

## Python 教材の紹介

デスクトップにある「Python WS」のフォルダの中に「全部入り.txt」というファイルがあり、このプリントのプログラムがすべて入っています。このファイルから必要な部分をコピーして、Visual Studio に貼り付けてください。

### 1 乱数を使ってみる

乱数を使うには `random` というモジュールを使います。`random.choice` はリストの中から 1 つの値をランダムに選びます。`random.randint(a,b)` は `a` 以上 `b` 以下の整数をランダムに選びます。

random1.py

```
import random
kuji = ['大吉', '吉', '小吉', '凶']
message = random.choice(kuji)
print(message)
```

random2.py

```
import random
print(random.randint(1,6))
```

ではサイコロを 10 回振ってみるプログラムを作りましょう。このような、回数の決まっている繰り返しは `for i in range(10):` のように書きます（この場合は 10 回。 `i` は別の文字でもかまいません）。

行末の「:」（コロン）を忘れないでください。また、コロンのあとで Enter を押して改行すると行の頭が少し空きます。これは繰り返す部分を表すための空白なので、消さないでください。

random3.py

```
for i in range(10):
    print(i)
```

random4.py

```
import random
for i in range(10):
    print(random.randint(1,6))
```

最初のおみくじで、「凶」が出たときには励ましの言葉を添えることにしましょう。このようにある条件を満たすかどうかで行う処理を変える場合には `if` を使います。やはり行末の「:」を忘れないように気をつけてください。また、イコールが 2 つになっていることにも注意してください。

random5.py

```
import random
kuji = ['大吉', '吉', '小吉', '凶']
message = random.choice(kuji)
print(message)
if message == '凶':
    print('ドンマイ!')
else:
    print('良かったね!')
```

## 2 タートルグラフィックス

タートルグラフィックスでは `turtle` というモジュールを使います。 `turtle0.py` のように `import turtle` としてもいいのですが、それだと毎回 `turtle` と書かなくてはいけなくて面倒なので、 `turtle1.py` のように `turtle` モジュールのオブジェクトを全部取り込んでしまうことにします。

— turtle0.py —

```
import turtle
turtle.shape('turtle')
turtle.forward(200)
turtle.right(90)
turtle.done()
```

— turtle1.py —

```
from turtle import *
shape('turtle')
forward(200)
right(90)
done()
```

最後にカメが90度右を向いたのがわかりますか。これを繰り返せば三角形や四角形などいろいろな図形が描けます。繰り返しには前に使った `for...` が使えます。うまくいったら、角度や回数を変えていろいろな図形を描いてみてください。

— turtle2.py —

```
from turtle import *
shape('turtle')
for i in range(5):
    forward(200)
    right(144)
done()
```

カメに散歩をさせてみましょう。 `turtle3.py` では向きを  $\pm 90^\circ$  の範囲で変化させています。 `turtle4.py` は元の場所に帰ってくるまでひたすら散歩を続けます。 `distance(0,0)` は出発点 (0,0) からの距離で、これが0.1より小さい値になったら繰り返しを抜け出す (`break`) ことにしています。

— turtle3.py —

```
from turtle import *
import random

shape('turtle')
for i in range(200):
    angle = random.randint(-90,90)
    forward(10)
    right(angle)
done()
```

— turtle4.py —

```
from turtle import *
import random
shape('turtle')
color('red')
stamp()
color('green')
while True:
    forward(10)
    right(random.choice([-90,0,90]))
    if distance(0,0) < 0.1:
        break
done()
```

### 3 ネットワーク通信

Python ではネットワーク通信のプログラムも簡単に書けます。待機して接続を待つコンピュータがサーバ、接続してサービスを要求するコンピュータがクライアントです。サーバ用のプログラムが `server.py`、クライアント用のプログラムが `client.py` です。サーバ用のプログラムは長めだし、ファイアウォールに穴を開けたりといった準備が必要になるので、今日はみなさんはクライアント用のプログラムを作って実行してみてください。

「ここはサーバの IP アドレスに置き換えます」の部分は作る時に指示します。自分の環境で使うときは、サーバのコンピュータで `ipconfig` で IP アドレスを調べてそれに書き換えてください。なお、30000 としているのはポート番号というもので、サーバとクライアントで同じ番号を指定する必要があります。

server.py

```
import socket
import threading

host = 'ここはサーバの IP アドレスに置き換えます'
port = 30000

def talk(conn,addr):
    print('Connected from {}'.format(addr))
    conn.sendall(' なにか言って! '.encode())
    data = conn.recv(1024)
    message = '「{}」って言いましたね! '.format(data.decode())
    conn.sendall(message.encode())
    conn.close()

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((host, port))
    s.listen(10)
    while True:
        conn, addr = s.accept()
        threading.Thread(target=talk, args=(conn,addr)).start()
```

client.py

```
import socket

host = 'ここはサーバの IP アドレスに置き換えます'
port = 30000

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((host, port))
    message = s.recv(1024)
    print(message.decode())
    message = input()
    s.sendall(message.encode())
    message = s.recv(1024)
    print(message.decode())
```

## 4 GUI

ここまでは操作をテキスト画面で行う CUI (Character User Interface) のプログラムばかりでしたが, GUI (Graphical User Interface) のプログラムも簡単に作ることができます。

— tk1.py —

```
import tkinter as tk
root = tk.Tk()
root.geometry('200x100')
label = tk.Label(text='LABEL')
button = tk.Button(text = 'PUSH')

label.pack()
button.pack()

tk.mainloop()
```

tk1.py はボタンを押しても何も起きません。tk2.py では, ボタンを押したらラベルの文字列を書き換える関数 `greeting` を呼び出すことにしました。

— tk2.py —

```
import tkinter as tk

def greeting():
    label.configure(text='こんにちは')

root = tk.Tk()
root.geometry('200x100')
label = tk.Label(text='LABEL')
button = tk.Button(text = 'PUSH', command = greeting)

label.pack()
button.pack()

tk.mainloop()
```

たとえば最初のおみくじと組み合わせてみるとこんなこともできます。

— tk3.py —

```
import tkinter as tk
import random

def greeting():
    kuji = ['大吉', '吉', '小吉', '凶']
    label.configure(text=random.choice(kuji))

root = tk.Tk()
root.geometry('200x100')
label = tk.Label(text='LABEL')
button = tk.Button(text = 'PUSH', command = greeting)

label.pack()
button.pack()

tk.mainloop()
```