Python Programming in non-English Language to improve readability for non-native English speaker

~ **using** a Set of 18 Turtle Demo Programs Translated into Traditional Chinese as an Example

Renyuan Lyu, Chang Gung University
renyuan.lyu@gmail.com,
https://plus.google.com/+RenyuanLyu

Abstract

- In this project, a set of 18 turtle demo programs has been translated into traditional Chinese (tc) as an example to show the possibility to write Python code conveniently in non-English language. In such a way, it will improve code clarity and readability for non-native English speaker, according to Python PEP 3131. In personal belief, this will definitely attract more people without English fluency to programming world. This project has been done by providing a full list of tc alias (turtle_tc.py) for the official python turtle module and also a tc document file for on-line help functions. A viewer program is also provided to browse them for convinience. The whole set of programs can be found in github.
    - [https://github.com/renyuanL/pythonTurtleInChinese/tree/master/examples/tcExamples2015]

The motivation was partially from PEP 3131

- Python code is written by many people in the world who are not familiar with the English language, or even well-acquainted with the Latin writing system.
- Such developers often desire ***to define classes and functions with names in their native languages***, rather than having to come up with an (often incorrect) English translation of the concept they want to name.
- By using identifiers in their native language, ***code clarity and maintainability*** of the code among speakers of that language improves.
- For some languages, common transliteration systems exist (in particular, for the Latin-based writing systems); for other languages, users have larger difficulties to use Latin to write their native words.

- Original from: [https://www.python.org/dev/peps/pep-3131/]

Python Code in English v.s. in Chinese

```
from turtle import *

print("Hello, this is turtle graphics.")

for i in range(100):
    forward(100)
    left(100)
```

```
from turtle_tc import *

印("哈囉，這是龜作圖。")

for i in 範圍(100):
    前進(100)
    左轉(100)
```

Source encoding of Python 3.0 in UTF-8

- After version 3.0, the Python language has changed its source coding from ASCII to UNICODE (UTF-8)
- This is quite significant because it will be possible that non-English characters can be used as identifiers, which contain names of variables, functions, classes and methods. Here are examples:

```
>>> 甲 =  100
>>> 某數 = 甲 - 10
```

```
>>>印 =  print
>>>範圍= range
```

Keep Python Keyword Unchanged

- Python keywords are usually common seen, short English functional words, used for grammatic purposes.
  - The number of them is about 30, quite few!
- This small set of words cannot be used as identifiers, so they are left as the original forms, i.e., English words.

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True',
'and', 'as', 'assert', 'break', 'class', 'continue',
'def', 'del', 'elif', 'else', 'except', 'finally',
'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
```

A short example of Python in Chinese

```
>>> 印= print
>>> 範圍= range

>>> 某字串= '你好，世界。'
>>> 重複的次數= 10
>>> for 數 in 範圍(重複的次數):
        印(某字串, 數)


你好，世界。 0
你好，世界。 1
你好，世界。 2
你好，世界。 3
你好，世界。 4
你好，世界。 5
你好，世界。 6
你好，世界。 7
你好，世界。 8
你好，世界。 9
>>>
```

A longer example

- A longer example to find prime numbers within 100
- You can **read it aloud**, if you like.
  - Do you mind I do it for you … ??

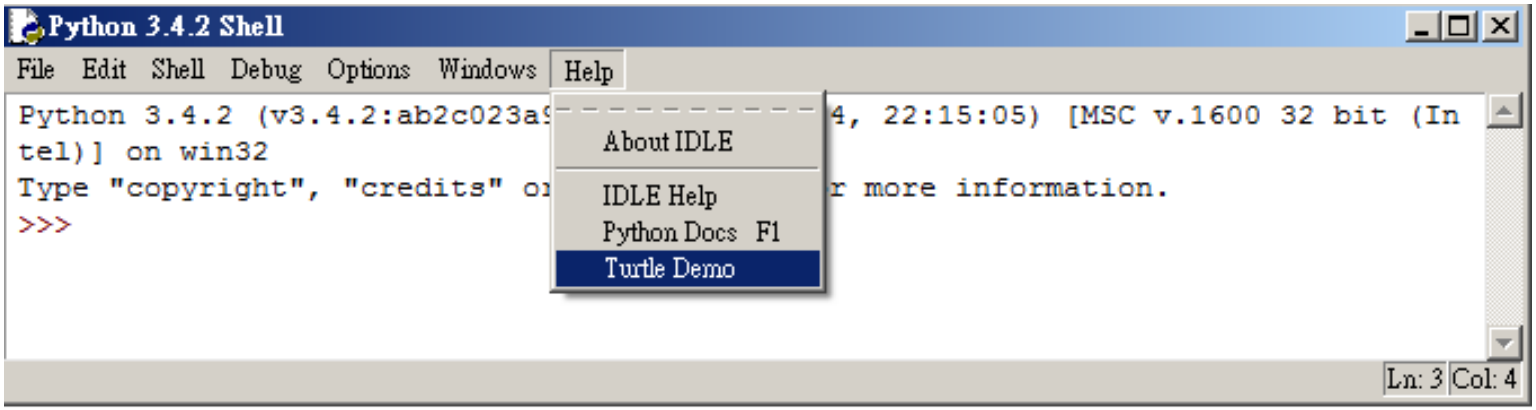
```
'''
prime100.py
本程式可以列出 100 以內的質數。
作者： 呂仁園，2015/03/04
'''
# 內建函數取中文別名
印=       print
範圍=     range

# 自定函數由此開始
def 主程式():

    質數列= []
    for 某數 in 範圍(2,101):
        if 某數為質數(某數):
            質數列 += [某數]
    印('質數列= ',質數列)

def 甲整除乙(甲, 乙):
    if 甲%乙 == 0:
        return True
    else:
        return False

def 某數為質數(x):
    答案= True   # 這是大膽假設，以下為小心求證
    for n in 範圍(2, x):
        if 甲整除乙(x, n):
            答案= False #答案在此逆轉
            break
    return 答案 # 此為 True 或者 False

# 主程式從以下開始執行
主程式()
```

Python Module for Turtle Graphics

- **Turtle graphics** is a term in computer graphics for a method of programming vector graphics using a relative cursor (the "turtle") upon a Cartesian plane.
  - [http://en.wikipedia.org/wiki/Turtle_graphics]
- It was part of the original Logo programming language developed by Wally Feurzig and Seymour Papert in 1966.
  - I am younger than the Turtle ☺
- The **turtle** module is an extended reimplementation of the same-named module from the Python standard distribution up to version Python 2.5.

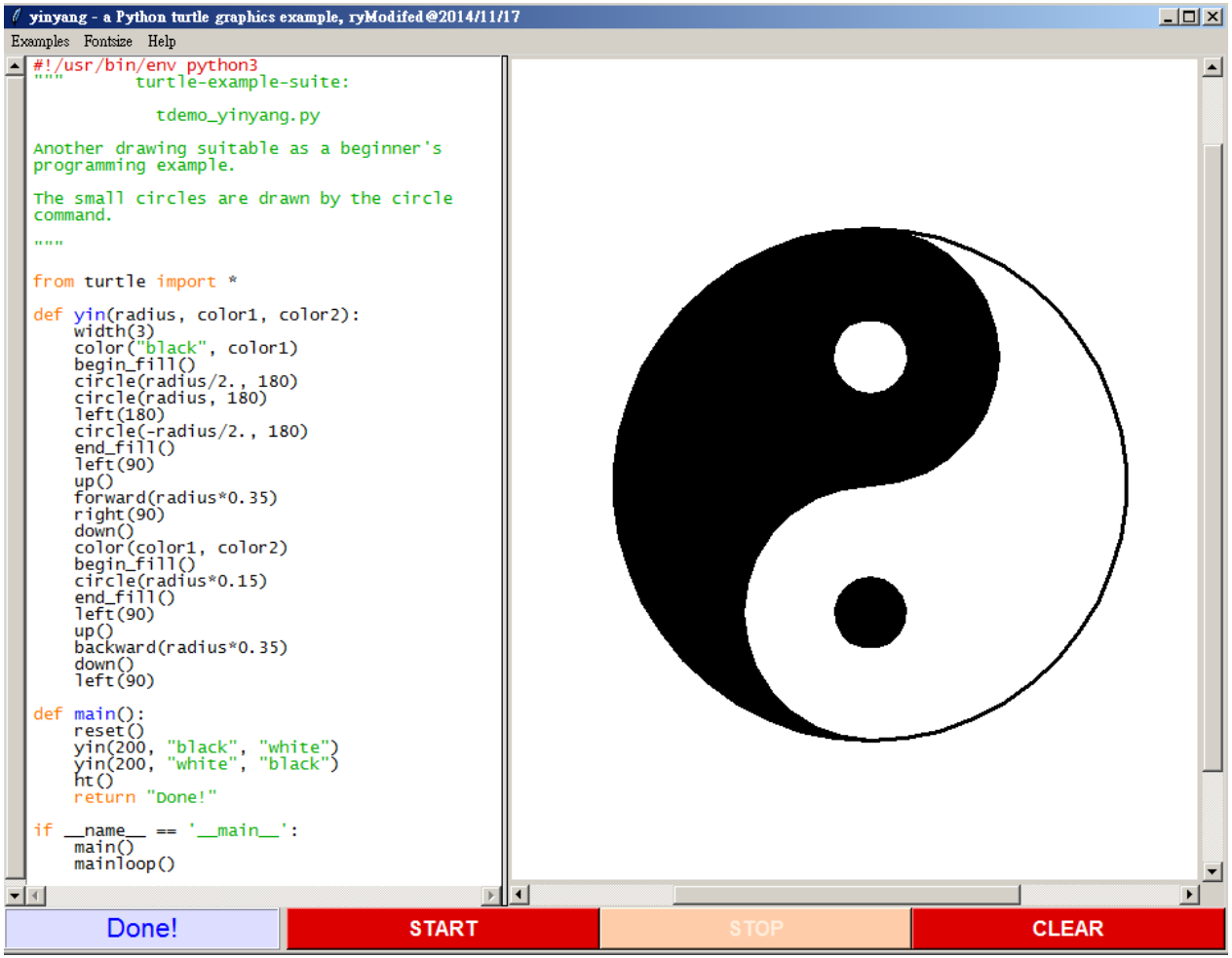
Turtle Demo in IDLE Shell

- Starting from Python 3.4.2, a set of 18 turtle demo programs was promoted to appear in the main menu of IDLE Shell, just below Python Docs within the **Help** submemu.



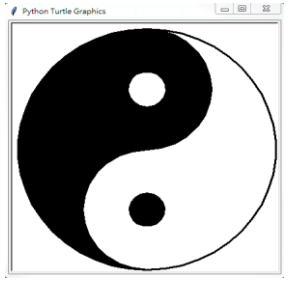
A typical example

- An example from the set of turtle demo programs: **yinyang.py**



Program Translation

- Is that possible we translate those beautiful and well-coded programs from one language into the other one, e.g., from English into traditional Chinese?
- Although pure English programs are globally readable, the Chinese programs are obviously more readable for those who speak Chinese as their native language.




```
from turtle import *

def yin(radius, color1, color2):
    width(3)
    color("black", color1)
    begin_fill()
    circle(radius/2., 180)
    circle(radius, 180)
    left(180)
    circle(-radius/2., 180)
    end_fill()
    left(90)
    up()
    forward(radius*0.35)
    right(90)
    down()
    color(color1, color2)
    begin_fill()
    circle(radius*0.15)
    end_fill()
    left(90)
    up()
    backward(radius*0.35)
    down()
    left(90)

def main():
    reset()
    yin(200, "black", "white")
    yin(200, "white", "black")
    ht()
    return "Done!"

if __name__ == '__main__':
    main()
    mainloop()
```

```
from turtle_tc import *

def 陰(半徑, 顏色1, 顏色2):
    筆寬(3)
    顏色(黑, 顏色1)
    開始填()
    畫圓(半徑/2., 180)
    畫圓(半徑, 180)
    左轉(180)
    畫圓(-半徑/2., 180)
    結束填()
    左轉(90)
    提筆()
    前進(半徑*0.35)
    右轉(90)
    下筆()
    顏色(顏色1, 顏色2)
    開始填()
    畫圓(半徑*0.15)
    結束填()
    左轉(90)
    提筆()
    後退(半徑*0.35)
    下筆()
    左轉(90)

def 主函數():
    重設()
    陰(200, 黑, 白)
    陰(200, 白, 黑)
    藏龜()
    return "完成！"

if __name__ == '__main__':
    主函數()
    主迴圈()
```

Readability counts

- Anybody remember this?
  - If you do not, please …

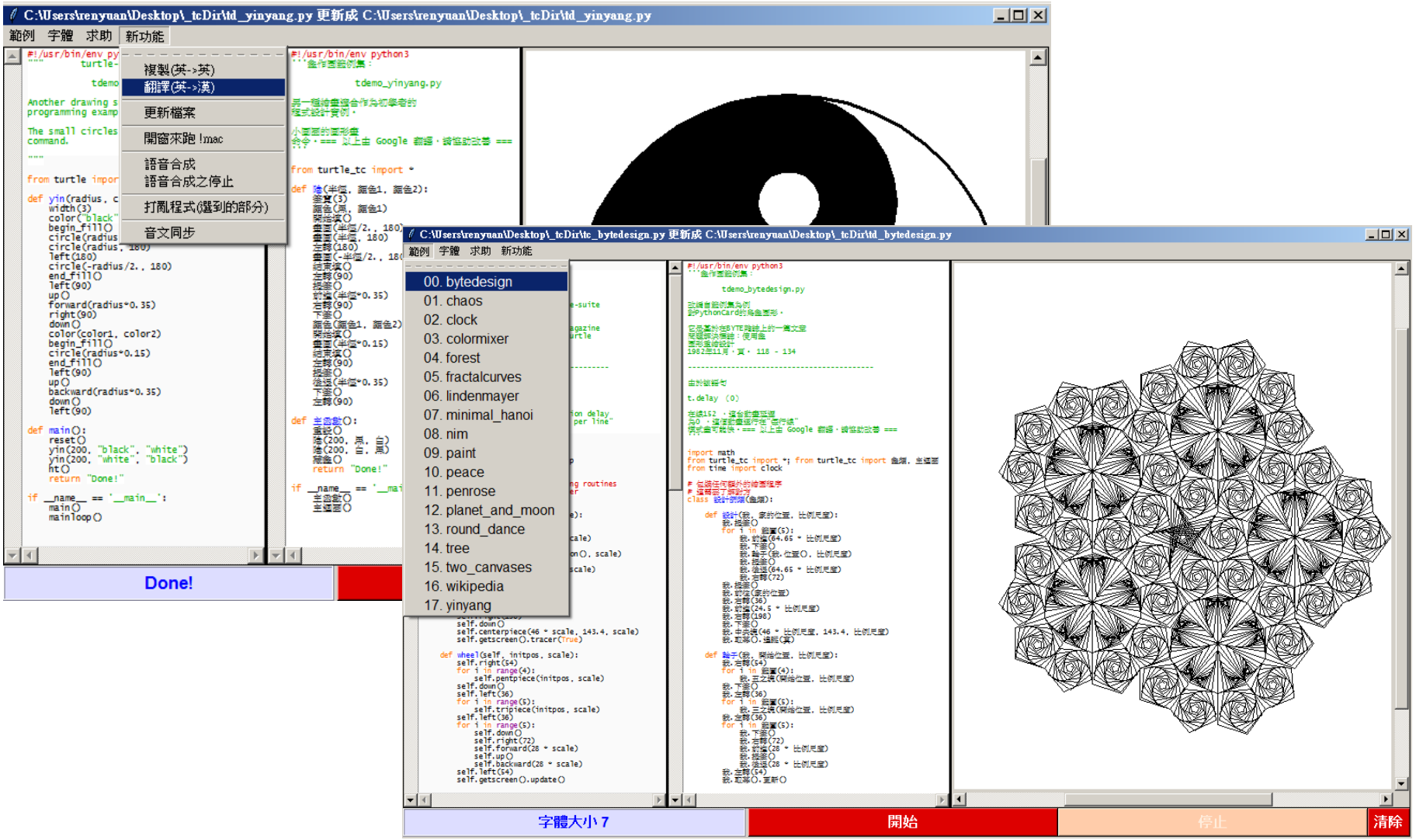
```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
...
...
>>>
```

If Readability really Counts,…

- Then, what can be competent to write programs in your own native or primarily educational language (if it is allowed) !
- It is indeed much more readable for most your friends using the same language, and even for yourself,

Believe me!

Translation of the whole set of 18 Turtle Demo programs

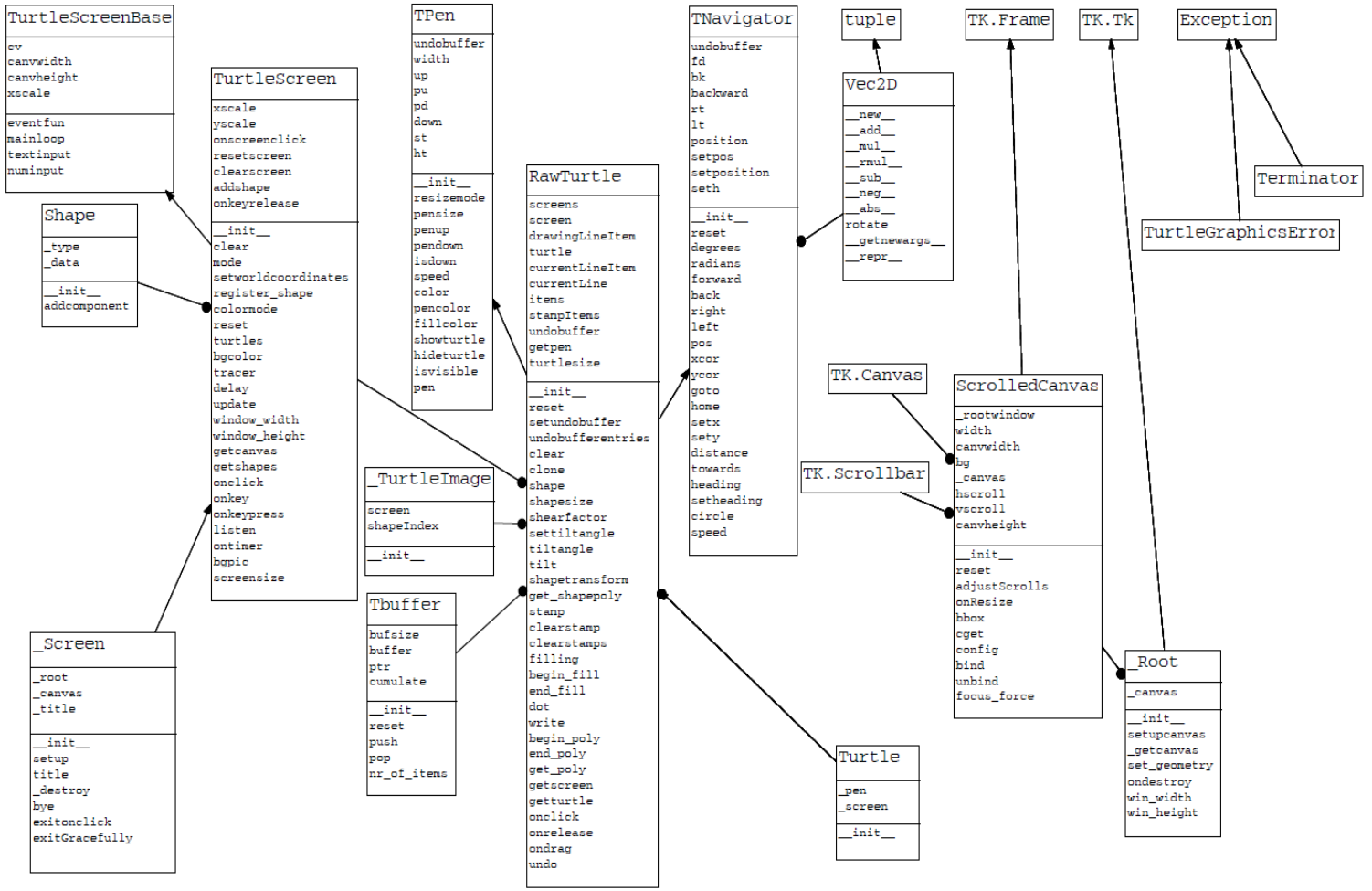


File list of the whole set of 18 programs

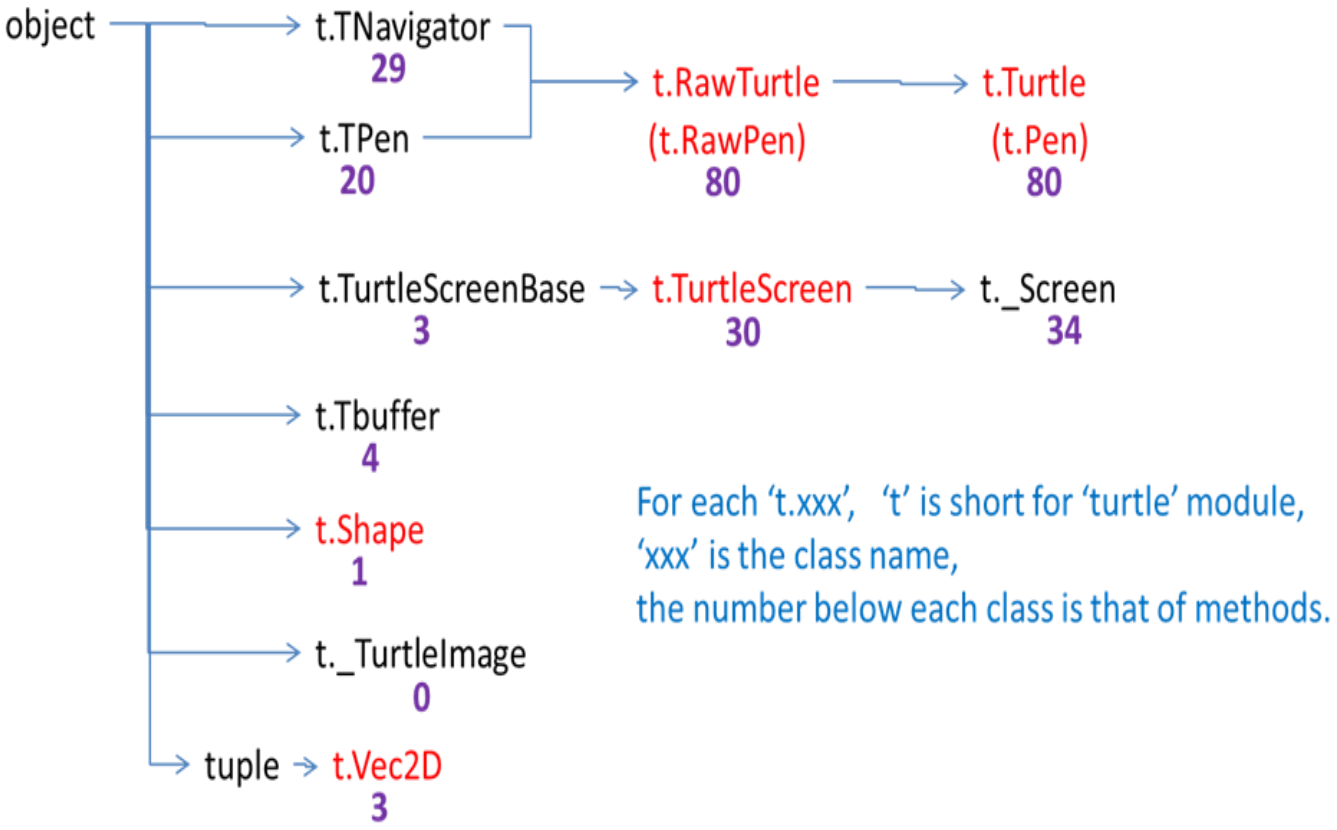
File path @ Windows	Line number
C:\Python34\Lib\turtledemo\bytedesign.py	163
C:\Python34\Lib\turtledemo\chaos.py	60
C:\Python34\Lib\turtledemo\clock.py	133
C:\Python34\Lib\turtledemo\colormixer.py	59
C:\Python34\Lib\turtledemo\forest.py	109
C:\Python34\Lib\turtledemo\fractalcurves.py	139
C:\Python34\Lib\turtledemo\lindenmayer.py	120
C:\Python34\Lib\turtledemo\minimal_hanoi.py	80
C:\Python34\Lib\turtledemo\nim.py	227
C:\Python34\Lib\turtledemo\paint.py	55
C:\Python34\Lib\turtledemo\peace.py	62
C:\Python34\Lib\turtledemo\penrose.py	182
C:\Python34\Lib\turtledemo\planet_and_moon.py	113
C:\Python34\Lib\turtledemo\round_dance.py	87
C:\Python34\Lib\turtledemo\tree.py	64
C:\Python34\Lib\turtledemo\two_canvases.py	55
C:\Python34\Lib\turtledemo\wikipedia.py	66
C:\Python34\Lib\turtledemo\yinyang.py	50
Total line number	1824

Inside the turtle module

- The class diagram of the turtle module



- A simplified class diagram with numbers of methods



Summary of the turtle module

- File path (@ Windows)
  - C:\Python34\Lib\turtle.py
- Number of lines in source code
  - About 4000
  - Rank 2 out of 160 python files in the standard library
- 2 major classes:
  - Turtle
    - About 80 methods
    - E.g., **forward , backward , left , right , …**
  - Screen
    - About 30 methods
    - E.g., **addshape, bgcolor, bgpic, clearscreen, …**
- **Top-level functions**
  - All methods from class Turtle and class Screen are redefine as the top-level functions with a default turtle and screen objects

Alias of the turtle module in traditional Chinese

- Upon the original turtle module, turtle.py, we create an associated module called turtle_tc.py, which provides the alias in traditional Chinese (thus the subscript "_tc" being used) for almost all identifiers (names) in turtle.py

Ref:
https://github.com/renyuanL/pythonTurtleInChinese/blob/master/turtle_tc_alias.py
https://github.com/renyuanL/pythonTurtleInChinese/blob/master/turtle_tc.py

Alias identifiers

- A partial list of the alias identifiers within classes in turtle.py in traditional Chinese

```
龜幕基類= TurtleScreenBase
烏龜螢幕地基類= TurtleScreenBase
龜幕類= TurtleScreen
烏龜螢幕類= TurtleScreen
龜行類= TNavigator
烏龜航行類= TNavigator
龜筆類= TPen
烏龜畫筆類= TPen
原龜類= RawTurtle
粗龜類= RawTurtle
原生龜類= RawTurtle
_幕類= _Screen
_螢幕類= _Screen
幕類= Screen
螢幕類= Screen
開幕= Screen
龜類= Turtle
烏龜類= Turtle
```

```
class TurtleScreen(TurtleScreenBase):

    加形狀= addshape
    背景色= bgcolor
    背景圖= bgpic
    清除= clear
    清除幕= clearscreen
    色模式= colormode
    延遲= delay
    取畫布= getcanvas
:
:
```

```
class TPen(object):

    筆粗= pensize
    筆粗細= pensize
    筆大小= pensize
    筆寬= width
    寬= width
    提筆= penup
    下筆= pendown
:
```

```
class TNavigator(object):

    重設= reset
    前進= forward
    後退= back
    右轉= right
    左轉= left
    位置= pos
    前往= goto
:
```

- A partial list of the alias identifiers in top-level functions within turtle.py in traditional Chinese

```
def x座標():...
def y座標():...
def 下筆():...
def 下筆嗎():...
def 下筆狀態():...
def 位置():...
def 傾斜():...
def 傾斜角度():...
def 前往():...
def 前進():...
def 半徑數():...
def 去到():...
def 點():...
def 龜大小():...
```

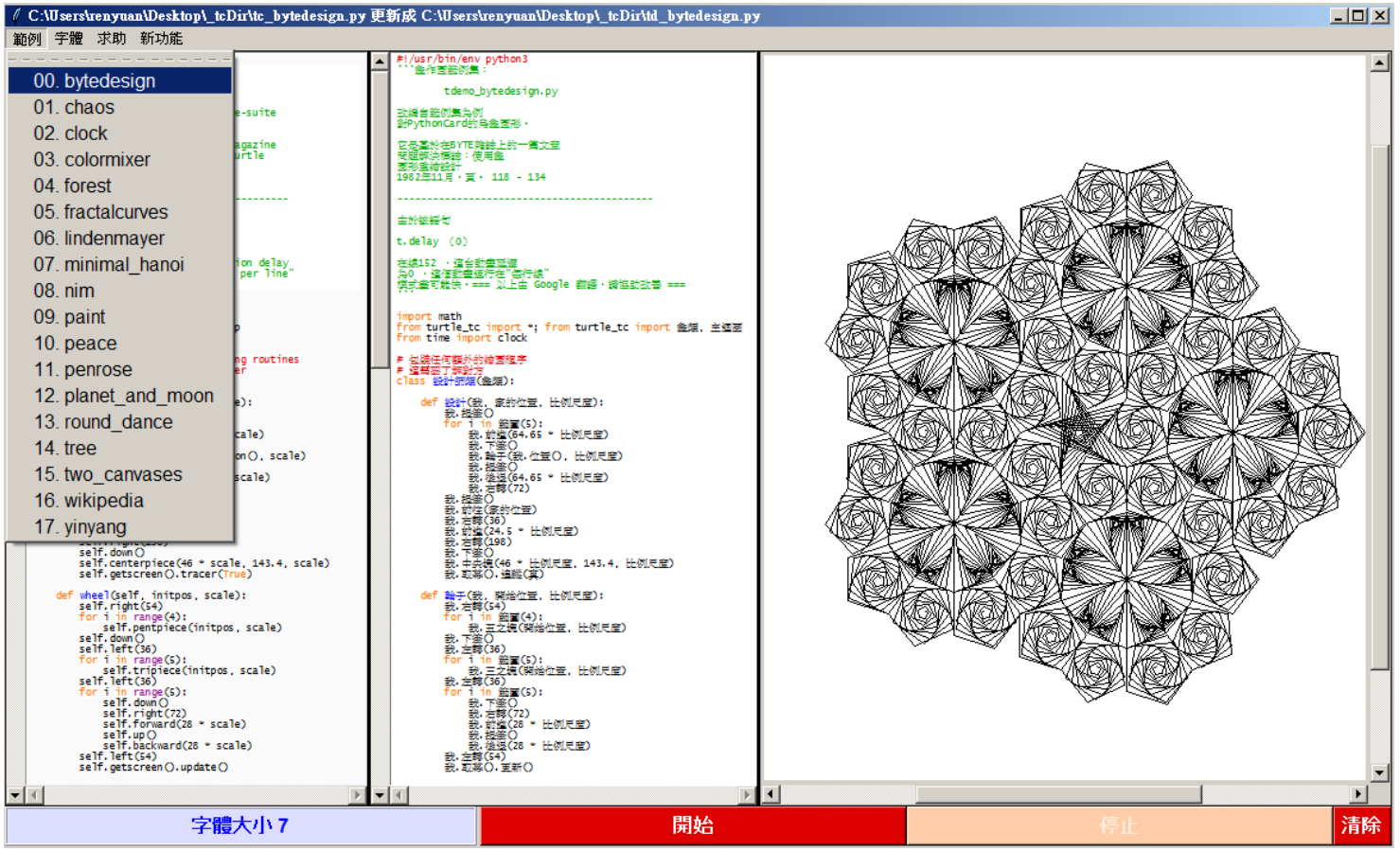
```
def 主迴圈():...
def 做完了():...
def 再見():...
def 加形狀():...
def 取幕寬():...
def 取幕高():...
def 取形():...
def 取形狀():...
def 取畫布():...
def 取龜列表():...
def 在幕點擊時():...
def 重設所有龜():...
def 閉幕():...
def 離開在點擊時():...
def 點擊x結束():...
def 龜列表():...
def 龜群():...
```

- A document file to provide the function of on-line help in Chinese.

```
>>> help(前進)
Help on function 前進 in module turtle_tc:
前進(distance)
    『0053  中文說明』
    龜前進指定的距離。
            別名: 前進 | forward | fd
            參數:
            距離, distance - 一個數字(整數或浮點數)
            龜前進指定的距離, 往龜的頭之方向。
            示例(物件名為「小龜」的實例):
            >>> from turtle_tc import *
            >>> 小龜= 龜類()
            >>> 小龜.位置()
            (0.00,0.00)
            >>> 小龜.前進(25)
            >>> 小龜.位置()
            (25.00,0.00)
            >>> 小龜.前進(-75)
            >>> 小龜.位置()
            (-50.00,0.00)
```

Demo

- http://youtu.be/sQFKjlxw2mw



Conclusion

- We teach Reading, Writing, and Arithmetic to kids in our native or official languages, which are usually not English in many countries, especially in the APAC area.
- Why not we try to teach kids programming in the same language which they have been natively familiar in learning Reading, Writing, and Arithmetic in their daily learning experiences.

Reference

- [1] The whole set of 18 turtle demo programs
  - https://github.com/renyuanL/pythonTurtleInChinese/tree/master/examples/tcExamples2015
  - Demo on youtube
    - http://youtu.be/sQFKjlxw2mw
- [2] renyuanL/**pythonTurtleInChinese**
  - https://github.com/renyuanL/pythonTurtleInChinese
- [3] ChinesePython
  - http://www.chinesepython.org/
- [4] Zhpy
  - https://code.google.com/p/zhpy/
- [5] Computer Programming for Everybody
  - https://www.python.org/doc/essays/cp4e/
- [6] PEP 3131 - Supporting Non-ASCII Identifiers
  - https://www.python.org/dev/peps/pep-3131/