

# 抜粋編

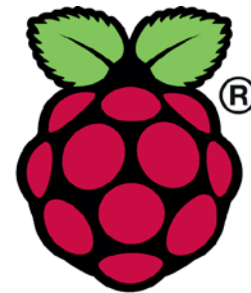
## Pythonで組み立てるFMラジオキット

～Pythonプログラミングを完全マスターするための実践キット～

## Python学習編



FMラジオ



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://toho-ho-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>

抜粋編になります。目次と内容は一致しません。

## 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

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。

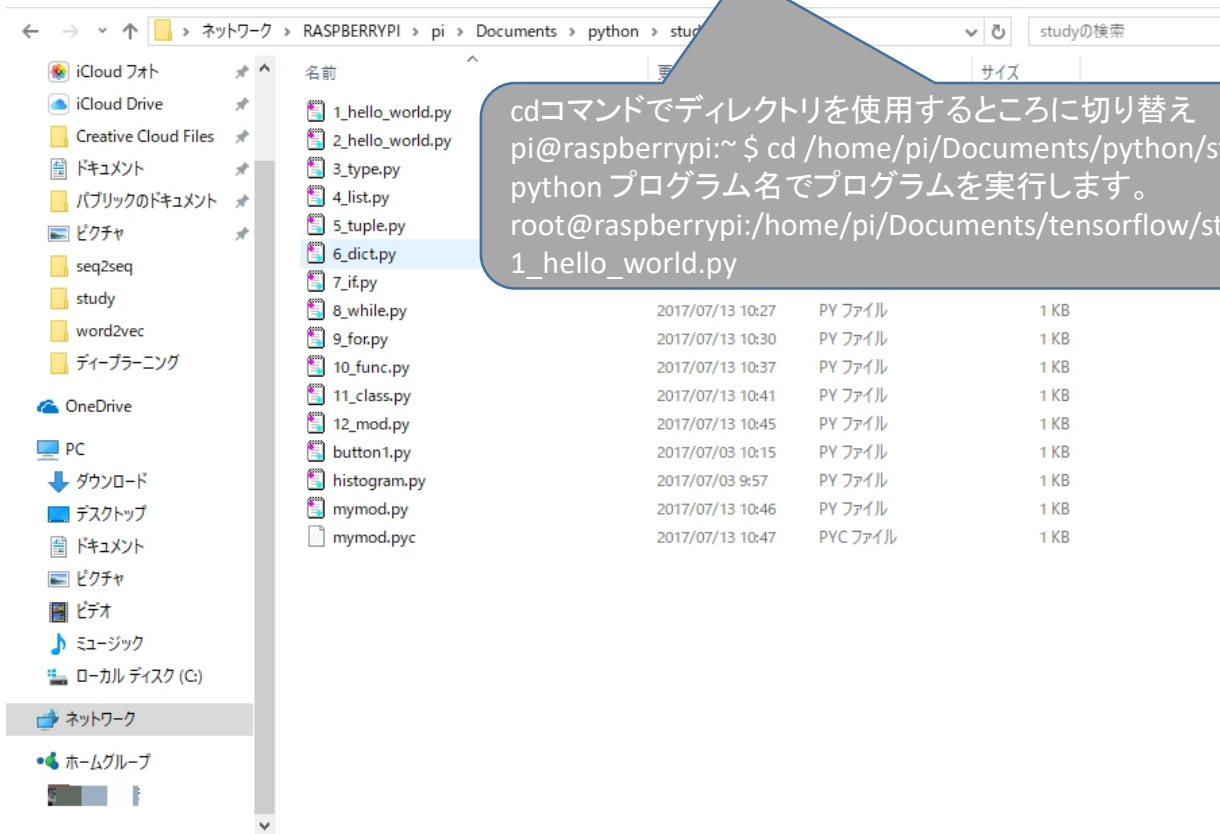
	Python2	Python3	コメント
Print文	print 'Hello, World!'	print('Hello, World!')	3は()が必ず必要になります
xrange	range, xrange	rangeのみ	
unicode	u"unicode"	エラー テキストはstr データはbytes	
整数除算	3/2=1 3/2.0=1.5	3/2=1.5 3/2.0=1.5	分母の数字をfloat, intをきちんと定義する必要がある
計算速度	早い	遅い	

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



## 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

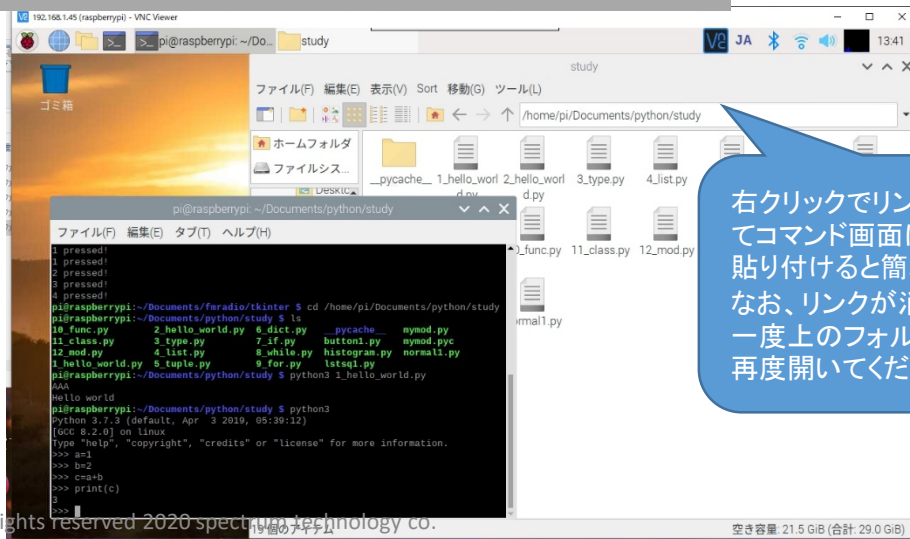
#### • コマンド

```
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



右クリックでリンクをコピーしてコマンド画面にcdの後に貼り付けると簡単です。なお、リンクが消える場合は一度上のフォルダに戻って再度開いてください

## python基礎編

### ①. 概要

仮想環境は構築していません。

- 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:~/Documents/python/study $ python3
Python 3.7.3 (default, Apr 3 2019, 05:39:12)
[GCC 8.2.0] 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

- >>>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

## • エンコード宣言

# coding: utf-8    1行目で使用するエンコードを宣言します。

## • インテンド

pythonの場合インテンドが重要になります。C++などの{}の役割と同じです。スペース(空白4個)又はタブを使用します。

```
if a == 5:
```

```
    print "AAA"
```

#if文の対象はこの文まで。

```
print ("Hello world")
```

#if文の対象外

## • コメント

プログラムの説明するためにコメントを加えます#で以降がコメントになります  
また複数文をコメントアウトする場合は、"""で囲みます。インテンドを同じ位置にします。

## 1\_hello\_world.py

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

```
a=5
```

```
if a == 5:
```

```
    print "AAA" # if文の対象
```

```
print ("Hello world") # 文字列やコメントに日本語を使えるようになります
```

```
""" a=1
```

```
b=3
```

```
"""
```



## 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")
```

アンダーバー2個になります

- プログラムの実行文

```
if __name__ == "__main__":  
  
    main()
```

#\_\_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

### ①. 概要

- プログラム作成ツール

- エディタ

- 複数人で作成する場合は、別のツールをお勧めしますが、一人で開発する場合は、さらエディタなどを使うと簡単です。特に文字の一括置換には便利です。色が変わらないのでその点は他のものが便利かもしれません



- eclipse

- 統合型の開発環境
    - pydevのプラグインを入れて使用

```
1 #coding: utf-8
2 import serial
3 import datetime
4 import time
5 import sys
6 import matplotlib.pyplot as plt
7 import matplotlib.dates as md #time scale
8 import dateutil
9 from tkinter import *
10 from decimal import (Decimal, ROUND_DOWN) #RSSI round_down
11 import codecs # hexascii convert ascii
12 from tkFileDialog import * #terminal
13 from ScrolledText import * #terminal
14 import sys, os.path #terminal
15 $sys.path.append("/home/pi/Documents/wisun")
16 from userdata.userdata_send import lpx0 #loc
17 import csv #csv file wir
18
19 # menu func
20 root = Tk()
21 root.title(u"Wi-Sun radio tester menu")
22 root.geometry("500x500")
23 root.option_add('*font', 'FixedSys 14')
24 #logo
25 savedir=""
26 filename=savedir+"ST_banner2.png"
27 im=PhotoImage(file=filename)
28 Label(root,image=im,width=400,height=60).grid(row=0,column=0,sticky=NW)
29
30 #def char
31 var = IntVar()
32 var.set(0)
33 m=0
34 name_table = {
35     "5. Ping",
36     "10. Set
37 }
38 for m, name in enumerate(name_table):
```

## 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。

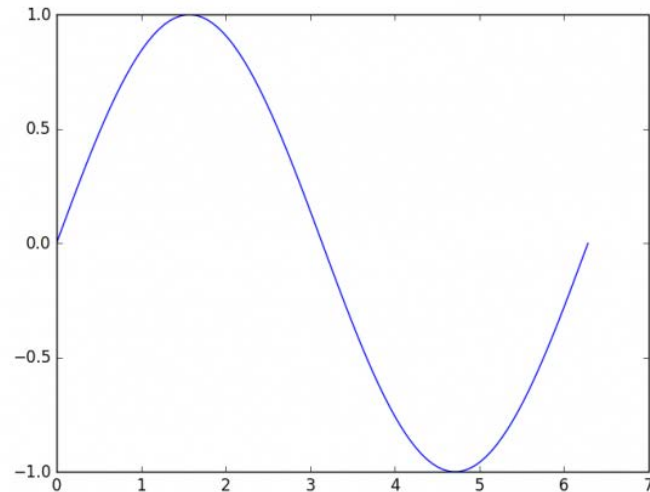
```
import numpy as np  
from matplotlib import pyplot
```

```
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

## ③. 便利なモジュール

- 数値計算: numpy
  - 積分、フーリエ変換、統計など数値計算用モジュールです。test1.py

```
# -*- coding: utf-8 -*-  
import numpy as np  
  
def main():  
    ans = np.random.normal(10,2)  
    print(ans) # 結果を表示  
  
if __name__ == '__main__':  
    main()
```

最小二乗法

正規分布(平均値:10、標準偏差:2):test2.py

```
# -*- coding: utf-8 -*-  
import numpy as np  
import matplotlib.pyplot as plt  
  
def main():  
    X = [1,2,3,4,5]  
    Y = [1.1, 2.1, 2.8, 4.3, 5.1]  
    A = np.array([X,np.ones(len(X))])  
    A = A.T  
    a,b = np.linalg.lstsq(A,Y)[0]  
    plt.plot(X,Y,"ro")  
    plt.plot(X,(int(a)*X+b),"g--")  
    plt.grid()  
    plt.show()  
  
if __name__ == '__main__':  
    main()
```

## python基礎編

### ④. 例

#### • ヒストグラム histogram.py

```
import matplotlib.pyplot as plt
import math

def sqr(x):
    return x * x

data = [66, 59, 62, 64, 63, 68, 65, 59, 68, 64, 65, 51, 67, 64, 83,
        59, 61, 62, 57, 72, 65, 64, 54, 60, 53, 65, 67, 60, 53, 79,
        74, 53, 61, 68, 75, 50, 57, 55, 66, 56, 55, 61, 70, 71, 49,
        69, 70, 80, 73, 72]

n = len(data)      # The number of scores
print ("n =%d"%n)

print ("Data...")
i = 1
for x in data:
    print("%5d:%i + " %f"%x)
    i+=1

plt.title("Histogram") # The title of the histogram
plt.xlabel("Score")    # The label of the x-axis
plt.ylabel("Frequency") # The label of the y-axis
plt.hist(data, bins = 10) # Making the histogram
plt.show()            # Show the histogram
```

出典

<http://y-okamoto-psy1949.la.coocan.jp/Python/sampleprgs/>

## python基礎編

### ④. 例

- ボタン表示

button1.py

```
import Tkinter

class my_button(Tkinter.Button):
    def __init__(self, master=None, cnf={}, **kw):
        Tkinter.Button.__init__(self, master, cnf, **kw)
        self.configure( font = ("", 14), bg = "yellow", fg = "red" )

root = Tkinter.Tk()
button1 = my_button(root, text="MyButton1")
button1.pack()
button2 = my_button(root, text="MyButton2")
button2.pack()
root.mainloop()
```

## 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。

