

國立虎尾科技大學

機械設計工程系

計算機程式 bg1 期末報告

PyQt5 事件導向計算器

PyQt5 Event-Driven Calculator Project

學生：

設計一乙 40623219 XXX

設計一乙 40623220 蔡崇廷

設計一乙 40623221 XXX

設計一乙 40623228 陳永鋆

設計一乙 40623229 陳宥安

設計一乙 40623230 陳柏亦

指導教授：嚴家銘

2017.12.18

摘要

這裡是摘要內容。+ 以 YAML 的方式插入。+ The ‘+’ indicator says to keep newlines at the end of text blocks. + 使用 Markdown 語法。+ 前面使用加號

本研究的重點在於 ...

目錄

摘要	i
目錄	ii
表目錄	iv
圖目錄	v
第一章 前言	1
第二章 可攜程式系統介紹	2
2.1 啟動與關閉	2
2.2 啟動與關閉 2	3
第三章 Calculator 程式	4
3.1 建立對話框	4
3.2 建立按鈕	6
3.3 建立程式碼	7
第四章 Python 程式語法	10
4.1 變數命名	10
4.2 print 函式	11
4.3 重複迴圈	11
4.4 判斷式	11
4.5 數列	11
第五章 PyQt5 簡介	12
5.1 PyQt5 架構	12
第六章 心得	13
6.1 Fossil SCM	13
6.2 網誌心得	13
6.3 Github 協同倉儲	13
6.4 學員心得	13

第七章 結論.....	14
7.1 結論與建議	14
參考文獻.....	15

表目錄

圖目錄

圖 2.1	system-1	2
圖 2.2	system-2	3
圖 3.1	newform	4
圖 3.2	Dialog into ui	5
圖 3.3	qtdesigner	5
圖 3.4	button	6
圖 3.5	grid	6
圖 3.6	digitCilcked	7
圖 3.7	equalClicked	9

第一章 前言

計算器程式期末報告前言

前言內容。

一個範例數學式：

$$\beta = \cos^{-1} \frac{L0^2 + d_{AB}^2 - R0^2}{2 \times L0 \times d_{AB}}$$

關於數學式可以參考這裡：<http://www.hostmath.com/>

提及了某篇刊物 [1] 在這裡。

第二章 可攜程式系統介紹

2.1 啟動與關閉

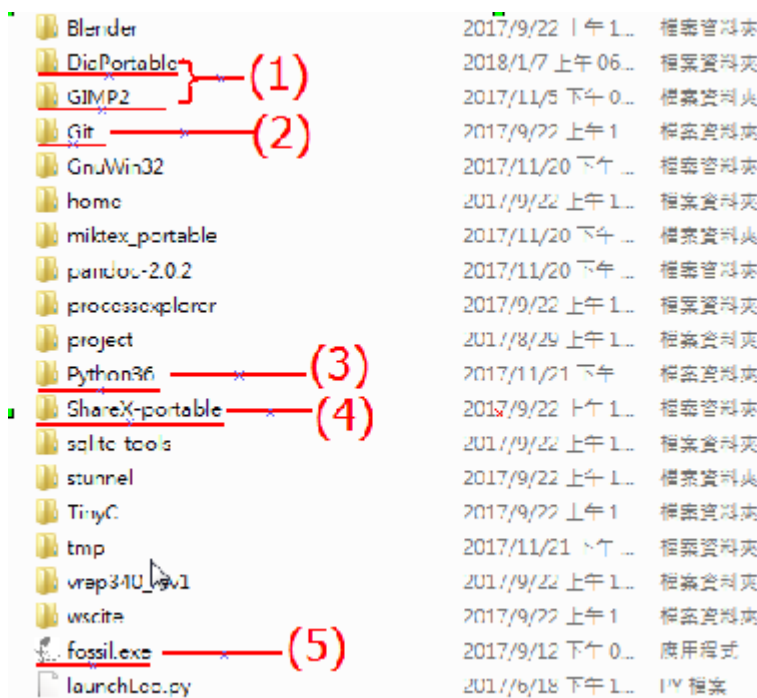


圖 2.1: system-1

可攜程式: 因為在不同的電腦擁有的程式也會有所不同所以使用可攜程式的話可以方便在任何電腦執行自己熟悉的程式也可使用建立自己習慣的開發環境

(1)GIMP2-可以做修剪圖片或是裁切圖片

DiaPortable-可繪製圖形幫助註解圖片

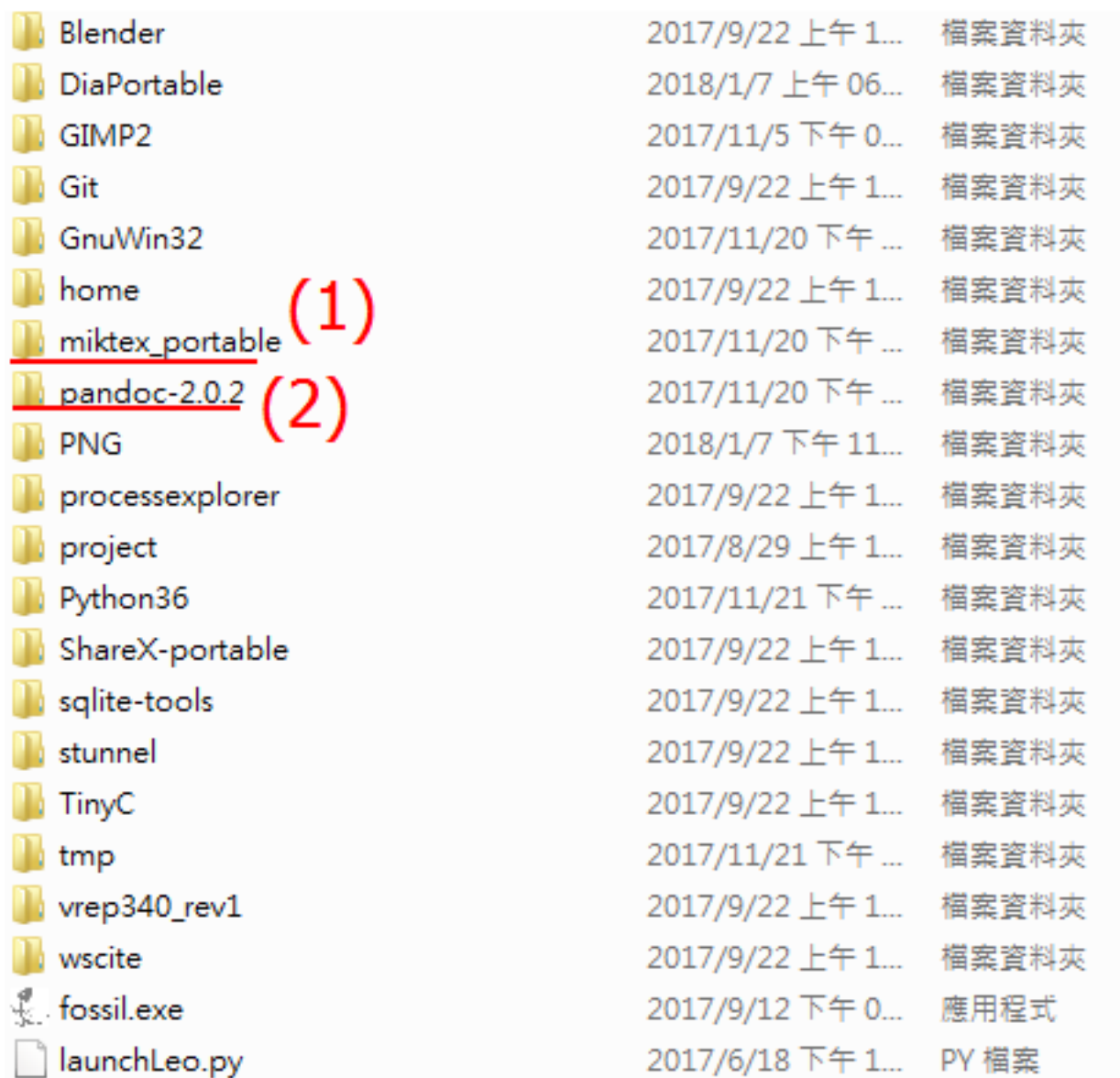
(2)GitHub-SCM(組態管理系統) 的一種, 特點多人協同,gh-pages, 公開 (不公開要花錢)

(3)Python36-在不同電腦都可以進行 Python 的程式開發

(4)ShareX-可截取螢幕畫面, 與錄製影片

(5)Fossil-SCM(組態管理系統) 的一種, 特點 Totally control 完全可以自己控制伺服到客戶端

2.2 啟動與關閉 2



Blender	2017/9/22 上午 1...	檔案資料夾
DiaPortable	2018/1/7 上午 06...	檔案資料夾
GIMP2	2017/11/5 下午 0...	檔案資料夾
Git	2017/9/22 上午 1...	檔案資料夾
GnuWin32	2017/11/20 下午 ...	檔案資料夾
home	2017/9/22 上午 1...	檔案資料夾
<u>miktex_portable</u>	2017/11/20 下午 ...	檔案資料夾
<u>pandoc-2.0.2</u>	2017/11/20 下午 ...	檔案資料夾
PNG	2018/1/7 下午 11...	檔案資料夾
processexplorer	2017/9/22 上午 1...	檔案資料夾
project	2017/8/29 上午 1...	檔案資料夾
Python36	2017/11/21 下午 ...	檔案資料夾
ShareX-portable	2017/9/22 上午 1...	檔案資料夾
sqlite-tools	2017/9/22 上午 1...	檔案資料夾
stunnel	2017/9/22 上午 1...	檔案資料夾
TinyC	2017/9/22 上午 1...	檔案資料夾
tmp	2017/11/21 下午 ...	檔案資料夾
vrep340_rev1	2017/9/22 上午 1...	檔案資料夾
wscite	2017/9/22 上午 1...	檔案資料夾
fossil.exe	2017/9/12 下午 0...	應用程式
launchLeo.py	2017/6/18 下午 1...	PY 檔案

圖 2.2: system-2

(1)miktex_portable-包含了 TeX 及其相關程式, 這些工具是以 TeX/LaTeX 所構成的

(2)pandoc-2.0.2-以命令列形式實現與用戶的互動, 可支援多種作業系統

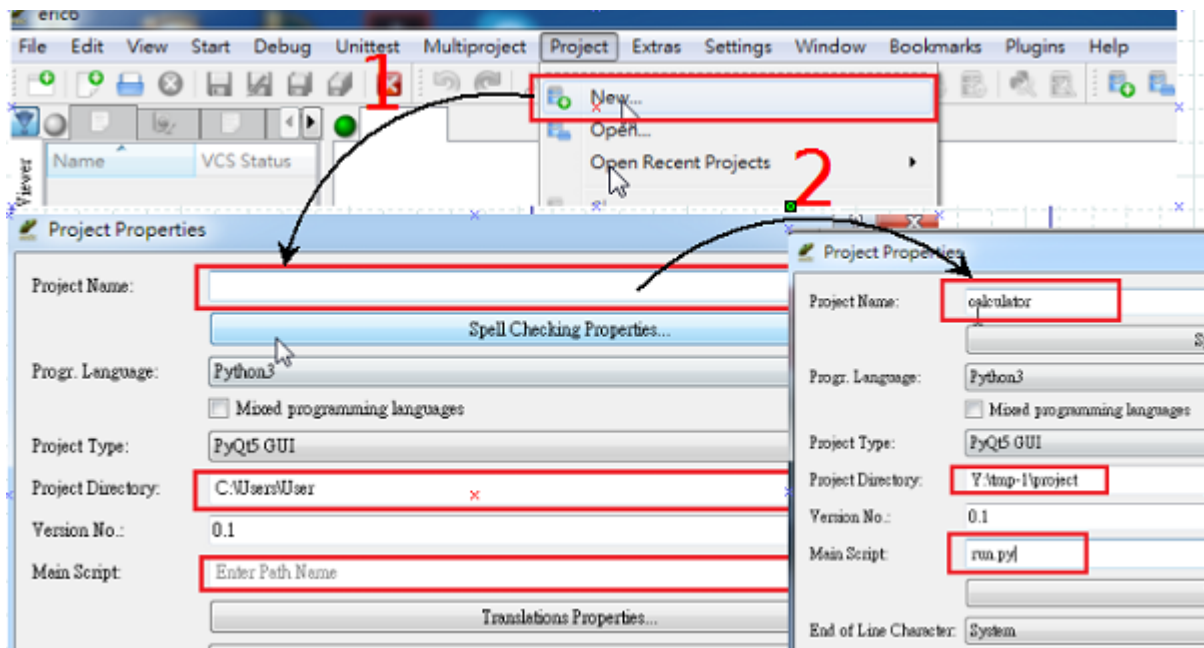
可攜程式系統介紹

第三章 Calculator 程式

Calculator 程式細部說明

3.1 建立對話框

step1



step2

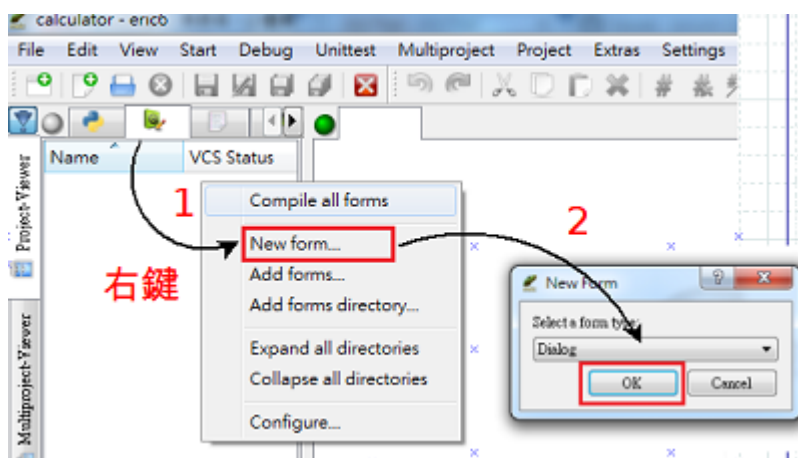
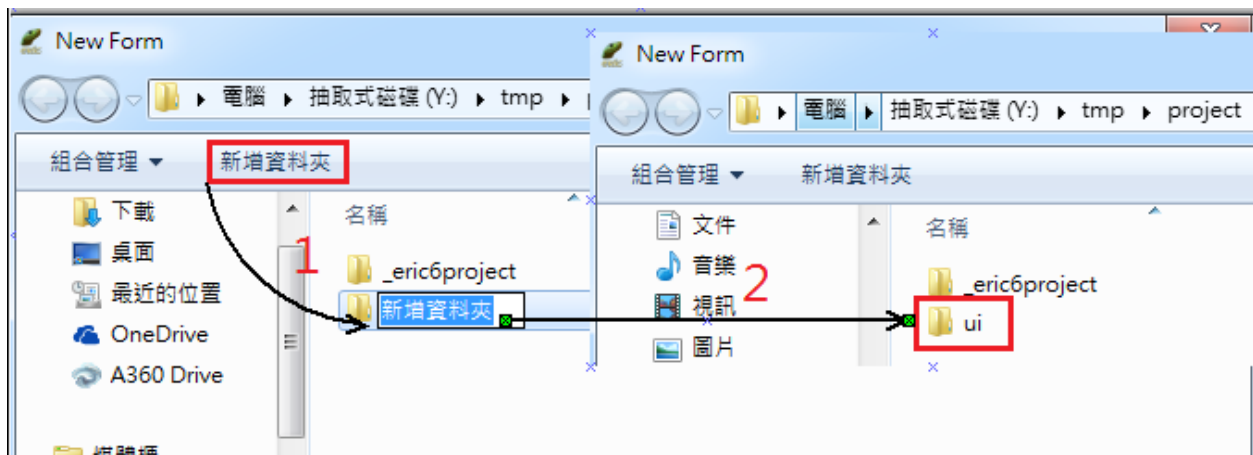


圖 3.1: newform

step3



step4



圖 3.2: Dialog into ui

step5

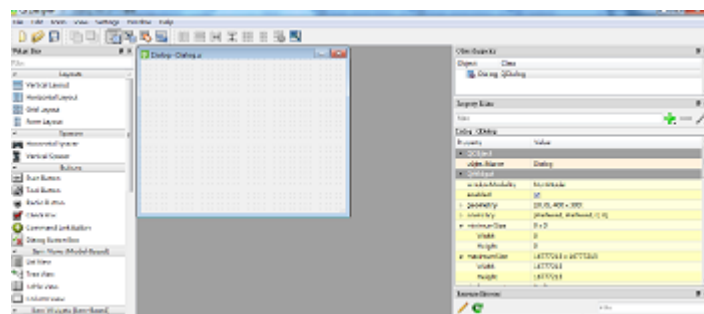


圖 3.3: qtdesigner

3.2 建立按鈕

step1

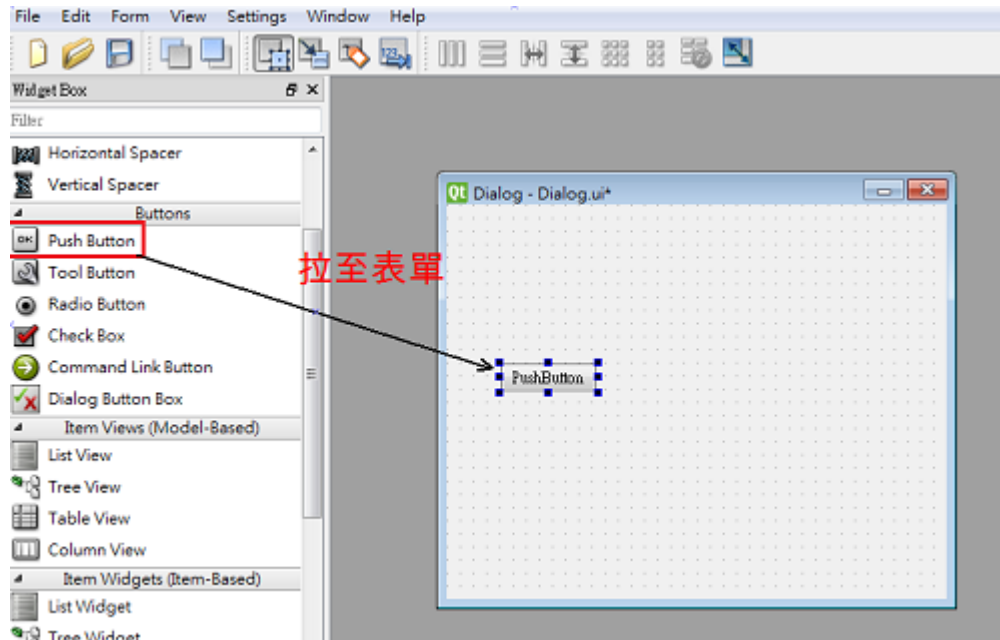


圖 3.4: button

step2

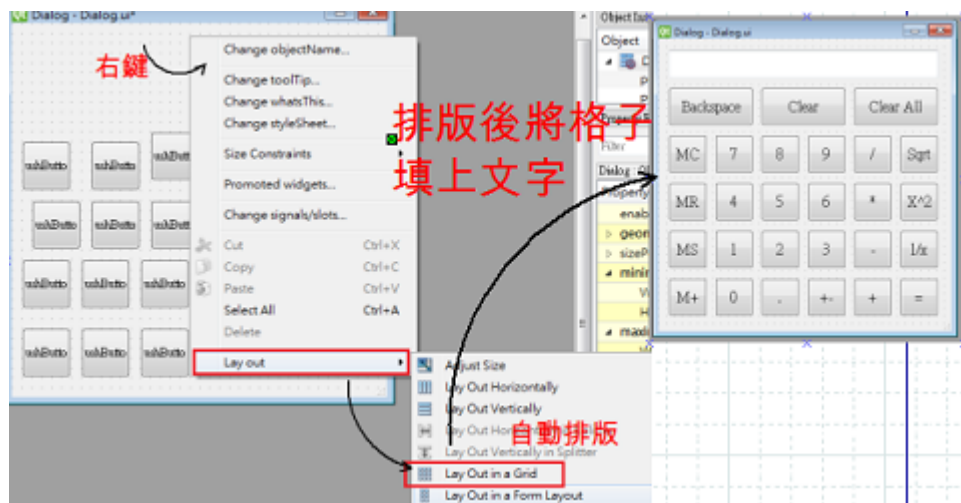


圖 3.5: grid

以上是由 Qt designer 製作

Qt designer 詳細請查閱第五章

3.3 建立程式碼

40623220

40623221

40623228

數字邏輯

```
self.display.setText('0')

num_button = [self.one, self.two, \
self.three, self.four, self.five, self.six, self.seven, self.eight, self.nine, self.zero]

for i in num_button:
    i.clicked.connect(self.digitClicked)
def digitClicked(self):
#40623228
    """
    使用者按下數字鍵，必須能夠累積顯示該數字
    當顯示幕已經為 0，再按零不會顯示 00，而仍顯示 0 或 0.0
    """
    #pass
    clickedButton = self.sender()
    digitValue = int(clickedButton.text())
    if self.display.text() == '0' and digitValue == 0.0:
        return
    if self.waitingForOperand:
        self.display.clear()
        self.waitingForOperand = False
    self.display.setText(self.display.text() + str(digitValue))
```



圖 3.6: digitClicked

```

plus_minus = [self.plusButton, self.minusButton]
for i in plus_minus:
    i.clicked.connect(self.additiveOperatorClicked)

self.pendingAdditiveOperator = ''

def calculate(self, rightOperand, pendingOperator):
    """計算"""
    #pass
    if pendingOperator == "+":
        self.sumSoFar += rightOperand

    elif pendingOperator == "-":
        self.sumSoFar -= rightOperand

    elif pendingOperator == "*":
        self.factorSoFar *= rightOperand

    elif pendingOperator == "/":
        if rightOperand == 0.0:
            return False

        self.factorSoFar /= rightOperand

    return True

    self.sumSoFar = 0.0

def additiveOperatorClicked(self):
    """加或減按下後進行的處理方法"""
    #pass
    clickedButton = self.sender()
    clickedOperator = clickedButton.text()
    operand = float(self.display.text())

    if self.pendingMultiplicativeOperator:
        if not self.calculate(operand, self.pendingMultiplicativeOperator):
            self.abortOperation()
            return

        self.display.setText(str(self.factorSoFar))
        operand = self.factorSoFar
        self.factorSoFar = 0.0
        self.pendingMultiplicativeOperator = ''

    if self.pendingAdditiveOperator:
        if not self.calculate(operand, self.pendingAdditiveOperator):
            self.abortOperation()
            return

        self.display.setText(str(self.sumSoFar))
    else:
        self.sumSoFar = operand

    self.pendingAdditiveOperator = clickedOperator
    self.waitForOperand = True

```

按鍵連接

判斷式

加減邏輯
等號邏輯

40623229

40623230

```

def equalClicked(self):
#40623228
    """等號按下後的處理方法"""
    #pass
    operand = float(self.display.text())

    if self.pendingMultiplicativeOperator:
        if not self.calculate(operand, self.pendingMultiplicativeOperator):
            self.abortOperation()
            return 判斷式子中是否有乘除
        operand = self.factorSoFar
        self.factorSoFar = 0.0
        self.pendingMultiplicativeOperator = ''

    if self.pendingAdditiveOperator:
        if not self.calculate(operand, self.pendingAdditiveOperator):
            self.abortOperation()
            return 判斷式子中是否有加減
        self.pendingAdditiveOperator = ''
    else:
        self.sumSoFar = operand

    self.display.setText(str(self.sumSoFar))
    self.sumSoFar = 0.0
    self.waitingForOperand = True

```

圖 3.7: equalClicked

第四章 Python 程式語法

Python 程式語法

4.1 變數命名

Python3 變數命名規則與關鍵字

一、Python 英文變數命名規格

1. 變數必須以英文字母大寫或小寫或底線開頭
2. 變數其餘字元可以是英文大小寫字母, 數字或底線
3. 變數區分英文大小寫
4. 變數不限字元長度
5. 不可使用關鍵字當作變數名稱

二、Python3 的程式關鍵字, 使用者命名變數時, 必須避開下列保留字.

1. Python keywords: ['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']

2. 選擇好的變數名稱:

使用有意義且適當長度的變數名稱, 例如: 使用 `length` 代表長度, 不要單獨使用 `l` 或 `L`, 也不要使用 `this_is_the_length` 程式前後變數命名方式盡量一致, 例如: 使用 `rect_length` 或 `RectLength` 用底線開頭的變數通常具有特殊意義

4.2 print 函式

4.3 重複迴圈

4.4 判斷式

4.5 數列

第五章 PyQt5 簡介

說明 PyQt5 基本架構與程式開發流程

5.1 PyQt5 架構

PyQt5-GUI frame work , 圖形使用者介面軟體框架, 可以快速製做 GUI 界面程式, 是由一系列 Python 組成。超過 620 個類, 6000 和函數和方法

Qt5 原本是 C++ 語法之後用 Python 製作而成 PyQt

Qt 採用了 signal 和 slot 的概念來處理 GUI 程式中的用戶事件。PyQt 同樣支援這種方法。任何 Python 類型都可以定義 signal 和 slot, 並與 GUI 控制項的 signal 和 slot 相連線。

第六章 心得

期末報告心得

6.1 Fossil SCM

6.2 網誌心得

6.3 Github 協同倉儲

6.4 學員心得

說明各學員任務與執行過程

第七章 結論

期末報告結論

7.1 結論與建議

結論與建議內容

參考文獻

- [1] 作者名字, “標題,” 刊物名稱, vol. 4, no. 2, pp. 201–213, Jul. 1993.