

プログラミング (Python)

第1回 introduction

早稲田大学本庄高等学院 2020年度版

飯島 涼



この授業について

選択科目 金曜 3・4 限

4月中はオンラインで動画を配信.

動画は見たい人だけ見てください.

教科書

- 大重美幸 詳細！Python 3 入門ノート ソーテック社 2017
- Bill Lubanovic 入門 Python 3 オライリージャパン, 2015

参考書

- 渡辺有隆[著], Ozy・秋葉拓哉[協力], プログラミングコンテスト攻略のためのアルゴリズムとデータ構造, マイナビ, 2015
- Bill Lubanovic[著], 齊藤康毅[監訳], 長尾高広[訳], 入門 Python3, O'REILLY Japan, 2015
- 大重美幸[著], 詳細! Python3 入門ノート, ソーテック社, 2017
- ロベール[著], ロベールのC++入門講座, マイナビ, 2007
- Anany Levitin, Maria Levitin[著], Algorithmic Problem Solving with Puzzles and Games, O'REILLY, 2014

⇒自分が何か1冊でもプログラミングの本を持っている場合購入の必要はありません。資料は配布します。

プログラミングとは

プログラミング

- プログラム
 - コンピュータへの命令のこと
 - コンピュータが処理できる言語で書く。
- プログラミング
 - プログラム（コンピュータへの命令）を書く行為

プログラミングでできること (概要)

- ソフトウェア
 - アプリ・ゲーム作成
 - 普段スマホやゲーム機でやっているゲームはプログラムされたもの
 - シミュレーション
 - 現実世界では負担の大きい実験をコンピュータ上で行う。
 - 計算不可能な数式、手書きで書くのが難しいグラフを描画する
- ハードウェア
 - 電子工作
 - 回路を完成させた後、回路を思い通りに動作させるためにプログラムが必要

etc……

プログラミング言語の種類

- コンピュータへの命令は専用のプログラミング言語を使う
 - C/C++
基本的 / 主にOS・ハードウェアの設計等 / 少し前までは入門といえばC
 - Java
Android アプリ / オブジェクト指向の入門
 - Python
インタプリタ / 統計・機械学習 / webフレームワーク / Cよりもわかりやすいと話題に
 - Ruby
Webフレームワーク
 - Verilog
ハードウェア・回路
 - BASIC / アセンブラ等
後述
 - Haskell
関数型プログラミング言語

プログラミング言語の種類

- コンピュータへの命令は専用のプログラミング言語を使う
 - C/C++
基本的 / 主にOS・ハードウェアの設計等 / 少し前までは入門といえばC
 - Java
Android アプリ / オブジェクト指向の入門
 - Python
インタプリタ / 統計・機械学習 / webフレームワーク / Cよりもわかりやすいと話題に
 - Ruby
Webフレームワーク
 - Verilog
ハードウェア・回路
 - BASIC / アセンブラ等
後述
 - Haskell
関数型プログラミング言語

Python

- メリット
 - 入門部分がわかりやすい
 - 型（後述）を気にせずプログラムがかける
 - 簡単に統計・機械学習ツールが試せる
- デメリット
 - ほかの言語よりも遅い傾向
 - OSやメモリがどのように扱われているのかまでは学習しづらい（気になったらC,C++に移行）

この授業の目的・大まかな流れ

- プログラミングに使われる特有の考え方・アルゴリズムを理解する（建前）
 - 基本的な文法（4-7月程度）
 - アルゴリズム・パズルを用いた演習等（8月-）
- プログラミングの楽しさを知ってもらう（本音）
 - CUI or GUI のゲームを作る（9月-）
 - 各種大会・コンテスト・イベントの案内/助言など（随時）
 - （応用、レベルにより調節）プログラムで高校数学・物理の内容を再現する

授業のルールなど

- 考え方や書き方を前で説明してもらうことがあります（ホワイトボードコーディング）
- 先生 or 生徒の説明は、わからなかった場合、途中で中断させ、質問することができます
⇒「わからないことを把握している」ことが非常に重要であり、この授業においては「わからないことを躊躇せず表明できる人」が最も優遇されます。
- テストをする場合、相談できるツールを除くすべての資料を閲覧することができます。覚えるよりも理解することを目指してください。

プログラミング環境の準備

- Online上で試す場合
 1. Google のアカウントを作成
 2. “Google Colaboratory”で検索
 3. ファイル > 新しいPython 3 ノートブック を選択
- 自分のPC上で試したい場合
 - Anaconda インストールで検索
- USBにインストールしてPython環境を持ち歩きたい場合
 - WinPythonをUSB上にインストール

授業で説明に使う環境

- Online上で試す場合
 1. Google のアカウントを作成
 2. “Google colaboratory”で検索
 3. ファイル > 新しいPython 3 ノートブック を選択
- 自分のPC上で試したい場合
 - Anaconda インストールで検索
- USBにインストールしてPython環境を持ち歩きたい場合
 - WinPythonをUSB上にインストール

授業準備

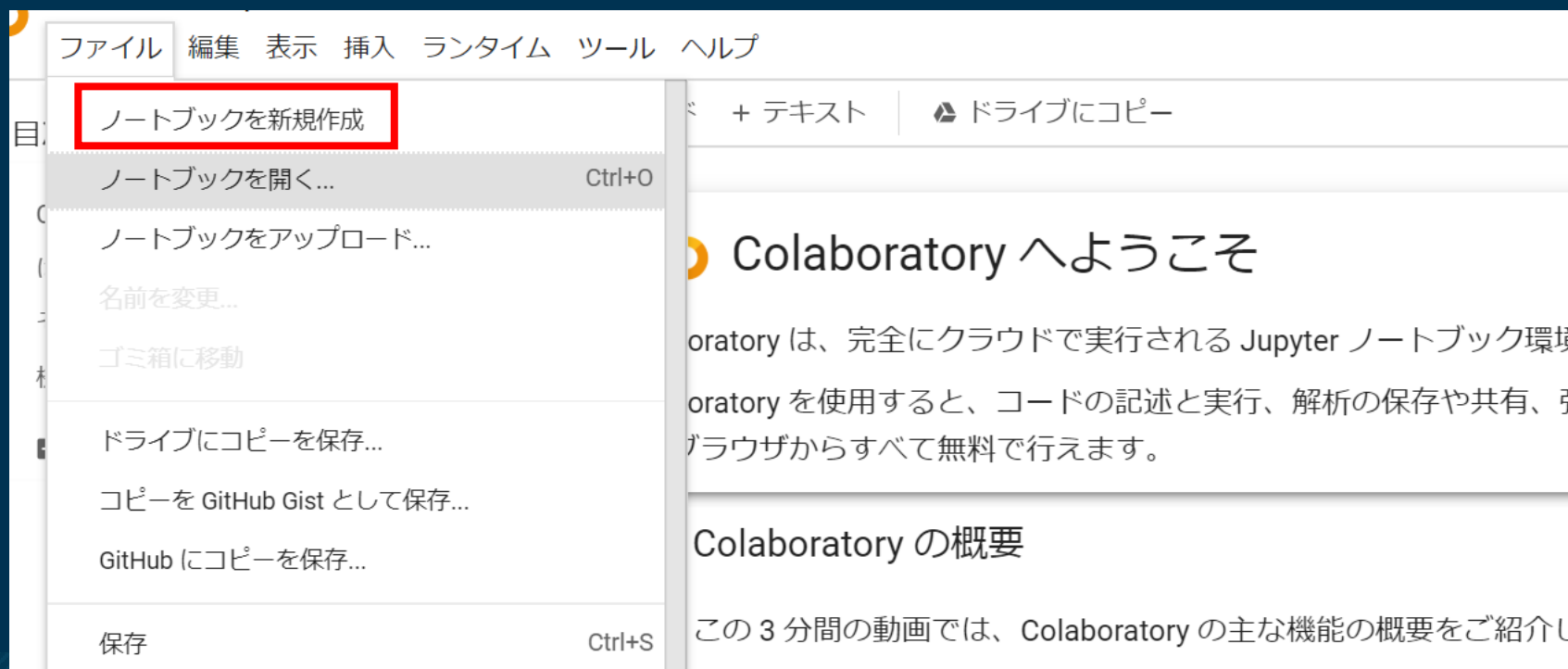
1. (アカウントない場合) Google のアカウントを作成し, Colaboratory を使えるようにする.

<https://accounts.google.com/signup/v2/webcreateaccount?continue=https%3A%2F%2Faccounts.google.com%2FManageAccount%3Fnc%3D1&hl=ja&flowName=GlifWebSignIn&flowEntry=SignUp>

2. AtCoder / Aizu Online Judge のアカウントを作成する.

Google Colaboratoryへアクセスして、持っているアカウントでログインする。

ファイル > ノートブックを新規作成 からプログラムのファイルを作成して開始



Google Colaboratory の利点

- スマートフォンでもコーディングが可能.
左上のメニューバー(三みたいなもの) => ファイル => ノートブックを新規作成
- スマートフォン, PCの両方を持っていない方は,
ノートなどにコードやフローチャートを書いて提出
を認めます.
- 格安で自宅にPCを購入したい場合は, Raspberry Pi 3 を購入してください. この科目のためにPCを購入する必要はありません.
 - 購入リスト
 - Raspberry Pi 3 (本体) 5000円
 - micro SDカード (8GB) 1000円, 秋葉原で買うと300円
 - マウス (家にあれば不要) 500円
 - キーボード (家にあれば不要) 1000円
 - ディスプレイ (テレビで代用可) 5000円
 - HDMI ケーブル (タイプAオス - タイプAオス) 500円
 - 計: 13300円 でプログラミング環境が整います (, サーバの運用の練習もできる) .

Aizu Online Judge



















































Aizu Online Judge : 会津大学が作った問題のリスト
初心者向けの練習問題がたくさんある

Problem Set from ITP1

Course Challenge Volume

44 problems

< prev | 1 / 5 | next > per-page : 10 20 30 50 100

		Problem Title	TL	ML	Success	
		ITP1_1_A: Hello World	1 sec	128 MB	49381  55.87 %	 x 18168 
		ITP1_1_B: X Cubic	1 sec	128 MB	37340  52.51 %	 x 14573 
		ITP1_1_C: Rectangle	1 sec	128 MB	34803  47.00 %	 x 12913 
		ITP1_1_D: Watch	1 sec	128 MB	23908  54.51 %	 x 10553 
		ITP1_2_A: Small, Large, or Equal	1 sec	128 MB	32071  40.33 %	 x 10592 
		ITP1_2_B: Range	1 sec	128 MB	21342  56.60 %	 x 10084 
		ITP1_2_C: Sorting Three Numbers	1 sec	128 MB	27115  42.46 %	 x 9353 
		ITP1_2_D: Circle in a Rectangle	1 sec	128 MB	21392  43.65 %	 x 7890 
		ITP1_3_A: Print Many Hello World	1 sec	128 MB	17975  64.62 %	 x 9148 
		ITP1_3_B: Print Test Cases	1 sec	128 MB	27396  36.95 %	 x 8162 

参考:

<http://judge.u-aizu.ac.jp/onlinejudge/finder.jsp?course=ITP1>



コンテスト



常設中のコンテスト

コンテスト名

- practice contest
- C++入門 AtCoder Programming Guide for beginners (APG4b)
- AtCoder Beginners Selection

予定されたコンテスト

開始時刻

コンテスト名

- | | |
|---------------|--------------------------------|
| 4/19(日) 21:00 | ● AtCoder Beginner Contest 163 |
| 4/30(木) 10:00 | ● 第6回 Asprova プログラミングコンテスト |

Job for Student 2020
Engineer / Data Specialist

膨大なプロダクト数と
8.8PBのデータ容量を持つ
リクルートの現場で開発/解析

時給2,000円

交通費・技術書購入・宿泊費(遠方者のみ)支給あり

エントリー
締切
05/20

インフォメーション

AtCoderは、オンラインで参加できるプログラミングコンテスト(競技プログラミング)のサイトです。リアルタイムのコンテストで競い合ったり、約3000問のコンテストの過去問にいつでも挑戦することが出来ます。

チュートリアル / 便利リンク集



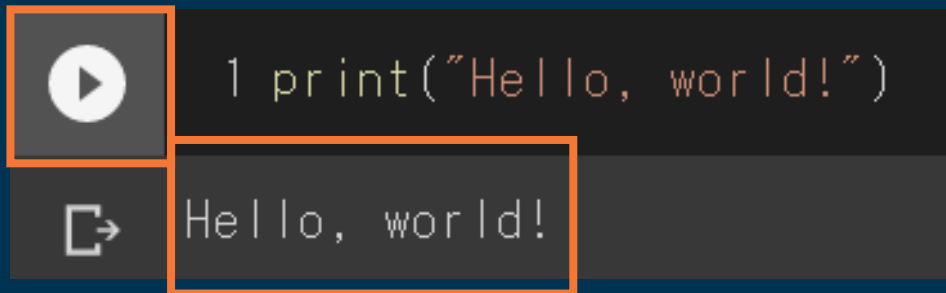
アルゴリズム実技検定過去問掲載中です。アルゴリズムの設計および実装能力を、資格で証明することが可能です。

プログラムを書く（実践）

□Hello world!

- 一番最初にかくプログラム
- プログラムで命令した文字を出力させる

実行ボタン



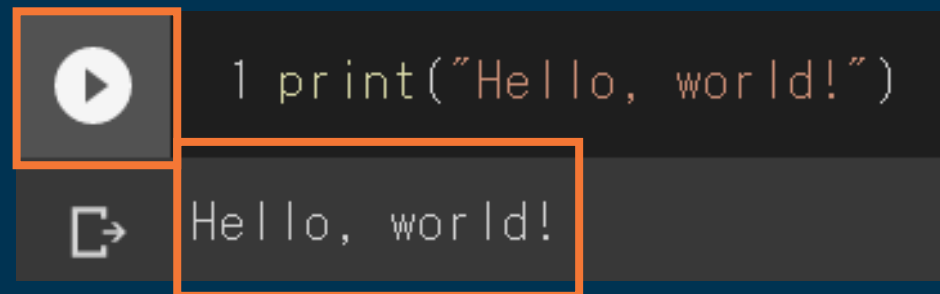
結果

解説

□print()

- 文字列を表示させる命令
- ()の中に表示させたい文字を書く
- 文字列は""(ダブルクォーテーション)で囲む。
- 基本的にプログラムはすべて半角英数字で書く。

実行ボタン



結果

変数

□変数

- 値を入れておける箱 のようなもの
- 変数の役割
 - 数字をおぼえておく
 - 計算の途中結果を保存しておく
 - 文字やデータを保存しておく

etc… …



代入

□代入

- 変数に具体的な値を入れること
- a に 5 を代入するという命令を

$a = 5$

と表記する。（数学では「等しい」の意味）

```
1 a = 5
2 print(a)
3
4 abc = 123
5 print(abc)
6
7 a123 = 56
8 print(a123)
9
10 # エラーになる変数名（先頭に数字がある）
11 #12a = 32
```

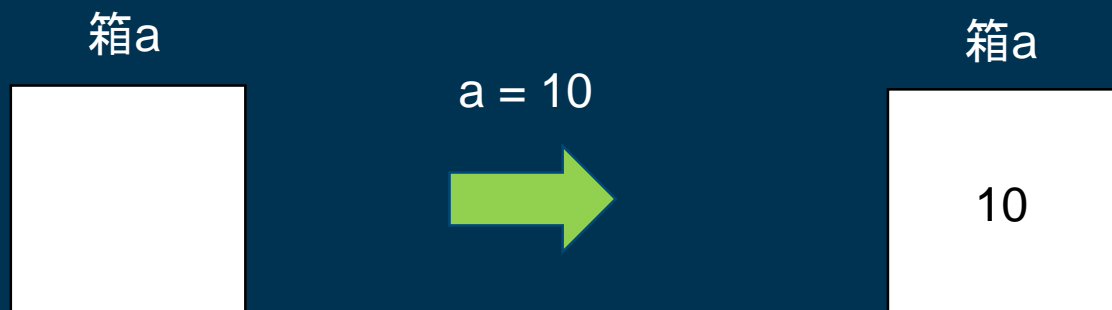
```
5
123
56
```

代入のイメージ

- 変数：何も入っていない箱
 - 箱の名前は左側、数字は右側

例). $a = 10$

- 代入前と代入後のイメージ



応用

- input()

- プログラムを実行した後、何か入力されるのを待って、入力された内容をsに代入します。
- プログラミングでは、**変数に文字列を入れることができます。**



```
1 s = input("Input your name:")  
2  
3 print("Hello,", s)
```



```
Input your name:asdf  
Hello, asdf
```

箱s

“入力した名前”

文字列の代入

文字列を代入する

- “ ” (ダブルクォート) 、 ’ ’ (シングルクォート) で囲む

変数の名前と、本当にprintで表現したい文字を区別するため

例)

```
s1 = "ryo"  
s2 = "iiijima"  
  
print(s1 + s2)
```

自由課題: 数学の代入とプログラミングの代入

- 予習したい方のための課題です。
- 「変数への代入」のプログラムを書かないまま、数学と同じような式を書いたらどうなるでしょうか？ その結果からわかることは何でしょうか？
 - 以下のプログラムはなぜエラーが出てしまったのでしょうか？

```
1 x
2 y = 2x
```

File "<ipython-input-2-603de413fd57>", line 2
y = 2x
^
SyntaxError: invalid syntax

SEARCH STACK OVERFLOW

```
1 x
2 y = 2*x
```

NameError Traceback (most recent call last)
<ipython-input-3-d3dd8f1e3df1> in <module>()
----> 1 x
 2 y = 2*x

NameError: name 'x' is not defined

エラーが出たら

- エラー文を読む
 - 英語を読んでください、どうしてもわからなければgoogle翻訳など使ってください。
- ググる
 - デバッグ
 - エラー
 - Qiitaなどの単語と合わせてエラー文を検索してください。