



pf-1. プログラミングとは

(Python プログラミング入門演習, 全6回)

<https://www.kkaneko.jp/cc/pf/index.html>

金子邦彦





アウトライン

1-1 全体計画

1-2 プログラミング

1-3 さまざまなプログラミング言語

- プログラミングを学ぶにあたって、必要な知識を得る



1-1 全体計画



この授業は

- コンピュータはプログラムで動く
- この授業では, プログラム作成 (プログラミング) を学ぶ
- Python 言語のプログラミングの入門を行う



この授業の 進め方

- プログラミングの入門者を歓迎
- プログラミングの主要知識
式, 変数, 条件分岐, リスト,
繰り返し, 式の抽象化, 関数
- パソコンを使用
 - オンラインサービスを利用.
 - Web ブラウザで動く.
 - 自宅でも学習可能.
- 体験型学習 (ハンズオン)
 - 見本となるプログラムを実行させて学ぶ.

6回の計画



1	プログラミングとは（開発環境，さまざまなプログラミング言語）
2	Python 言語，式，変数
3	条件分岐
4	リスト，繰り返し
5	式の抽象化と関数
6	関数呼び出し，スコープ



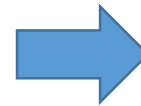
1 - 2. プログラミング

プログラミング (programming)



- コンピュータは、プログラムで動く
- プログラミングは、プログラムを設計、製作すること
- 何らかの作業を、コンピュータで実行させるために行う

```
a = [200, 400, 300]
for i in a:
    print (i * 1.08)
```



```
216.0
432.0
324.0
```

**プログラムの
ソースコード**

**プログラムの
実行結果**

ソースコード (source code)



- プログラムを, 何らかのプログラミング言語で書いたもの
- 「ソフトウェアの設計図」ということも.
- 人間も読み書き、編集できる
- 複数のプログラミング言語を使うことも

```
import picamera  
camera = picamera.PiCamera()  
camera.capture("1.jpg")  
exit()
```

Raspberry Pi で, カメラを使って
撮影し, 画像を保存するプログラムの
ソースコード

Python プログラムの実行手順例



- プログラムの起動は、シェル (Windows のコマンドプロンプトなど) から、コマンドで行える

```
x = 100
if (x > 20):
    print("big")
else:
    print("small")
s = 0
for i in [1, 2, 3, 4, 5]:
    s = s + i
print(s)
```

Python プログラムのソースコードを, foo.py のようなファイル名で保存しておく

ソースコード

```
kaneko@www:/tmp$ python foo.py
big
15
```

プログラミングで気を付けること



- コンピュータは「万能のマシン」と言われることもある
- プログラムで行わせたい「作業」について、深い理解が必要
- プログラム中の誤り（バグ）を、コンピュータが自動で発見してくれるわけではない。
- 「プログラムが期待通りに動いているか」を検証する、テストが必要

開発環境とは



- 開発環境とは、プログラミングにおけるさまざまなことを支援する機能をもったプログラム
 - プログラムの作成、編集（エディタ）
 - プログラム中の誤り（バグ）の発見やテストの支援（デバッガ）
 - プログラムの実行
 - マニュアルの表示
 - プログラムが扱うファイルのブラウズ
 - プログラムの配布（パッケージ機能など）、共有、共同編集
 - 公開、共有、共同編集
 - バックアップ、バージョン管理
 - ※ これらが簡単に行えるようになる

プログラム作成ができるウェブサービス (オンラインの開発環境)



python tutor.com

VISUALIZE CODE AND GET LIVE HELP

Learn Python, Java, C, C++, JavaScript, and Ruby

[Python Tutor](#) (created by [Philip Guo](#)) helps people overcome a fundamental barrier to learning programming: understanding what happens as the computer runs each line of code.

Write code in your web browser, see it visualized step by step, and get live help from volunteers.

Related services: [Java Tutor](#), [C Tutor](#), [C++ Tutor](#), [JavaScript Tutor](#), [Ruby Tutor](#)

Over five million people in more than 180 countries have used Python Tutor to visualize over 100 million pieces of code, often as a supplement to textbooks, lectures, and online tutorials.

[Visualize your code and get live help now](#)

PythonTutor

<http://pythontutor.com/>

paiza.io/ja/projects/new

Beta paiza.io 新規コード 一覧

Python3 Enter a title here

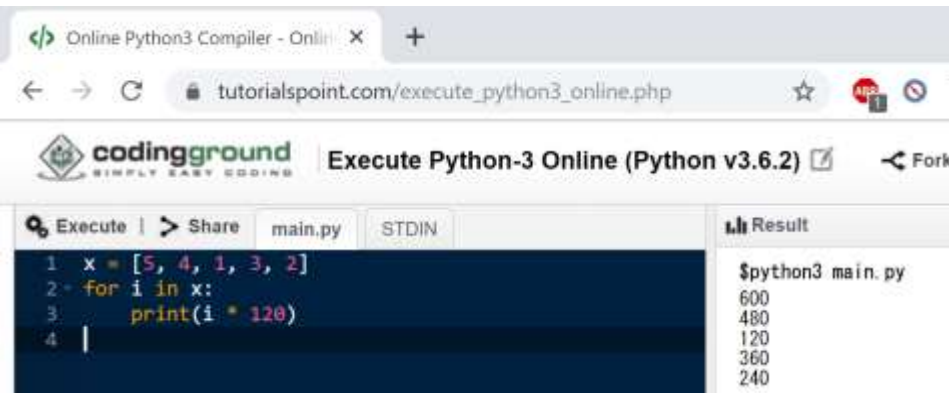
Main.py x +

```
1 # coding: utf-8
2 # Your code here!
3
4
```

Paiza.IO

<https://paiza.io/>

プログラム作成ができるウェブサービス (オンラインの開発環境) の例



CodingGround

<https://www.tutorialspoint.com/codingground.htm>



Google Colaboratory

<https://colab.research.google.com/>

ライブラリ類



ライブラリとは

- 複数のプログラムが共有して使えるような機能を持ったプログラムのこと.
- 多くの場合, プログラムの実行時にリンク (結合) される
- パッケージ (モジュール, インクルードファイルなどともいう)

※ パッケージの種類、豊富は、プログラミング言語とに違う



1 - 3. さまざまな プログラミング言語



プログラミング 言語は複数ある

- 複数のプログラミング言語を学ぶことは大事。
賛成できますか？
- 「1つを知っていれば、どの言語も大体似ているので、応用が利く」という考え方もある。
- 「やりたいこと、学びたいことに向いた言語を、そのときどきで選ぶのが、一番良い」とも。
- 人によって「好きな言語が違う」ということも

さまざまなプログラミング言語










- Python
- C
- Java
- JavaScript
- R
- Octave
- Scheme
- など

ここで行う作業

1. 20 より大きければ「big」、
さもなければ「small」と表示
2. $0 + 1 + 2 + 3 + 4 + 5$ を求める

なぜプログラミング言語は たくさんあるのでしょうか？



						
それぞれ 特徴があ る	Java	Python	C / C++	R	SQL	MATLAB / Octave
	どのコン ピュータ でも同じ プログラ ムが動く。	初心者向 け。その おかげで、 多数の拡 張機能も。	コン ピュータ の性能を 最大限引 き出す。	「データ 処理」に 特化した コマンド 言語	「データ ベース」 に特化し たコマン ド言語	「数値計 算」、 「信号処 理」など に特化し たコマン ド言語
	普及度は トップレ ベル。					

Python プログラム見本



```
x = 100
```

```
if (x > 20):
```

```
    print("big")
```

```
else:
```

```
    print("small")
```

```
s = 0
```

```
for i in [1, 2, 3, 4, 5]:
```

```
    s = s + i
```

```
print(s)
```

- すぐに実行できる
- さまざまな「パッケージ」で機能を拡張できる
- Windows でも Linux でも、ほぼ同じプログラムで動く

Java プログラム見本



```
public class Main {  
    public static void main(String[] args) throws Exception {  
        int x = 100;  
        if (x > 20) {  
            System.out.printf("big%n");  
        } else {  
            System.out.printf("small%n");  
        }  
        int s = 0;  
        for(int i = 1; i <= 5; i++) {  
            s = s + i;  
        }  
        System.out.printf("%d%n", s);  
    }  
}
```

- Windows でも Linux でも Android アプリでも, 同じプログラムで動く

C プログラム見本



```
#include <stdio.h>
int main(void){
    int x, s, i;
    x = 100;
    if (x > 20) {
        printf("big¥n");
    } else {
        printf("small¥n");
    }
    s = 0;
    for(i = 1; i <= 5; i++) {
        s = s + i;
    }
    printf("%d¥n", s);
    return;
}
```

- コンピュータの決め細かなコントロール
- 高速実行できるチューニング

JavaScript プログラム見本



Webアプリに向く

```
process.stdin.resume();
process.stdin.setEncoding('utf8');
var util = require('util');
var x = 100;
if (x > 20) {
    process.stdout.write('big¥n');
} else {
    process.stdout.write('small¥n')
}
var s = 0;
for(var i = 1; i <= 5; i++) {
    s = s + i;
}
process.stdout.write(util.format('%d¥n', s));
```

R プログラム見本



データ専門家向け

```
x <- 100
if (x > 20) {
  print("big")
} else {
  print("small")
}
s <- 0
for (i in c(1,2,3,4,5)) {
  s <- s + i
}
print(s)
```


Octave プログラム見本



```
x = 100
```

```
if (x > 20)
```

```
    printf("big¥n")
```

```
else
```

```
    printf("small¥n")
```

```
endif
```

```
s = 0
```

```
for i = [1 2 3 4 5]
```

```
    s = s + i
```

```
endfor
```

```
printf("%d", s)
```

行列計算, 信号処理など
に向く

Scheme プログラム見本



関数型言語

```
(define (decide x)
```

```
  (cond
```

```
    ((> x 20) "big")
```

```
    (else "small")))
```

```
(define (sum n)
```

```
  (cond
```

```
    ((= n 0) 0)
```

```
    (else (+ (sum (- n 1)) n))))
```

```
(begin
```

```
  (print (decide 100))
```

```
  (print (sum 5)))
```

全体まとめ



- プログラミングは、プログラムを設計、製作すること
- プログラム開発環境とは、プログラミングにおけるさまざまなことを支援する機能をもったプログラム
- プログラミング言語は多数ある