

# レポートの書き方 & Pythonプログラミングの初歩 (第1回)

氏名 山口 泰  
クラス 理科1類1組  
学生証番号 J4-123456

今後、レポートは、本書式にしたがって作成するものとする。各課題ごとに、プログラムリスト、実行コマンド、実行結果、考察を書くこと。ただし、プリント中にプログラムの全文が示されている(例題プログラムの実行の場合)には、プログラムリストは不要である。

## ○プログラムリスト

プリントのプログラムはリストを載せなくても良いが、第1回は練習の意味でリストをつけること。穴埋めや修正を施したプログラムは、リストを載せること。例のプログラムでも、うまく動かない場合には、リストを載せた方が良好だろう。

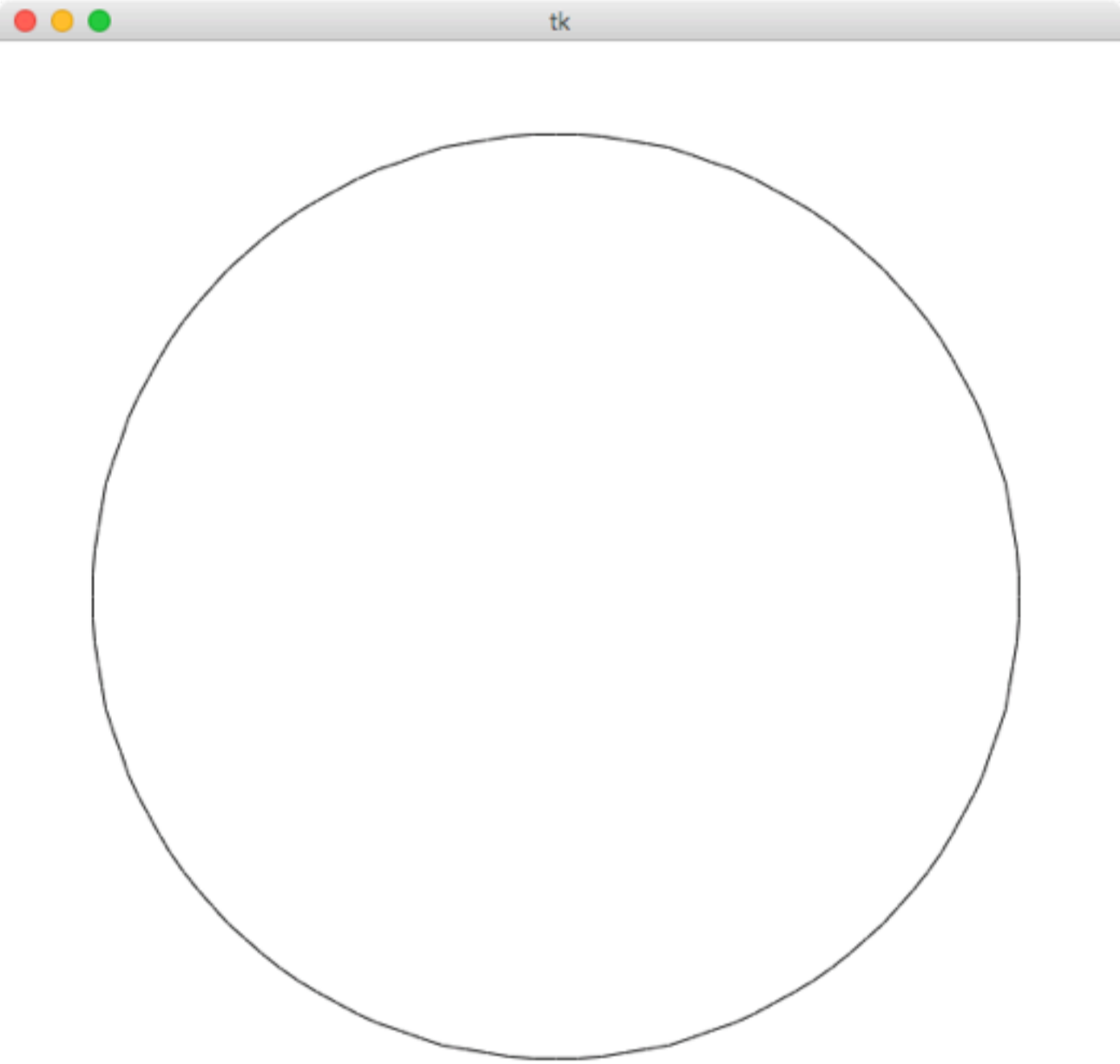
```
1 from tkinter import *          # tkinterモジュールのimport
2 import sys                     # sysモジュールのimport
3 import math                    # mathモジュールのimport
4
5 W, H = (600, 600)              # canvasの幅と高さ
6 R    = 250                     # 円の標準半径
7
8 def circle(cen = (W/2, H/2), r = R): # 円周上 (正多角形) 頂点の作成
    : (ここでは参考なのでプログラム省略)
42
43 if __name__ == '__main__':    # 起動の確認 (コマンドラインからの起動)
44     main()                    # main関数の呼出
```

## ○実行コマンド

```
pc123456m:~ 1234567890$ python circle.py
# of points -> 128
```

## ○実行結果

ターミナル上のメッセージの場合は、ここで表示。



## ○考察

プログラムリストには、行番号が振ってあると考察が書きやすいと思われる(考察を書くのに支障がなければ、行番号を振らなくてもよい)。簡単に行番号つける方法としては、コマンド `cat -n` がある。たとえば、元のファイル名が `orginal.py` であれば、

```
pc123456m:~ 1234567890$ cat -n original.py > numbered.py
```

のようにリダイレクションを用いて、`numbered.py` というファイルを作成する。ただし、`cat -n` コマンドを利用するには、改行コードが `NL` でなくてはならない。たとえば、Emacs の場合、`Control-x <Return> f` として、ファイルのコードに `utf-8-unix` を指定する必要がある。

実行結果はマウスの左ボタンを使って、Cut and Paste するのが最も簡単だろう。ただし、「<」と「>」の2つの記号は、そのままだとHTMLタグと理解されてしまっていて、うまく表示されない。したがって、「<」は「&lt;」、 「>」は「&gt;」に置換することが望ましい。

## □課題や授業に関して

### ○レポート作成に要した時間

およそ45分。

### ○特に苦労した点

プログラムの入力と実行は簡単だったが、考察を書くのに手間取った。

### ○授業についての感想や希望

ここでは授業を改善するための材料として、授業についての感想や希望を書いて貰いたい。なるべく気をつけて欲しいのは、「何が」「なぜ」「どのように」というように、できるだけ具体的に書くということである。たまに「授業が難しくついていくのが大変」(表現は違っていても内容的には同程度)などの感想が見かけられる。しかし、これでは有用な情報とは言いがたい。授業の中のどの部分が難しく感じられるのか、どうすれば改善できると感じられるのかなどを書いて貰わないと対策が採れないのは容易に想像できると思う。