

「アルゴリズムとプログラム」

Python入門(Google Colaboratory版)

```
def total(a):  
    b = 0  
    for i in a:  
        b = b + i  
    return(b)
```

```
x = [0, 1, 2, 3, 4, 5, 6, 7]
```

```
s = total(x)  
print(s)
```

Google Colaboratory



1

どうしてPython(1)

言語の標準機能を拡張する各種のライブラリーが用意されています。特にAI分野で豊富にそろっています。

YouTubeやInstagramも、Pythonで開発されています。

| | |
|--------------|---------------------------|
| tensorflow | ディープラーニングのフレームワーク |
| keras | ディープラーニングのフレームワーク |
| Jupyter | Webブラウザで常時操作できるPythonカーネル |
| Flask | Webアプリのフレームワーク |
| Django | Webアプリのフレームワーク |
| requests | Web(HTTP)ライブラリー |
| scikit-learn | 機械学習ライブラリー |
| matplotlib | グラフライブラリー |

ただし、スマホアプリはJava/JavaScriptが主流で、Python用のライブラリーは今後期待されます。

2

どうしてPython(2)

○世界レベルのIEEEの2021年のプログラミング言語ランキングで1位です。

○特にAI分野やビッグデータの分析でPythonが使われるため、高収入プログラム言語になっています。

○海外の大学・中学・高校での学習用プログラミング言語のトップです。他のプログラムより読みやすく、動かしやすい。

国内では海外に比べてAI等が遅れているため、まだまだPythonのエンジニアが不足しています。

3

どうしてPython(3)

New

ブロック型ビジュアル言語

- ・習得しやすい
- ・見た目で理解しやすい
- ・すぐに楽しいものが作れる
- ・エラーが出にくい



プログラムの
基本や考え
方は同じ



テキスト言語

- ・やや難しい
- ・プログラムの動作を考える必要がある
- ・いろいろな物が作れる
- ・プログラムの入力でもエラーがでる

python



JavaScript

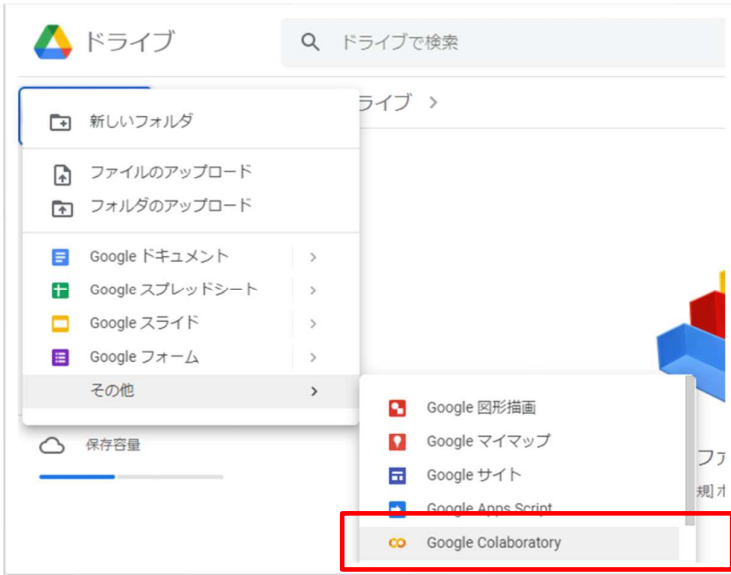


Java

似ているので、一つの言語が使えると他のものを覚えるのも楽

4

Colaboratory使ってみよう:起動



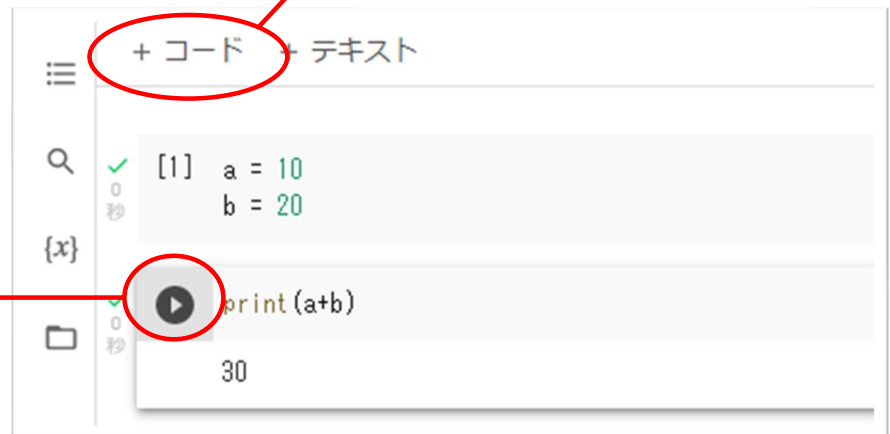
Googleドライブのファイルを保存するフォルダーで、
[新規]-[その他]-
[Google Colaboratory]
を指定

[Google Colaboratory]
が使えるようになる。



Pythonを使ってみよう

ファイルに新しいコード(プログラム)を追加



一つのコードを実行

学習で使用するColaboratoryというPythonの開発実行環境では、初めにGoogleドライブに作成した一つのファイルの中に複数のコードを入れることができます。一つ一つのコード(プログラム)を実行することができます。

学習の進め方(教材の内容)



| 資料 | 課題 No | 内容 | [読む/理解] | [打ち込み/実行] | [開発/実行] |
|--------------------|-------|--------------------------|---------|-----------|---------|
| Python 入門 | 課題 0 | どうして Python~Python の起動まで | | | |
| | 課題 1 | 1 行実行: 足し算の計算 | | | |
| | 課題 2 | 1 行実行: 変数と計算 | | | |
| | 課題 3 | 1 行実行とプログラムを区別しよう | | | |
| Python で アルゴリズム | 課題 4 | 打ち込 0: ドルから円に変換。 | | | |
| | 課題 5 | 開発 1: 入力した 2 個の数で四則演算 | | | |
| | 課題 6 | 打ち込み 1: 合格判断 | | | |
| | 課題 7 | 開発 2: 合格不合格判断 | | | |
| | 課題 8 | 開発 3: 成績 A~C | | | |
| | 課題 9 | 打ち込み 2: 1 から 10 までの数を言う | | | |
| | 課題 10 | 打ち込み 3: 1 から 10 までの合計を言う | | | |
| | 課題 11 | 開発 4: 2 から X までの偶数の合計 | | | |



指示書

- Python入門
- Pythonでアルゴリズム

この中の課題を自分でやっていきます。

チェックシート

課題をやるか確認します。課題ができたならチェックを書き込みます。

機能部品カード

プログラムを構成する Python の部品です。各課題にどの部品を使うか下記のように指示するので参照してください。



学習の進め方(チェックシート)

チェックシートが用意されています。チェックしながら学習してみましょう。

- **[読む/理解]** 資料の内容を理解します。
- **[打ち込み/実行]** 資料の内容をPCに入力して動作を確認します。
- **[開発/実行]** 資料をみて考えて、その課題プログラムを作成します。

| 資料 | 課題 No | 内容 | [読む/理解] | [打ち込み/実行] | [開発/実行] |
|--------------------|-------|--------------------------|--------------------------|--------------------------|--------------------------|
| Python 入門 | 課題 0 | どうして Python~Python の起動まで | | | |
| | 課題 1 | 1 行実行: 足し算の計算 | | | |
| | 課題 2 | 1 行実行: 変数と計算 | | | |
| | 課題 3 | 1 行実行とプログラムを区別しよう | | | |
| Python で アルゴリズム | 課題 4 | 打ち込 0: ドルから円に変換。 | <input type="checkbox"/> | | |
| | 課題 5 | 開発 1: 入力した 2 個の数で四則演算 | | | <input type="checkbox"/> |
| | 課題 6 | 打ち込み 1: 合格判断 | | <input type="checkbox"/> | |
| | 課題 7 | 開発 2: 合格不合格判断 | | | <input type="checkbox"/> |
| | 課題 8 | 開発 3: 成績 A~C | | | <input type="checkbox"/> |
| | 課題 9 | 打ち込み 2: 1 から 10 までの数を言う | | <input type="checkbox"/> | |
| | 課題 10 | 打ち込み 3: 1 から 10 までの合計を言う | | <input type="checkbox"/> | |
| | 課題 11 | 開発 4: 2 から X までの偶数の合計 | | | <input type="checkbox"/> |

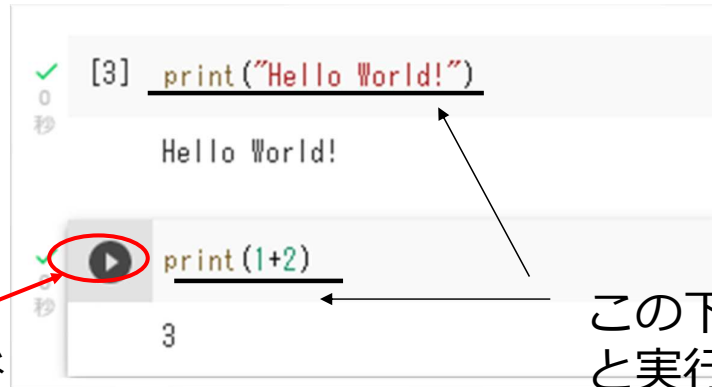
課題が終わったら、にチェックしてください。

課題1

1行実行: 表示と足し算の計算

一行ずつ命令を入力して、実行することができます。

初めてのPythonプログラム



新しいコード
(プログラム)を追加

一つのコード
を実行

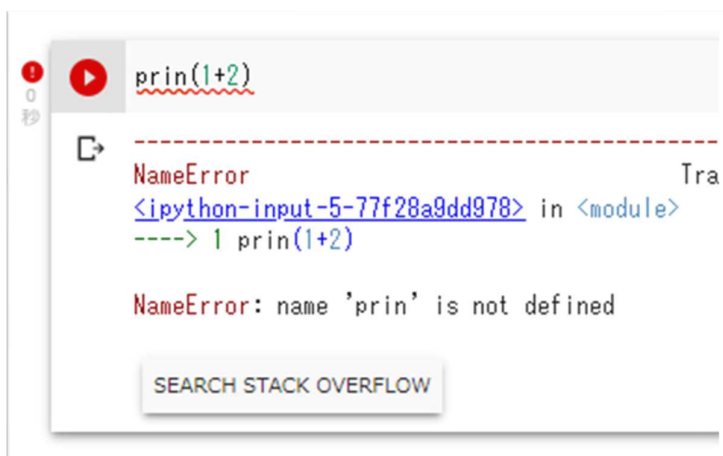
この下線を入力する
と実行結果
print() 表示する

部品
01

エラーが出た人はつぎのスライドを見て対応

エラーの対応(1)

プログラムを実行しているとき、エラーがあっても、次に正しく入力すれば問題ありません。



printの綴りが間違ってエラーになりますが、問題ありません。

正しく打ち直せばOKです。

課題2

1行実行: 変数と四則演算

変数



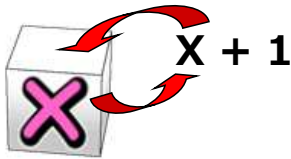
$x = 10$

Xと名前をつけた箱(変数)に1を入れる

```
x = 10
print(x + 20)
```

30

変数に入れた数を、計算で使うことができます。



$x = x + 1$

($x \leftarrow x + 1$ のイメージ)

初めにxの箱(変数)の中を取り出し+1する。計算結果をXの箱(変数)に入れなおす。

変数への代入は普通の数学の=とは違う意味なのでイメージを示してみました。



11

1行実行: 変数と四則演算(その1)

下の下線部の入力して、変数と四則演算の動作を確認しよう

```
[6] a = 6
    b = 2
    print(a+b)
    8

[7] print(a-b)
    4

[8] print(a*b)
    12

print(a/b)
    3.0
```

部品
01

部品
02

| | |
|-------|---|
| 足し算 + | + |
| 引き算 - | - |
| 掛け算 x | * |
| 割り算 ÷ | / |

エラーが出ても
正しく打ち直せばOKです。

12

1行実行: 変数と四則演算(その2)

プロンプトの後を入力して試してみよう

```
x = 1
print(x)
1

x = x + 1
print(x)
2
```

青色は実行結果

重要: 変数名の付け方

使える文字

- ・小文字英字
- ・大文字英字
- ・数字
- ・_(アンダーバー)

先頭に数字は使えない

○ abc ○ kakaku ○ total_a

× 1ban × take@jp × 日本語

13

2. 文字列の定義

文字列の扱いを確認しよう

確認・実行

```
print("Hello World!")
Hello World!

a = "Hello"
b = 'World!'
c = a + b
print(c)
HelloWorld!
```

- ・文字列は"(ダブルクォート)又は'(シングルクォート)'で囲んで定義します。
- ・必ず同じ" 又は'でくくります。
- ・文字列同志をくっつける時は + を使います。
- ・文字列としては日本語も使用できます。

14

課題3

打ち込み1:1行実行とプログラムを区別しよう

複数の命令をまとめて一つのプログラムにして実行してみよう。

実行(Run)

```
a = 6
b = 2
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```

8
4
12
3.0

プログラム
(命令の集まり)

プログラムが実行
され、結果が表示
される。

打ち込み1:1行実行とプログラムを区別しよう(その3)

実行してエラーが出る場合。
1行ずつエラーが出ると止まります。

重要:

Pythonは大文字・小文字を区別します

```
NameError                                Traceback
<ipython-input-13-e51ce7828a0c> in <module>
      1 a = 6
      2 b = 2
----> 3 prit(a+b)
      4 print(a-b)
      5 print(a*b)

NameError: name 'prit' is not defined
```

• Printの命令が、
pritと誤ってエラー
になった。

```
8
4
12
-----
NameError                                Traceback:
<ipython-input-15-eebd90d0c90a> in <module>
      4 print(a-b)
      5 print(a*b)
----> 6 print(y/b)

NameError: name 'y' is not defined
```

• 5行目まではエラーがなくて
正常に実行したが、
6行目に使っていない変数yが
あったので、エラーになった。

• 1個ずつエラーの箇所
を修正して、再度実行し
ていきます。

課題4

打ち込み2: ドルから円に変換。

1 \$ = 130円 とする



\$?

10

1300

プログラムを実行して、\$?が出た後、
キーボードから数を入力すると、
それに130を掛けた数を表示する
プログラムを打ち込みます。

17

打ち込み2: ドルから円に変換。(その1)

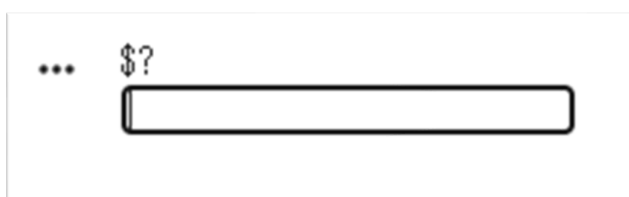
```
[16] print("$?")  
      a = int(input())  
      print(a*130)
```

```
$?  
10  
1300
```

部品
01

部品
02

新しくプログラムを打ち込んで実行してみよう。



部品
03

補足: 情報を入力する時は上図のような画面になります。

18

これで「Python入門」の学習は終わりです。
続いて「Pythonでアルゴリズム」で
学習を続けてください。