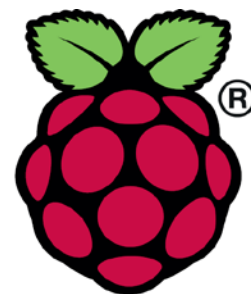


# 抜粋版

## はじめてのAIプログラム学習キット4

グーグル開発のTensorflow、Pythonの基礎をマスター、体験

## Python学習編(Pi4B版)



**Raspberry Pi**

スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

[sales@spectrum-tech.co.jp](mailto:sales@spectrum-tech.co.jp)

# python学習編 目次

python入門

<http://tohoho-web.com/python/index.html>

python-izm

<http://www.python-izm.com/>

python公式HP

<https://docs.python.jp/2.7/tutorial/index.html>

python自習テキスト

[http://giraffe.topaz.ne.jp/wiki/doku.php/py:python\\_curriculum](http://giraffe.topaz.ne.jp/wiki/doku.php/py:python_curriculum)

	ページ
① 概要	
• pythonとは	<a href="#">3</a>
• pythonの設定	<a href="#">4</a>
• Python2,3 の違い	<a href="#">5</a>
• 仮想環境の利用	<a href="#">6</a>
• プログラムの実行	<a href="#">8</a>
• 対話モード	<a href="#">9</a>
• プログラムの構造	<a href="#">10</a>
• プログラム作成ツール	<a href="#">12</a>
② 基本	
• 数値・文字列・型	<a href="#">13</a>
• リスト、タプル、辞書	<a href="#">14</a>
• 演算子	<a href="#">17</a>
• 制御文	<a href="#">18</a>
• 関数	<a href="#">19</a>
• クラス	<a href="#">20</a>
• モジュール、パッケージ	<a href="#">21</a>
③ 便利なモジュール	
• 日付・時刻	<a href="#">23</a>
• 画面作成	<a href="#">24</a>
• グラフ	<a href="#">25</a>
• 数値計算	<a href="#">26</a>
④ 例	
• ヒストグラム	<a href="#">27</a>
• ボタン表示	<a href="#">28</a>
⑤ 練習問題	<a href="#">29</a>
⑥ jupyter notebook	<a href="#">30</a>

抜粋版のためページと内容が異なります。

## python基礎編

### ①. 概要

- Pythonとは

- Pythonはさまざまな分野のアプリケーションで使われている、極めてパワフルな動的プログラミング言語です。Pythonはよく、Tcl、Perl、Scheme、Javaなどと比較されます。Pythonには次のような特徴があります:
  - とてもクリーンで読みやすい文法
  - 強力な内省(イントロスペクション)機能
  - 直感的なオブジェクト指向
  - 手続き型のコードによる、自然な表現
  - パッケージの階層化もサポートした、完全なモジュール化サポート
  - 例外ベースのエラーハンドリング
  - 高レベルな動的データ型
  - 事実上すべてのタスクをこなせる、広範囲に及ぶ標準ライブラリとサードパーティのモジュール
  - 拡張とモジュールはC/C++で書くのが容易(JythonではJava、IronPythonでは.NET言語を利用)
  - アプリケーションに組み込んでスクリプトインタフェースとして利用することが可能

出典: <http://www.python.jp/about/>より

## python基礎編

### ①. 概要

- Pythonの設定

- Python2,3 を切り替えて使用可能

- Pyenv を使用してバージョンを確認、\*のところが使用中のバージョン

```
pi@raspberrypi:~ $ pyenv versions
* system (set by /home/pi/.pyenv/version)
  2.7.16
  3.5.4
  3.6.9
  3.7.3
```

- バージョン切替は pyenv global 3.6.9

```
pi@raspberrypi:~ $ pyenv global 3.6.9
pi@raspberrypi:~ $ pyenv versions
  system (set by /home/pi/.pyenv/version)
  2.7.16
  3.5.4
* 3.6.9
  3.7.3
```

デスクトップ上の  
「よく使うコマンド集.txt」  
でコピーして使ってください

(注) Rootでは、pyenv  
は動作しません

Python学習編では、基本  
は、system python3,又は  
3.6.9で利用ください。  
Python2は、2020年1月  
からサポート停止

## python基礎編

## ①. 概要

詳細は、こちらをご覧ください

<https://postd.cc/the-key-differences-between-python-2-7-x-and-python-3-x-with-examples/>

## • Python2, 3の違い

python2は2020年1月に  
サポート終了になりました。

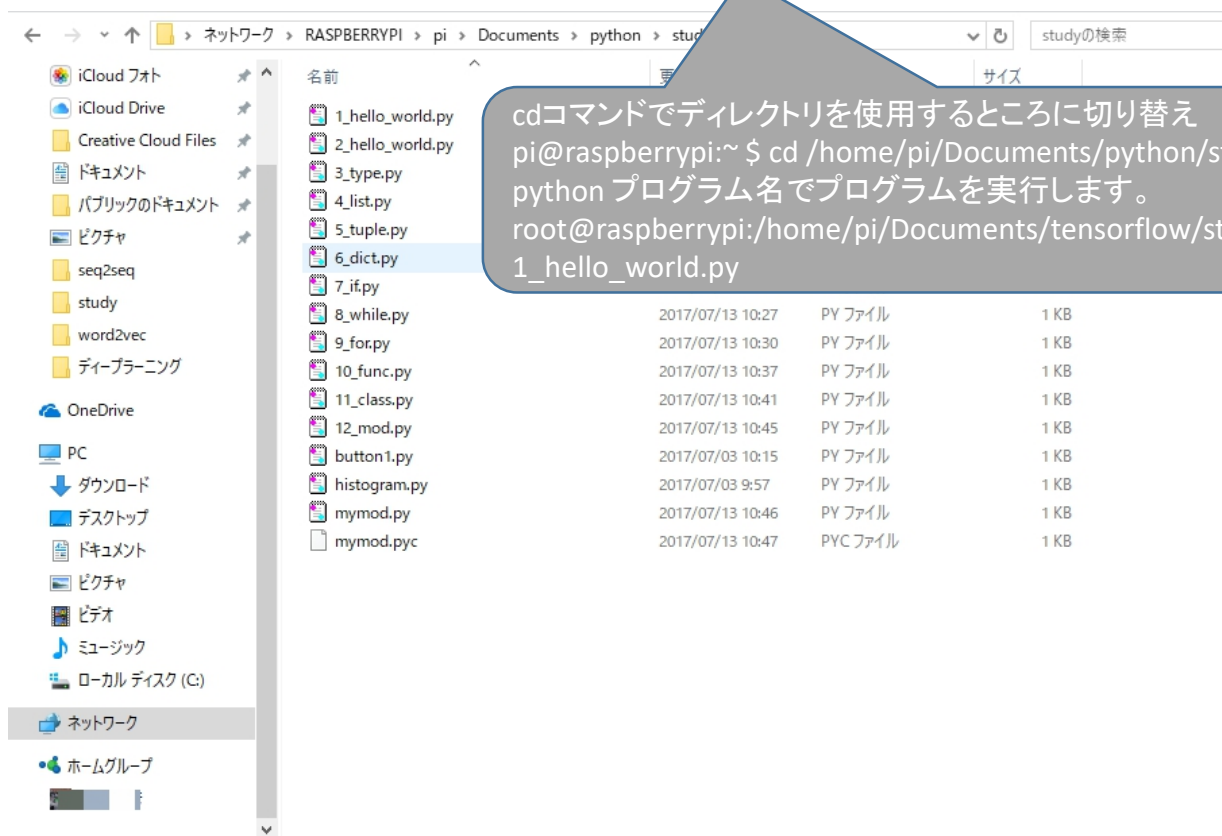
	Python2	Python3	コメント
Print文	<code>print 'Hello, World!'</code>	<code>print('Hello, World!')</code>	3は()が必ず必要になります
xrange	<code>range, xrange</code>	<code>rangeのみ</code>	
unicode	<code>u"unicode"</code>	エラー テキストはstr データはbytes	
整数除算	<code>3/2=1</code> <code>3/2.0=1.5</code>	<code>3/2=1.5</code> <code>3/2.0=1.5</code>	分母の数字をfloat, intをきちんと定義する必要がある
計算速度	早い	遅い	

本資料に記載しているプログラムはpython3で動作します

## python基礎編

## ①. 概要

- Python学習用ディレクトリ 全て体験できます。
- /home/pi/Documents/python/study



cdコマンドでディレクトリを使用するところに切り替え  
pi@raspberrypi:~\$ cd /home/pi/Documents/python/study  
python プログラム名でプログラムを実行します。  
root@raspberrypi:/home/pi/Documents/tensorflow/study# python  
1\_hello\_world.py

## python基礎編

### ①. 概要

#### • プログラムの実行

- Pythonの入ったディレクトリに移動して、pythonコマンドで実行します。
- hello\_world.pyのプログラムの中身

```
print "Hello world!"
```

以下、プログラムの場合の様式

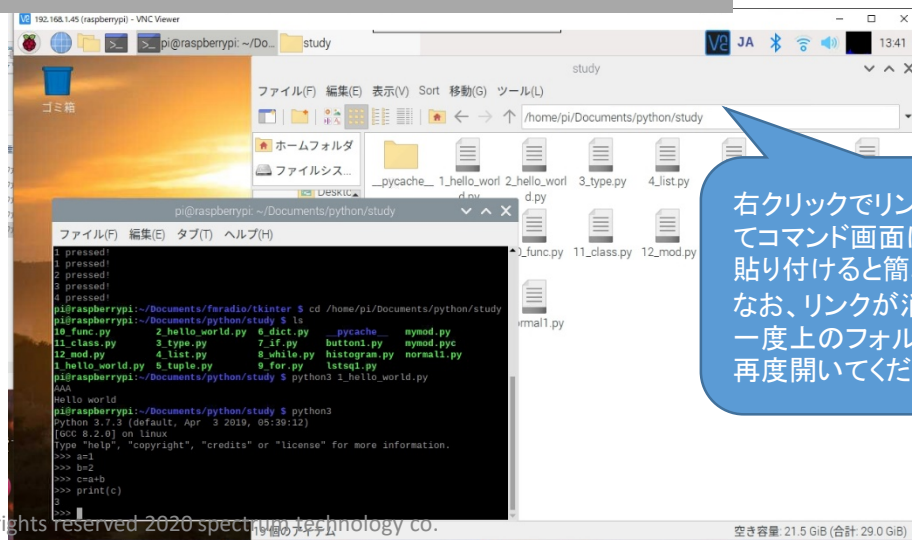
#### • コマンド

```
pi@raspberrypi:~ $ cd /home/pi/Documents/python/study
pi@raspberrypi:/home/pi/Documents/python/study$ python3 1_hello_world.py
Hello world!
```

- pythonプログラムに入ったディレクトリに移動
- プログラムの実行
- Hello worldの表示

以下、コマンドプロンプトの場合の様式

raspberry pi画面



## python基礎編

### ①. 概要

- 対話モード

- Pythonのひとつの特徴でもあり、プログラムを作成せず、直接コマンドを入力して、その結果を直ぐに得ることができます。
- 簡単なテストをする場合に有効です。
- python3と入力すると対話モードに入り、CTL+dで抜け出します

- コマンド

```
pi@raspberrypi:~ $ python3
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a=1
>>> b=2
>>> c=a+b
>>> print(c)
3
```

- python3と入力(systemのpython3で実施してください、他の仮想環境ではtkinterが動作しない場合があります)
- >>>のプロンプトになると対話型



## python基礎編

## ①. 概要

## • プログラムの構造

## • 関数定義

```
def main()      #main関数を定義します。
```

```
    print("hello world")
```

## • プログラムの実行文

```
if __name__ == "__main__":
```

```
    main()
```

アンダーバー2個になります

#\_\_name\_\_は現在のモジュール名、\_\_main\_\_が設定、別のモジュールから呼び出された時は実行しない。  
#main()プログラムが実行されます。

2\_hello\_world.py

```
# coding: utf-8
```

```
def main():
```

```
    print("hello world")
```

```
if __name__ == "__main__":
```

```
    main()
```

大きなプログラムも関数の定義部分が大きくなるだけです

## python基礎編

## ②. 基本

## • 数値、文字列、型

- c++と違って、型を宣言する必要がありません。使う前にintの場合は小数点なし、floatの場合は、小数点、文字列は、シングルのクォーテーションで宣言します。
- 使うときに数字をいれると間違いやすいので定数の初期化で型を宣言します。

```
# define  
x=0      int  
y=0.0    float  
z=""     string
```

## • Print文での型宣言

```
3_type.py
```

```
print ("%d" % a)  
print ("%2f" % b)  
print ("%s" % c)
```

123  
1.02  
hello

d:digitで整数  
f:floatで前の2が小数点2桁  
s:stringで文字

## python基礎編

### ②. 基本

- 演算子

- 代数演算子

```
+a # 正数  
-a # 負数  
a + b # 加算  
a - b # 減算  
a * b # 乗算  
a / b # 除算  
a % b # a を b で割った余り  
a ** b # a の b 乗
```

- 比較演算子

```
a == b # a が b と等しい if文でよく使われる  
a != b # a が b と異なる  
a < b # a が b よりも小さい  
a > b # a が b よりも大きい  
a <= b # a が b 以下である  
a >= b # a が b 以上である
```

## python基礎編

### ②. 基本

- 制御文

- if, elif, else      もし～ならば

7\_if.py

```
if a > 10:
    print ("high")
elif a == 10:
    print ("fair")
else: print ("small")
```

3分岐の条件文

- while, else      ～あいだ

8\_while.py

```
n = 0
while n < 5:
    print (n)
    n += 1
else: print ('END')
```

nが5までの間、処理を繰り返す。

0,1,2,3,4  
END

- for in    ～のあいだ

9\_for.py

```
for n in [1, 2, 3]:
    print (n)
```

リスト内のデータを全て:n  
回処理を繰り返します

1, 2, 3

## python基礎編

## ②. 基本

## • クラス

- classを使って定義し、インスタンス化(実体化)します。クラス名の先頭は大文字
- オブジェクト指向のプログラミングの肝になります。ひとつのオブジェクトで複数のデータを持つことができます。

11\_class.py

```
class Dog:
    kind = 'canine'
    def __init__(self, name):
        self.name = name

>>> d = Dog('Fido')
>>> e = Dog('Buddy')
>>> d.kind
>>> e.kind
>>> d.name
>>> e.name
```

#Dog:クラス名  
# クラス変数

#インスタンス変数  
#クラスを使ってnameに値を代入し、dに確保

#dのクラス変数を表示 'canine'  
#eのクラス変数を表示 'canine'  
#dのインスタンス変数を表示 'Fido'  
#eのインスタンス変数を表示 'Buddy'

## python基礎編

### ③. 便利なモジュール

- 日付・時刻
  - 最も使われるモジュールです。

```
>>> from datetime import date
>>> now = date.today()
>>> now
datetime.date(2020, 2, 14)
>>> now.strftime("%m-%d-%y. %d %b %Y is a %A on the %d day
of %B.")
'12-02-03. 02 Dec 2003 is a Tuesday on the 02 day of December.'

>>> birthday = date(1964, 7, 31)
>>> age = now - birthday
>>> age.days
14368
```

datetimeモジュールのdate関数  
をインポートして使用  
<https://docs.python.jp/2.7/library/datetime.html>

## python基礎編

## ③. 便利なモジュール

## • 画面作成: tkinter

- GUI画面を作成するのに使います。ラベル、ボタン、リストボックス、メニュースクロールバーなどがあり、簡単に作成できるので便利です。
- pythonで一番時間がかかるのは、入出力などのプログラムを作成するところです。根気強く実施しましょう。

```
import tkinter as tk
```

```
root = tk.Tk()  
button = tk.Button(root, text = 'Python/Tkinter')  
button.pack()  
root.mainloop()
```

ボタンのwidgetです。

```
# coding: utf-8  
import tkinter as tk  
root = tk.Tk()  
str = 'Hello, world, こんにちは世界'  
tk.Label(root, text=str, font=(u'MS ゴシック', 12)).pack()  
tk.Label(root, text=str, font=(u'MS 明朝', 12, 'italic')).pack()  
tk.Label(root, text=str, font=(u'MS ゴシック', 16, 'underline')).pack()  
root.mainloop()
```

ラベルのwidgetです。

## python基礎編

### ③. 便利なモジュール

- グラフ: matplotlib

- 非常に便利なツールです。内容によって表示に時間がかかる場合もあります。

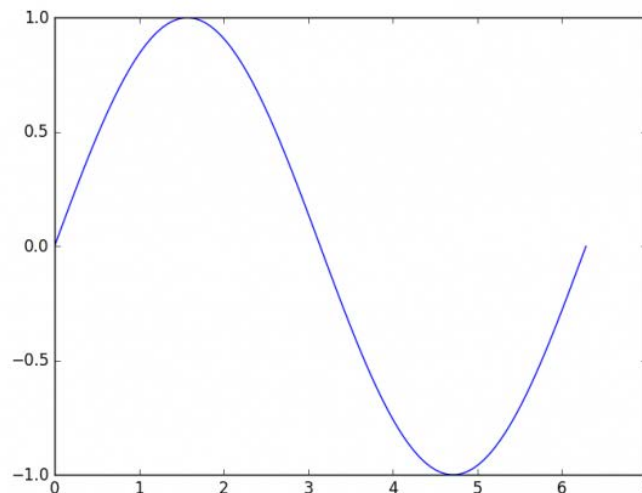
```
import numpy as np
from matplotlib import pyplot
```

```
pi = 3.1415
x = np.linspace(0, 2*pi, 100)
y = np.sin(x)
```

```
pyplot.plot(x, y)
pyplot.show()
```

いろいろなグラフが書けます。

<https://matplotlib.org/>

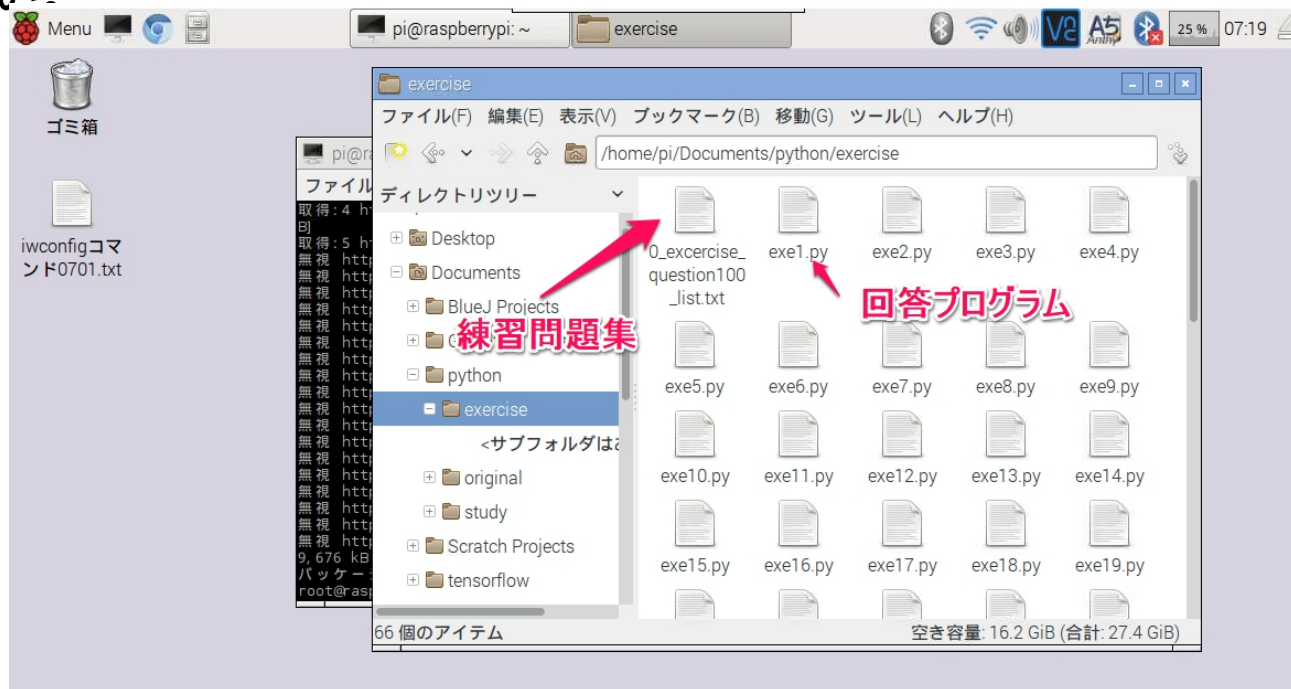




## python基礎編

## ⑤. 練習問題

- home/pi/Documents/python/exercise
  - 0\_exercise\_question100\_list.txt:練習問題:PCのローカル側にコピーして開いてください。
  - PC側のローカルでさくらエディタなどで回答を作成して、pi側にsamba経由で移動させてプログラムを動作させ、エラーがなくなったら模範解答を見てください。



## ⑥. jupyter notebook

- 概要: プログラムを対話型で記録できる汎用型のメモ帳。

- インストール方法

\$ pip install jupyter

- 運用方法

\$ cd 使用するディレクトリ

\$ **jupyter notebook** jupyter起動

chromeブラウザが起動し、**http:// 127.0.0.1:8888/?token=xxx**(トークン認証が追加)  
セルにプログラムを記載して▶でrun, 結果がそのノート上に表示、グラフの出力も可能

参考サイト

<https://qiita.com/takuyanin/items/8bf396e7b6b051670147>

