

Articles from FreeSandal

Physical computing : python 《補充》 : IDE 用法.....

2015-02-07 06:02:53 懸鉤子



1990s-2005

Python 語言的創造者吉多·范羅蘇姆 Guido van Rossum 先生非常注重『程式風格』！主導著整體 Python 語言之內蘊。一九九四年元月，那時網際網路還盛行用『地鼠』 Gopher 挖掘資訊的時代，Python 1.0 發布。同年七月 Michael McLay (mclay@eeel.nist.gov)

Wed, 29 Jun 94 10:07:42 EDT 寫了一篇

《 **If Guido was hit by a bus?** 》 — 萬一吉多被巴士撞了？ —

的文章，或許促進了 Python 語言的開始標準化。之後 Python 社群開始稱呼 Guido 『**生殺決策之慈悲獨裁者**』 **BDFL** Benevolent dictator for life :

BDFL is a title given to a small number of open-source software development leaders, typically project founders who retain the final say in disputes or arguments within the community.

無疑的吉多是第一位擁有這個殊榮稱號的第一人。二零零一年三月六日『**Python 軟體基金會**』 PSF Python Software Foundation 成立，一個月後 Python 2.1 發行，所有的發行版開始使用 PSF 授權。事實上吉多的熱情與睿智，更多的展現在『**Python 改進提案**』 **PEPs** Python Enhancement Proposals 的發文裡。有關『可讀性』之『程式風格』的論述主要集中在：

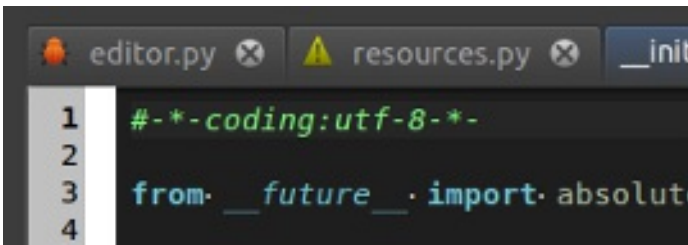
『 **PEP 7 – Style Guide for C Code** 』

『 **PEP 8 – Style Guide for Python Code** 』

也許吉多的理念是：

程式雖是一時一人之寫作，確有多時多人的閱讀，因此『可讀性』是十分重要的事情。

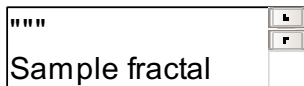
這個『**Ninja-IDE**』有一個特色，就是能找出 Python 程式不符合『**PEP8**』建議的『原始碼』，彰顯了『**Pythonic Way**』的重要組成部分。在此讓我們舉個出自美國 Simpson College 的 **Paul Vincent Craven** 所著的線上書《**Program Arcade Games With Python And Pygame**》第二十章中的『**fractal.py**』為例，比較符不符合『**PEP8**』的異同：

A screenshot of a code editor window. The top bar shows three tabs: 'editor.py' with a red bug icon, 'resources.py' with a yellow warning icon, and a partially visible '_init'. The code in the editor shows line numbers 1 through 4. Line 1: `1 #-*-coding:utf-8-*-`. Line 2: `2`. Line 3: `3 from __future__ import absolut`. Line 4: `4`.

Highlight Static and PEP8 errors in the document, you can also see that the files containing PEP8 errors are shown with an icon in the tab where the file is opened, and files containing code static errors are shown with a bug icon in that tab.

【PEP8 的版本】

Python

A screenshot of a code editor window showing a docstring. The text reads: `"""` on the first line, `Sample fractal` on the second line, and `"""` on the third line. The editor has a standard window title bar and a scrollbar on the right.

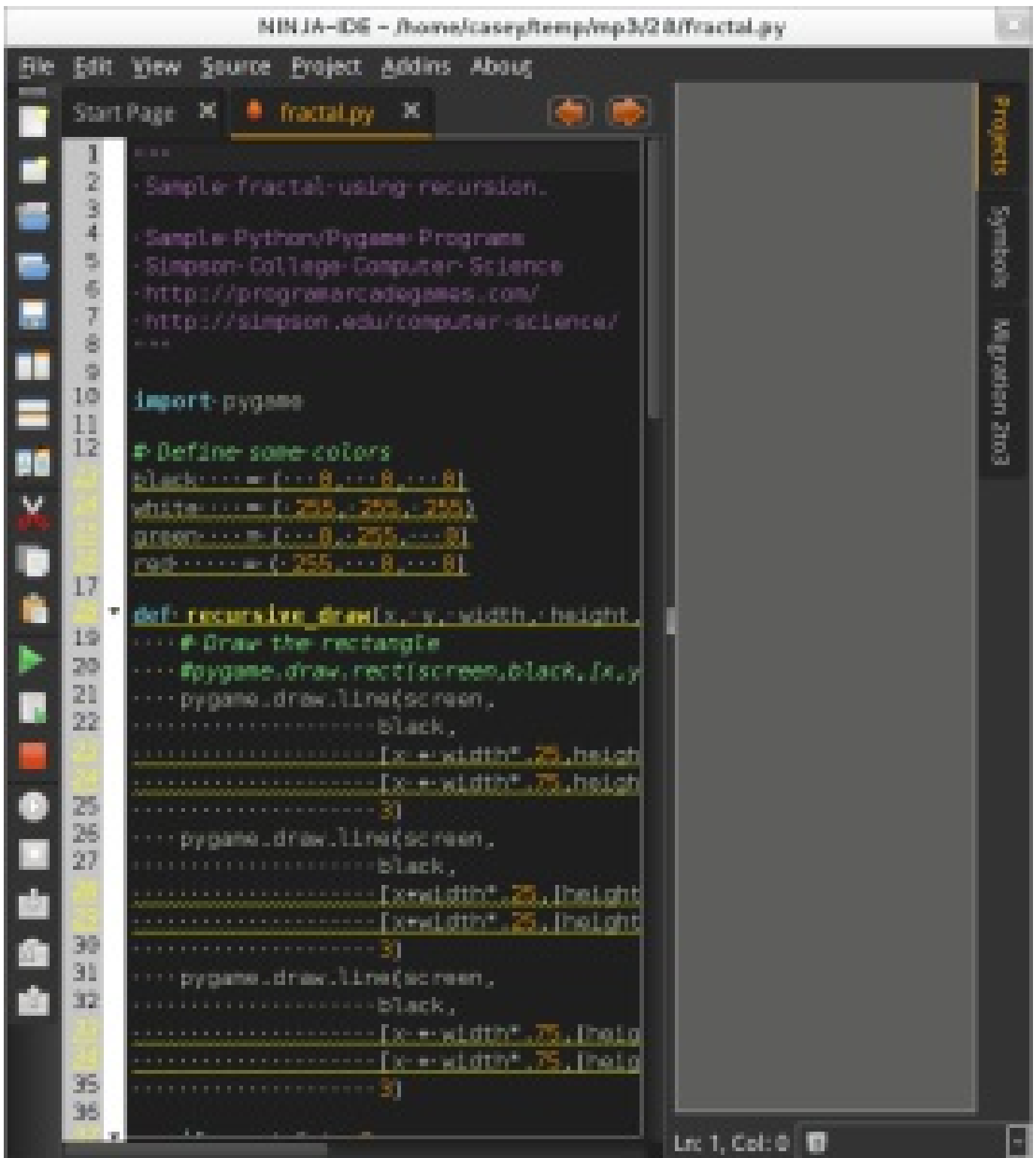
```
1 """
2 Sample fractal using recursion.
3 Sample Python/Pygame Programs
4 Simpson College Computer Science
5 http://programarcadegames.com/
6 http://simpson.edu/computer-science/
7 """
8 import pygame
9
10 # Define some colors
11 black = (0, 0, 0)
12 white = (255, 255, 255)
13 green = (0, 255, 0)
14 red = (255, 0, 0)
15
16
17 def recursive_draw(x, y, width, height, count):
18     # Draw the rectangle
19     #pygame.draw.rect(screen,black,[x,y,width,height],1)
20     pygame.draw.line(screen,
21         black,
22         [x + width * .25, height // 2 + y],
23         [x + width * .75, height // 2 + y],
24         3)
25     pygame.draw.line(screen,
26         black,
27         [x + width * .25, (height * .5) // 2 + y],
28         [x + width * .25, (height * 1.5) // 2 + y],
29         3)
30     pygame.draw.line(screen,
31         black,
32         [x + width * .75, (height * .5) // 2 + y],
33         [x + width * .75, (height * 1.5) // 2 + y],
34         3)
35     if count > 0:
```

```

36     count -= 1
37     # Top left
38     recursive_draw(x, y, width // 2, height // 2, count)
39     # Top right
40     recursive_draw(x + width // 2, y, width // 2, height // 2, count)
41     # Bottom left
42     recursive_draw(x, y + width // 2, width // 2, height // 2, count)
43     # Bottom right
44     recursive_draw(x + width // 2, y + width // 2, width // 2,
45                   height // 2, count)
46 pygame.init()
47 # Set the height and width of the screen
48 size = [700, 700]
49 screen = pygame.display.set_mode(size)
50 pygame.display.set_caption("My Game")
51 # Loop until the user clicks the close button.
52 done = False
53 # Used to manage how fast the screen updates
54 clock = pygame.time.Clock()
55 # ----- Main Program Loop -----
56 while not done:
57     for event in pygame.event.get(): # User did something
58         if event.type == pygame.QUIT: # If user clicked close
59             done = True # Flag that we are done so we exit this loop
60     # Set the screen background
61     screen.fill(white)
62     # ALL CODE TO DRAW SHOULD GO BELOW THIS COMMENT
63     fractal_level = 3
64     recursive_draw(0, 0, 700, 700, fractal_level)
65     # ALL CODE TO DRAW SHOULD GO ABOVE THIS COMMENT
66     # Go ahead and update the screen with what we've drawn.
67     pygame.display.flip()
68     # Limit to 20 frames per second
69     clock.tick(20)
70 # Be IDLE friendly. If you forget this line, the program will 'hang'
71 # on exit.
72 pygame.quit()

```

【 Ninja-IDE 】 顯示之原始版本



【原始程式】

Python

```
"""  
Sample fractal
```

```
1 """  
2 Sample fractal using recursion.  
3  
4 Sample Python/Pygame Programs  
5 Simpson College Computer Science  
6 http://programarcadegames.com/
```

```

7 http://simpson.edu/computer-science/
8 """"
9
10 import pygame
11
12 # Define some colors
13 black = ( 0, 0, 0)
14 white = ( 255, 255, 255)
15 green = ( 0, 255, 0)
16 red = ( 255, 0, 0)
17
18 def recursive_draw(x, y, width, height, count):
19     # Draw the rectangle
20     #pygame.draw.rect(screen,black,[x,y,width,height],1)
21     pygame.draw.line(screen,
22         black,
23         [x + width*.25,height//2+y],
24         [x + width*.75,height//2+y],
25         3)
26     pygame.draw.line(screen,
27         black,
28         [x+width*.25,(height*.5)//2+y],
29         [x+width*.25,(height*1.5)//2+y],
30         3)
31     pygame.draw.line(screen,
32         black,
33         [x + width*.75,(height*.5)//2+y],
34         [x + width*.75,(height*1.5)//2+y],
35         3)
36
37     if count > 0:
38         count -= 1
39         # Top left
40         recursive_draw(x, y, width // 2, height // 2, count)
41         # Top right
42         recursive_draw(x + width // 2, y, width // 2, height // 2, count)
43         # Bottom left
44         recursive_draw(x, y + width // 2, width // 2, height // 2, count)
45         # Bottom right
46         recursive_draw(x + width // 2, y + width // 2, width // 2, height // 2, count)
47
48
49 pygame.init()
50
51 # Set the height and width of the screen
52 size = [700, 700]
53 screen = pygame.display.set_mode(size)
54
55 pygame.display.set_caption("My Game")
56
57 #Loop until the user clicks the close button.
58 done = False
59
60 # Used to manage how fast the screen updates

```

```
61 clock = pygame.time.Clock()
62
63 # ----- Main Program Loop -----
64 while not done:
65     for event in pygame.event.get(): # User did something
66         if event.type == pygame.QUIT: # If user clicked close
67             done = True # Flag that we are done so we exit this loop
68
69     # Set the screen background
70     screen.fill(white)
71
72     # ALL CODE TO DRAW SHOULD GO BELOW THIS COMMENT
73     fractal_level = 3
74     recursive_draw(0, 0, 700, 700, fractal_level)
75     # ALL CODE TO DRAW SHOULD GO ABOVE THIS COMMENT 闡釋
76
77     # Go ahead and update the screen with what we've drawn.
78     pygame.display.flip()
79
80     # Limit to 20 frames per second
81     clock.tick(20)
82
83 # Be IDLE friendly. If you forget this line, the program will 'hang'
84 # on exit.
85 pygame.quit()
```