

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 陳永鋈

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。A pipe character, followed by an indented block of text is treated as a literal block, in which newlines are preserved throughout the block, including the final newline.

- 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	2
第三章 Calculator 程式	3
3.1 建立對話框	3
3.2 建立按鈕	5
3.3 建立程式碼	6
第四章 Python 程式語法	9
4.1 變數命名	9
4.2 print 函式	9
4.3 重複迴圈	10
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

表 5.1	Python 網際框架比較	12
表 5.2	價目表	12

圖目錄

圖 3.1	newform	3
圖 3.2	Dialog into ui	4
圖 3.3	qtdesigner	4
圖 3.4	button	5
圖 3.5	grid	5
圖 3.6	mult	6
圖 3.7	change	6
圖 3.8	calculator	7
圖 3.9	abo	7
圖 4.1	print	10
圖 4.2	for-1	10
圖 4.3	for	10
圖 4.4	s	11

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

可攜程式系統介紹

2.1 啟動與關閉

Windows 的內容

有一張圖片：

[Kmol][[]

稱為圖 {???}。

各 md 檔案可以在 images 目錄下自訂與 md 檔案名稱相同的子目錄存放影像檔案

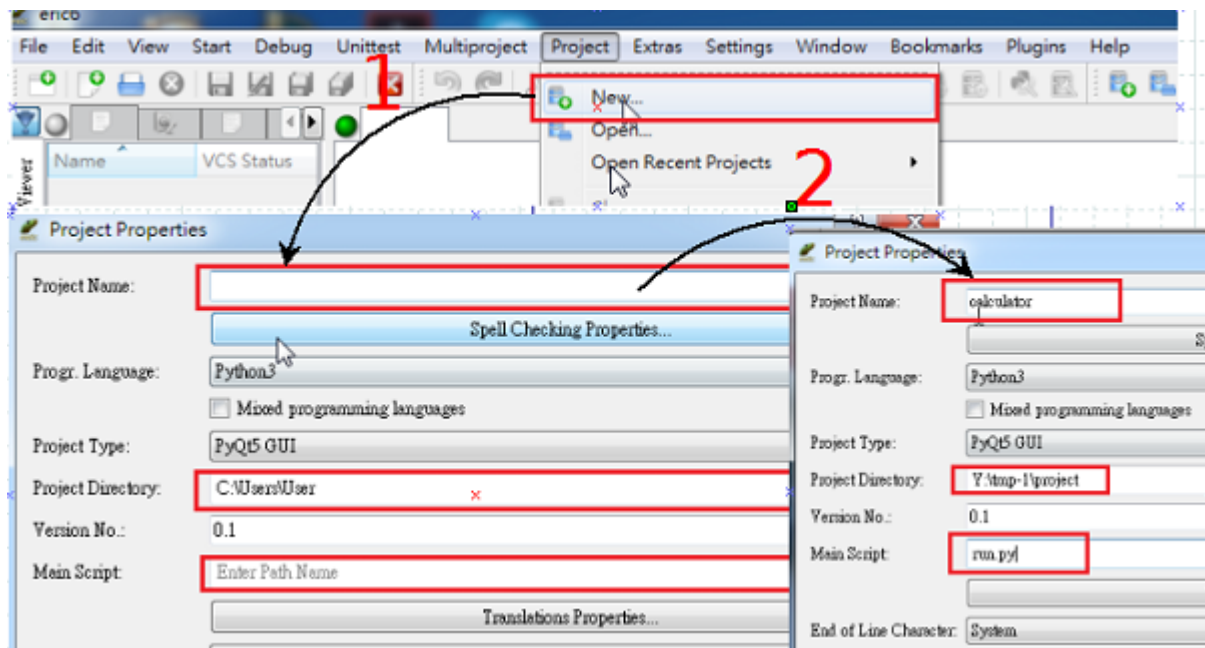
2.2 啟動與關閉 2

第三章 Calculator 程式

Calculator 程式細部說明

3.1 建立對話框

step1



step2

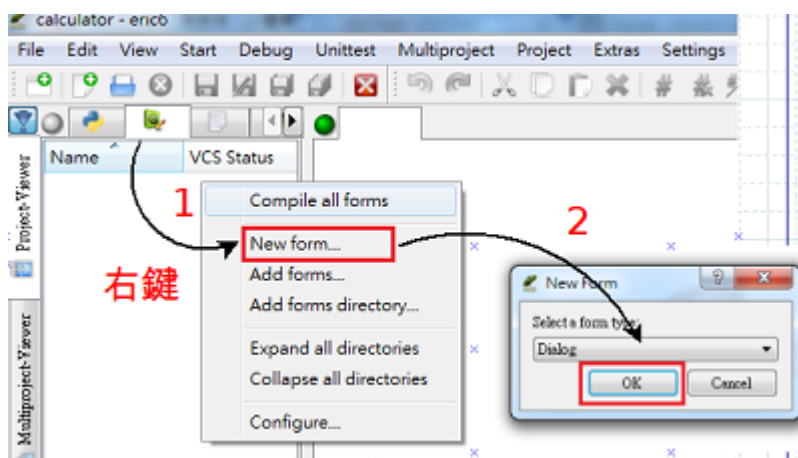
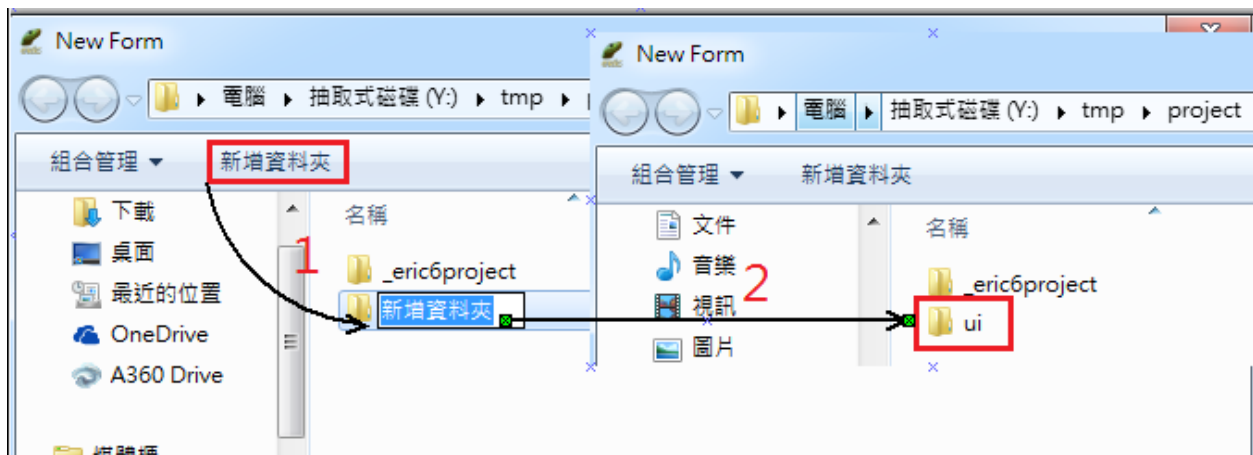


圖 3.1: newform

step3



step4



圖 3.2: Dialog into ui

step5

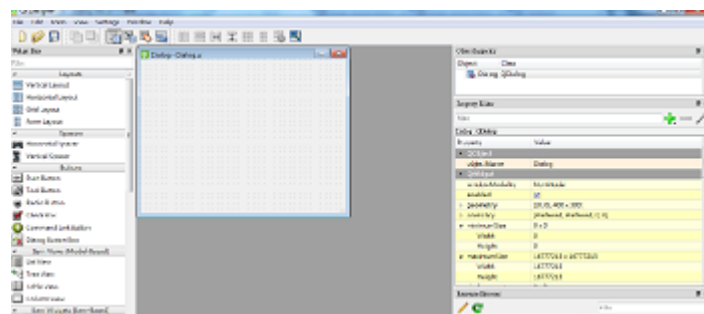


圖 3.3: qtdesigner

3.2 建立按鈕

step1

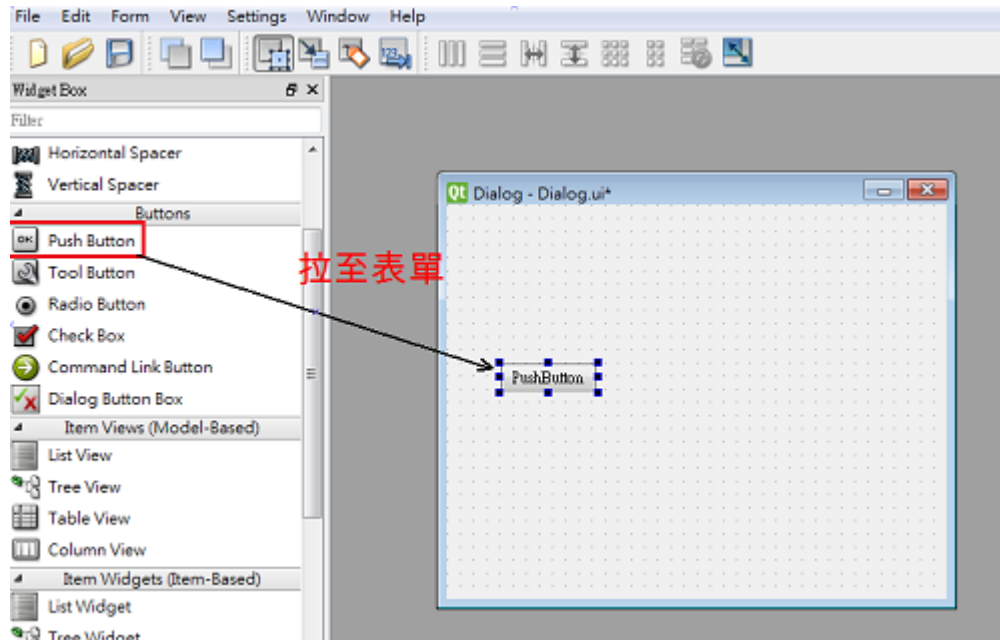


圖 3.4: button

step2

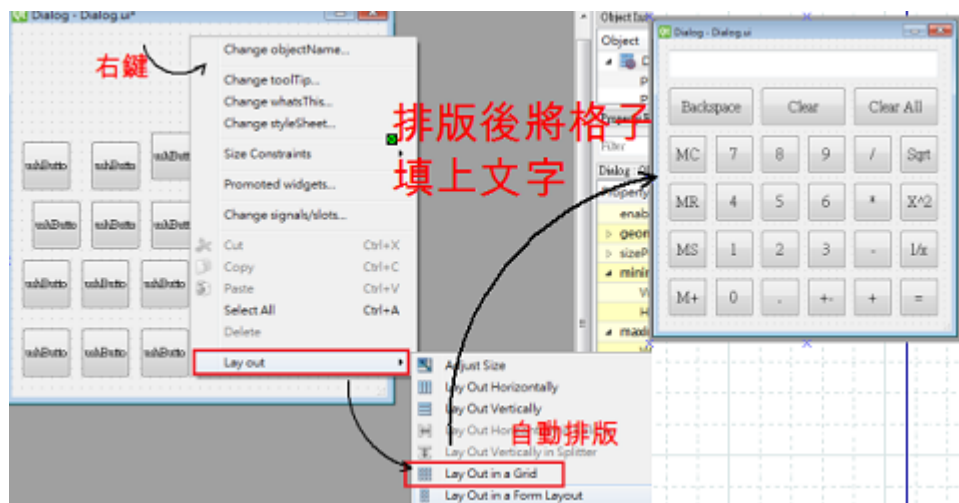


圖 3.5: grid

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

•

乘除

```
def multiplicativeOperatorClicked(self):  
    """乘或除按下後進行的處理方法"""  
    # pass  
    clickedButton = self.sender()  
    clickedOperator = clickedButton.text() 變數定義  
    operand = float(self.display.text()) 浮點數  
  
    if self.pendingMultiplicativeOperator:  
        if not self.calculate(operand, self.pendingMultiplicativeOperator):  
            self.abortOperation()  
            return 如果按的按鍵不符calculator跳至abortOperation  
    else:  
        self.display.setText(str(self.factorSoFar)) 將目前累計運算數  
        self.factorSoFar = operand 顯示在 display 上  
        self.pendingMultiplicativeOperator = clickedOperator  
        self.waitingForOperand = True  
        等待運算為真
```

圖 3.6: mult

變號

```
def changeSignClicked(self):  
    """變號鍵按下後的處理方法"""  
    # pass  
    text = self.display.text()  
    value = float(text) 浮點數  
  
    if value > 0.0:  
        text = "-" + text 如果數值小於零的話在前面  
    elif value < 0.0:  
        text = text[1:] 加-，如果大於零則為正數  
  
    self.display.setText(text)  
    數值顯示在display
```

圖 3.7: change

計算

```
def calculate(self, rightOperand, pendingOperator):  
  
    """計算""" 右運算數與等待運算子當作輸入  
    #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 進入計算時, 用目前輸入的運算數值  
                與 self.sumSoFar 執行計算
```

圖 3.8: calculator

中斷運算

```
def abortOperation(self):  
  
    """中斷運算""" 清除後, display顯示erro  
    # pass  
    self.clearAll()  
    self.display.setText("erro")
```

圖 3.9: abo

40623221

40623228

40623229

40623230

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.

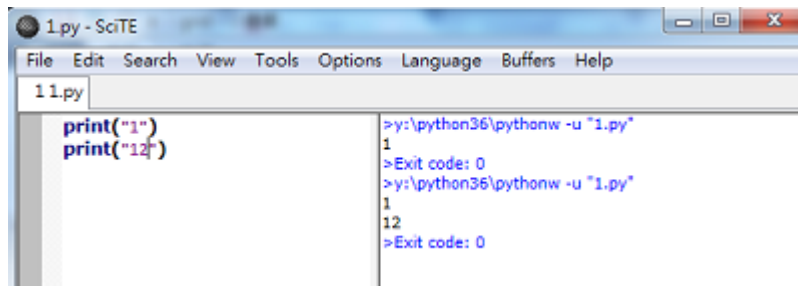
1. Python keywords: ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

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

`print()` 為 Python 程式語言中用來列印數值或字串的函式, 其中有 `sep` 變數定義分隔符號, `sep` 內定為 “,”, `end` 變數則用來定義列印結尾的符號, `end` 內定為跳行符號



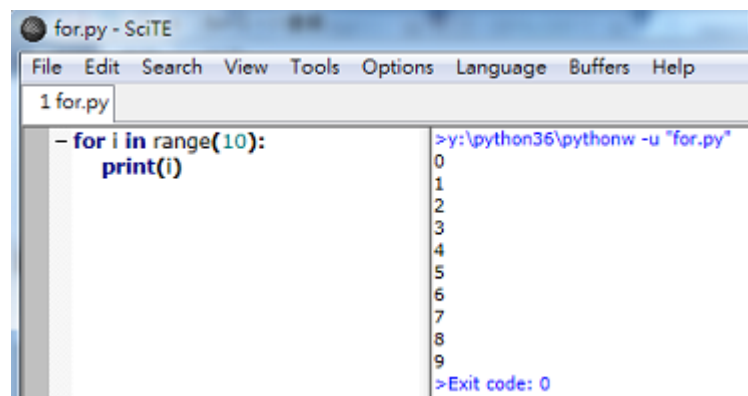
```
1.py - SciTE
File Edit Search View Tools Options Language Buffers Help
1.py
print("1")
print("12")

>y:\python36\pythonw -u "1.py"
1
>Exit code: 0
>y:\python36\pythonw -u "1.py"
1
12
>Exit code: 0
```

圖 4.1: print

4.3 重複迴圈

簡單的迴圈

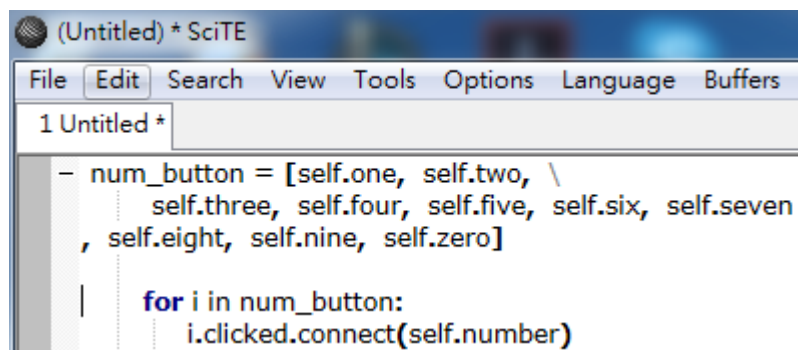


```
for.py - SciTE
File Edit Search View Tools Options Language Buffers Help
1 for.py
for i in range(10):
    print(i)

>y:\python36\pythonw -u "for.py"
0
1
2
3
4
5
6
7
8
9
>Exit code: 0
```

圖 4.2: for-1

計算機中的迴圈



```
(Untitled) * SciTE
File Edit Search View Tools Options Language Buffers
1 Untitled *
- 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.number)
```

圖 4.3: for

4.4 判斷式

if 判斷式 1:

要處理的指令1

elif 判斷式 2:

要處理的指令2

else: 要處理的指令 3

注意事項

1. 每個判斷式的結束要加:
2. 要處理的指令不可以用 {} 括起來
3. 最後的 else 可以不用加

4.5 數列

python 的數列是一個 []

[] 是容器中能放容器也能放物件和字串

容器例如: list、set、dict、tuple

計算機中的數列

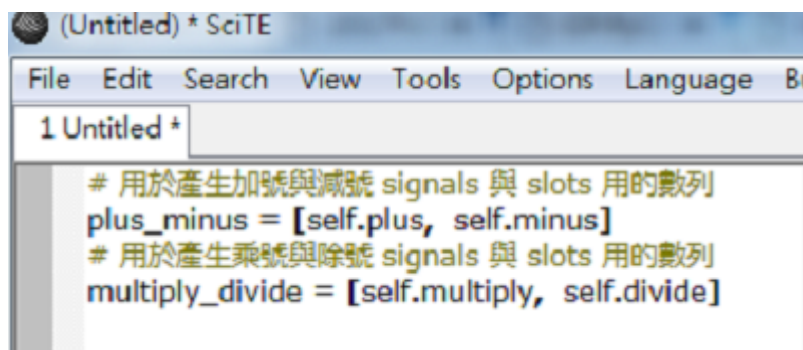


圖 4.4: s

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

C 的內容

其中包含一個表格：

表 5.1: Python 網際框架比較

Framework	Started	Py2	Py3	ORM	Template Engine	Auth Moudule	Database Admin	Project Scale
Pyramid	2005	V	V			V		large
Django	2006	V	V	V	V	V	V	large
Flask	2010	V						small

稱為表 5.1。

表 5.2: 價目表

Tables	Are	Cool
col 1 is	left-aligned	\$1600
col 2 is	centered	\$12
col 3 is	right-aligned	\$1

稱為表 5.2。

關於表格生成可以參考這裡：http://www.tablesgenerator.com/markdown_tables

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.