "さようなら":  
        break  
    print("コンピュータ : ",a,"と言いましたが、それってあなたの感想ですよ")
```

## 6章 ライブラリの活用

### 1. ライブラリとは？

「誰か他の人が作ったプログラムの部品（関数）をまとめたもの」です。

Python 標準で用意されているものとしては

日時を操作する `datetime` や毎回異なる値を発生させる `random` などがあります。

`random` 関数を使った占いプログラムを作ってみましょう

#### プログラム 6 – 1

```
import random
a = ["Very Happy","Happy","Useally","Bad"]
print("今日の運勢は",random.choice(a))
```

## 7章 簡単なゲーム

#### プログラム 7 – 1 数当てゲーム

<ゲームの概要>

このゲームでは、プログラムが 1 から 10 までの間でランダムな数を選びます。

ユーザーはその数を 3 回以内に当てる必要があります。

各予想の後で、ユーザーは「もっと大きい数です」とか「もっと小さい数です」というヒントを得ます。

3 回の試行後に数を当てられなければ、正解が表示されます。

```
import random

# 1 から 10 までのランダムな数を生成
kotae = random.randint(1, 10)
kaisu = 3

print("数当てゲームへようこそ！ 1 から 10 までの数を当ててください。")

# ユーザーが 3 回まで推測できる
```

```

while kaisu > 0:
    yosou = int(input("あなたの予想: "))

    if yosou == kotae:
        print("正解です！おめでとうございます！")
        break
    elif yosou < kotae:
        print("もっと大きい数です。")
    else:
        print("もっと小さい数です。")

    kaisu -= 1

if kaisu == 0:
    print("残念！正解は" , kotae , "でした。")

print("ゲーム終了。また挑戦してください！")

```

## 🚩 プログラム 7-2 犯人を追え！（コピペでどうぞ）

<ゲームの概要>

刑事が犯人を探すゲームです。

ゲーム盤上にランダムに配置された犯人と刑事の位置を考慮して、犯人を見つけるまで刑事が移動します。

犯人から刑事までの距離をヒントとして表示します。

```

import random
import math

def calc_distance(x1, y1, x2, y2):
    # 2点間の距離を求める
    diff_x = x1 - x2
    diff_y = y1 - y2

```

```

return math.sqrt(diff_x**2 + diff_y**2)

hannin_x = random.randrange(0, 5) # 犯人の x 座標
hannin_y = random.randrange(0, 5) # 犯人の y 座標

keiji_x = random.randrange(0, 5) # 刑事の x 座標
keiji_y = random.randrange(0, 5) # 刑事の y 座標

# 犯人と刑事の位置が異なる間、処理を繰り返す
while (hannin_x != keiji_x) or (hannin_y != keiji_y):

    # 犯人と刑事の距離を表示する
    distance = calc_distance(keiji_x, keiji_y, hannin_x, hannin_y)
    print("犯人への距離:", distance)

    # キー入力に応じて、刑事を移動する
    c = input("n:北に移動 s:南に移動 e:東に移動 w:西に移動")
    if c == "n":
        keiji_y = keiji_y - 1
    elif c == "s":
        keiji_y = keiji_y + 1
    elif c == "w":
        keiji_x = keiji_x - 1
    elif c == "e":
        keiji_x = keiji_x + 1

print("犯人を捕まえました!")

```

おまけ (タイピングゲーム)

<ゲームの概要>

ランダムな単語が表示され、ユーザーがそれをできるだけ早くタイプするゲームです。

```
import random
import time

words = ["apple", "banana", "cherry", "dog", "elephant", "frog"]
random_word = random.choice(words)

print("タイピングゲームを始めます！")
input("Enter キーを押してスタート...")

start_time = time.time()
user_input = input(f"{random_word}: ")
end_time = time.time()

if user_input.lower() == random_word:
    elapsed_time = end_time - start_time
    print(f"正解！タイム: {elapsed_time:.2f}秒")
else:
    print("間違いです。")
```