

「アルゴリズムとプログラム」

Python入門

```
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)
```



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	グラフィックライブラリー

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



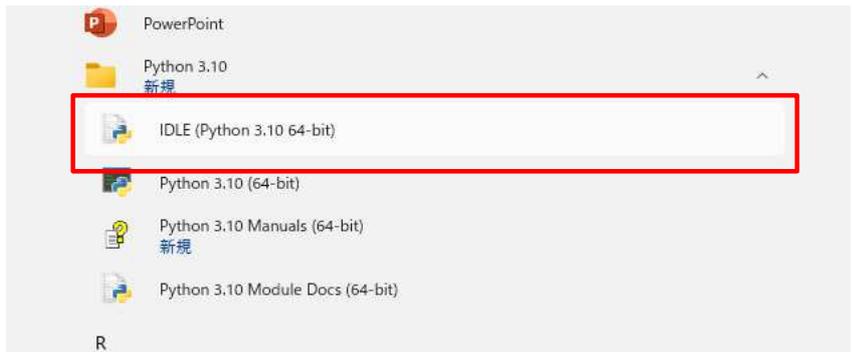
Java

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

IDLE使ってみよう:起動

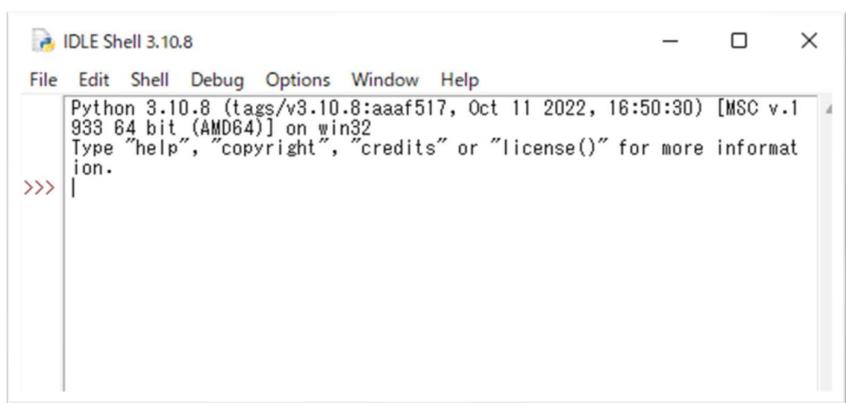
Windowsの例



IDLEを起動

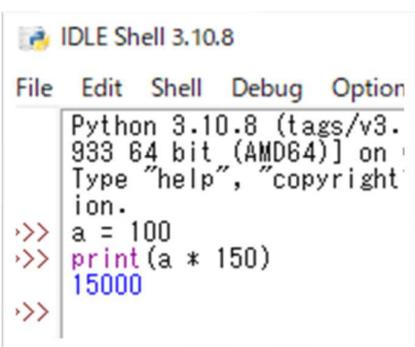


Pythonシェルの画面が
起動する
(シェルウィンドウ)

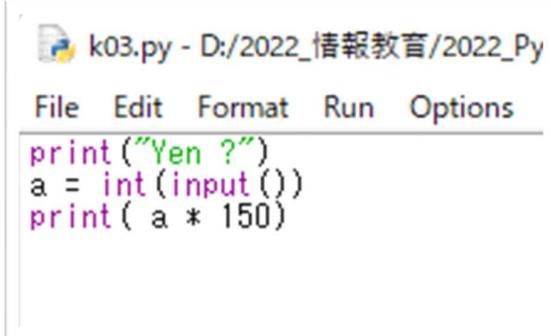


Pythonを使ってみよう(後で詳しく説明します)

方法1: Pythonシェルを使って、1行ずつ入力して実行させる。



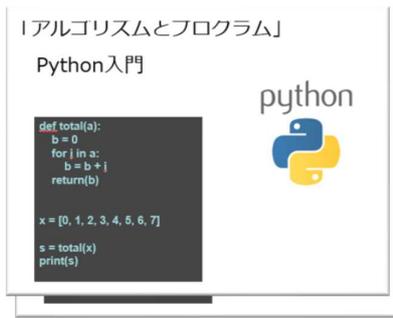
方法2: Pythonのプログラムファイルを一括して実行させる。



IDLE(Integrated Development Environment)
Pythonの統合開発環境
方法1も方法2も簡単に使えます

プログラム言語単体ではなく、各言語には統合開発環境が整備されていて、普通はそれを使って開発します。

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。			
	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			

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

シェルウィンドウでは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

プロンプト、ここに1行のプログラムを入れる。

```
>>>
>>> print("Hello World!")
Hello World!
>>> print(1 + 2)
3
>>>
```

部品
01

この下線を入力すると青字の部分が実行結果
print() 表示する

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)

シェルウィンドウでプログラムを1行ずつ実行しているときは電卓みたいなモードです、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。

```
>>> prin(1 + 2)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    prin(1 + 2)
NameError: name 'prin' is not defined. Did you mean: 'print'?
>>> print(1 + 2)
3
>>>
```

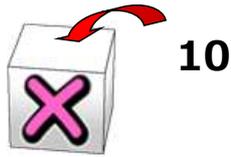
printの綴りが間違っエラーになりますが、問題ありません。

正しく打ち直せばOKです。

課題2

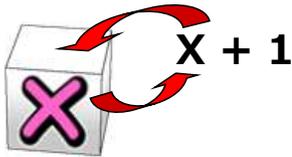
1行実行: 変数と四則演算

変数



$x = 10$

Xと名前をつけた箱(変数)に1を入れる



$x = x + 1$

($x \leftarrow x + 1$ 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。

```
>>>
>>> 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。



11

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。

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

```
>>> a = 6
>>> b = 2
>>> print(a + b)
8
>>> print(a - b)
4
>>> 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
>>>
```

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!
>>> print("こんにちは")
こんにちは
>>>
```

- ・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

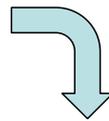
課題3

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ファイルにして実行してみよう。

```
*k03.py -
File Edit Format Run Options V
a = 6
b = 2
print(a + b)
print(a - b)
print(a * b)
print(a / b)
```

実行(Run)



プログラムが実行され、結果が表示される。

プログラムファイル
(命令の集まり)

```
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
>>>
===== RESTART: D:/2022_情報教育/2022_Pyth
on教材/ProgPy/k03.py =====
8
4
12
3.0
>>>
```

打ち込み1:1行実行とプログラムを区別しよう(その1)

プログラムを入力して、プログラムファイルを作成してみよう

```
IDLE Shell 3.10.8
File Edit Shell Debug Options W
New File Ctrl+N
Open... Ctrl+O
Open Module... Alt+M
Recent Files
Module Browser Alt+C
Path Browser
Save Ctrl+S
Save As... Ctrl+Shift+S
Save Copy As... Alt+Shift+S
Print Window Ctrl+P
Close Window Alt+F4
Exit IDLE Ctrl+Q
```

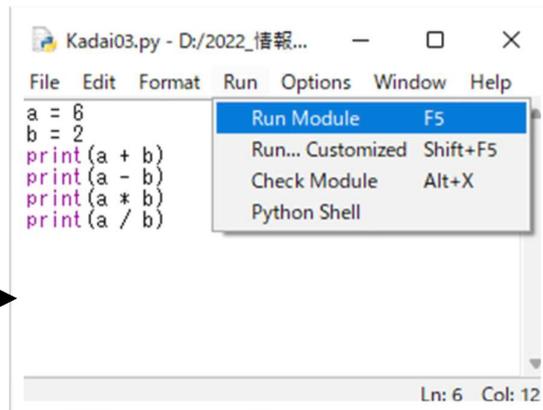
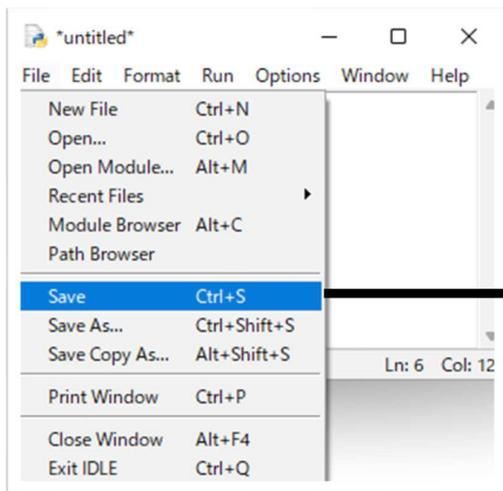
```
*untitled*
File Edit Format Run Options Window Help
a = 6
b = 2
print(a + b)
print(a - b)
print(a * b)
print(a / b)|
Ln: 6 Col: 12
```

コードウィンドウが開いたら、上のプログラムを入力していく

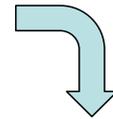
シェルウィンドウの
[File]-[New File]を指定する。
→ コードウィンドウが新しく開く

打ち込み1:1行実行とプログラムを区別しよう(その2)

プログラムの保存をして、実行する。



[Run]-
[Run Module]
で実行する。



コードウィンドの
[File]-[Save]でプログラムを
保存する。

ファイル名では、このプログ
ラムではKadai03としていま
す。



17

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。

```
=====  
=====  
Traceback (most recent call last):  
  File "D:/2022_情報教育/2022_Python教材/ProgPy/Kadai03.py", line 3,  
    in <module>  
      prit(a + b)  
NameError: name 'prit' is not defined. Did you mean: 'print'?
```

- Printの命令が、pritと誤ってエラーになった。

```
=====  
=====  
8  
4  
12  
Traceback (most recent call last):  
  File "D:/2022_情報教育/2022_Python教材/ProgPy/Kadai03.py", line 6,  
    in <module>  
      print(x / b)  
NameError: name 'x' 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のx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。

重要:
Pythonは大文字・小文字を区別します

18

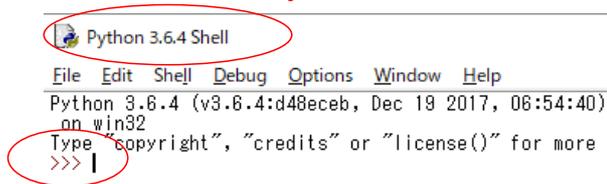
シェルウィンドとコードウィンドの見分け方

シェルウィンドとコードウィンドは見た目には見えますが、違い違う使い方をします。しっかり区別しましょう。

シェルウィンドウ

- ・一行ずつ入力して実行します。
- ・プログラムの実行結果(入力・出力)を出します。
- ・プログラムのエラーを表示します

ウィンドウの名前は
Python xxxx Shell

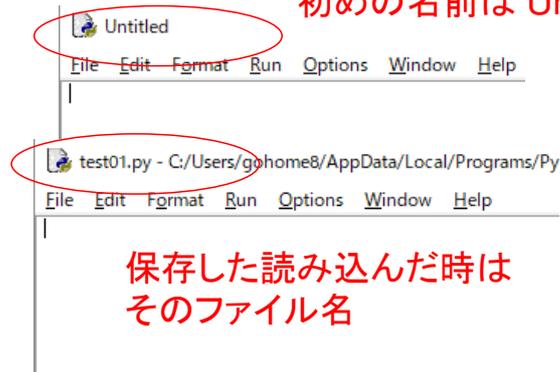


>>> のプロンプトが表示

コードウィンドウ

- ・まとまったプログラムの作成と編集をします。
 - ・プログラムの保存と読み込みをします。
 - ・プログラムの実行を指示します。
- (実行結果はシェルウィンドウ)

初めの名前は Untitled



保存した読み込んだ時は
そのファイル名

課題4

打ち込み2: ドルから円に変換。

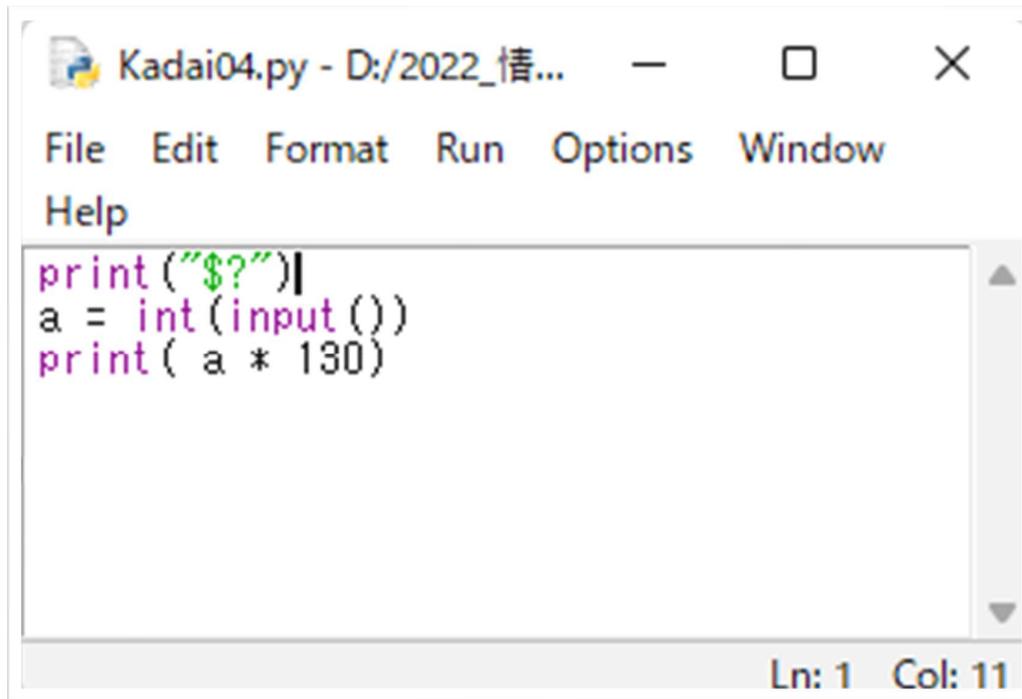
1 \$ = 130円 とする



\$?
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。

打ち込み2: ドルから円に変換。(その1)



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

Ln: 1 Col: 11

部品
01

部品
02

部品
03

新しくプログラムを打ち込んで実行してみよう。
(プログラムはkadai04で保存しています。)

21

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。

22