

# 國立虎尾科技大學

## 機械設計工程系

### 計算機程式 ag4 期末報告

#### PyQt5 事件導向計算器

#### PyQt5 Event-Driven Calculator Project

學生：

設計一甲 40623119 歐宗韋

設計一甲 40623120 蔡宗穎

設計一甲 40623121 蔡朝旭

設計一甲 40623128 張華偉

設計一甲 40623129 陳威誠 (組長)

設計一甲 40623130 陳鉅忠

指導教授：嚴家銘

2017.01.08

# 摘要

Fossil SCM

github

簡易 python 語法

網路自我診斷

免費開源軟體錄影

計算機程式語法

使用網路表單 Ethercalc 彙整所學

# 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
第三章 Python 程式語法 . . . . .	5
3.1 變數命名 . . . . .	5
3.2 print 函式 . . . . .	6
3.3 重複迴圈 . . . . .	6
3.4 判斷式 . . . . .	8
3.5 數列 . . . . .	8
第四章 PyQt5 簡介 . . . . .	9
4.1 PyQt5 優點 . . . . .	9
4.2 PyQt5 基本撰寫知識 . . . . .	9
4.3 程式開發流程 . . . . .	9
第五章 Calculator 程式 . . . . .	11
5.1 建立對話框 . . . . .	11
5.2 建立按鈕 . . . . .	12
5.3 建立程式碼 . . . . .	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.....	15
7.1	結論與建議 .....	15
第八章	參考文獻 .....	16

## 表目錄

表 5.1	Python 網際框架比較 . . . . .	11
表 5.2	價目表 . . . . .	11

## 圖目錄

圖 2.1	啟動與關閉 . . . . .	2
圖 2.2	Y 槽 . . . . .	2
圖 2.3	cmd . . . . .	2
圖 2.4	SciTE . . . . .	3
圖 2.5	把 star.bat 檔案拉進 SciTE . . . . .	4
圖 3.1	Print 簡易範例 . . . . .	6
圖 3.2	for 迴圈簡易範例 . . . . .	6
圖 3.3	while 迴圈 . . . . .	7
圖 4.1	程式開發流程圖 . . . . .	10

# 第一章 前言

## 計算器程式期末報告前言

為了將來資訊化的時代，讓電腦幫你解決問題，幫你計算，幫你節省時間，幫你處理事情，而學習計算機相關資訊，雖然連入門都還不到，但已經踏出第一隻腳了，接著只要整理所學並擴充所學，一定可以適應將來資訊化的時代。

自己寫報告覺得很累嗎? 不願擔心，有了 `github`，幫你輕鬆大家一起同時寫報告，不用擔心版本新舊，自動幫你合併整合資料，想找舊版資料嗎? 別擔心 `github` 都幫你記住了，有了 `github` 再也不用擔心豬隊友把你的報告搞得一團亂。

## 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

請參考下圖

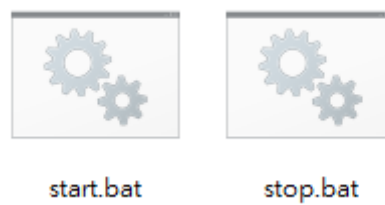


圖 2.1: 啟動與關閉

按下 start 後會產生新的磁碟:Y 槽



圖 2.2: Y 槽

並同時跳出多個 cmd 跟 SciTE



圖 2.3: cmd



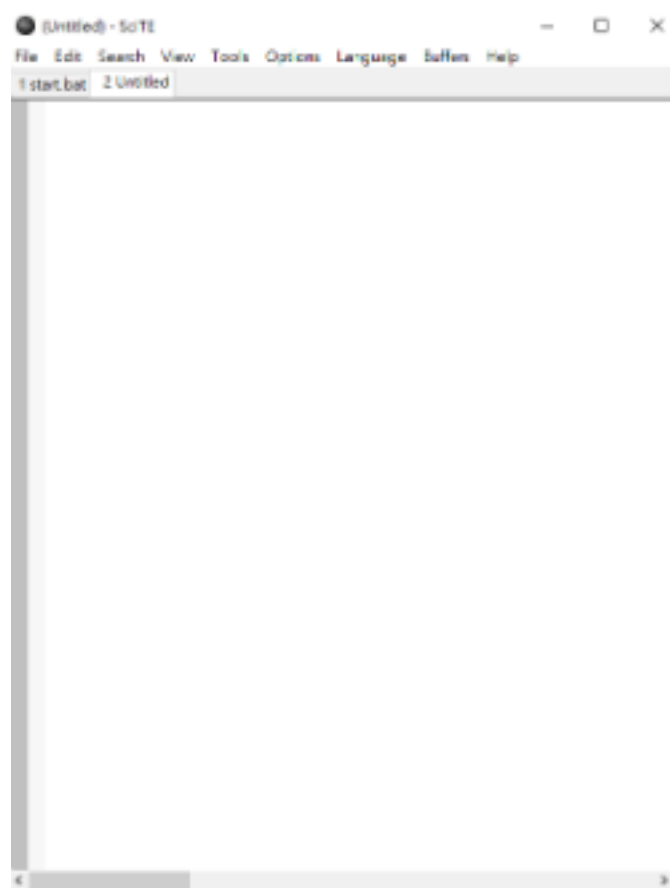


圖 2.4: SciTE

如果要設定跳出的數量只需要把 star.bat 拉進 SciTE



圖 2.5: 把 star.bat 檔案拉進 SciTE

## 第三章 Python 程式語法

### Python 程式語法

#### 3.1 變數命名

變數必須以英文字母大小寫或底線開頭

其餘字元可以是英文大小寫字母, 數字或底線

變數區分英文大小寫

變數不限字元長度

不可使用關鍵字當作變數名稱

變數命名時須避開下列字串

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

```
1 number = 10
2 Number = 20
3 _number = 30
4 _Number = 40
5 number_ = 50
6 Number_ = 60
7 print(number, Number, _number, _Number, number_, Number_)
```

Filename: .py

10 20 30 40 50 60

## 3.2 print 函式

print() 為 Python 程式語言中用來列印數值或字串的函式

列印數值只需要在括號中打入數字，列印文字則需要加上”“來包住文字，也可以用逗號隔開兩個要列印的數值或文字，它就會直接列印兩個了。

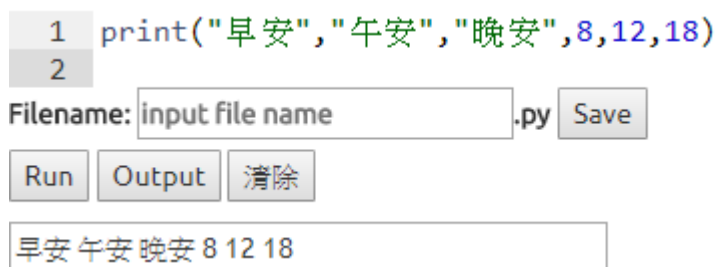


圖 3.1: Print 簡易範例

## 3.3 重複迴圈

重複迴圈分為 while 迴圈與 for 迴圈

for 迴圈:

在計算機科學中，for 迴圈（英語：for loop）是一種程式語言的疊代陳述，能夠讓程式碼反覆的執行。

它跟其他的迴圈，如 while 迴圈，最大的不同，是它擁有一個迴圈計數器，或是迴圈變數。這使得 for 迴圈能夠知道在疊代過程中的執行順序。(取自 wiki)

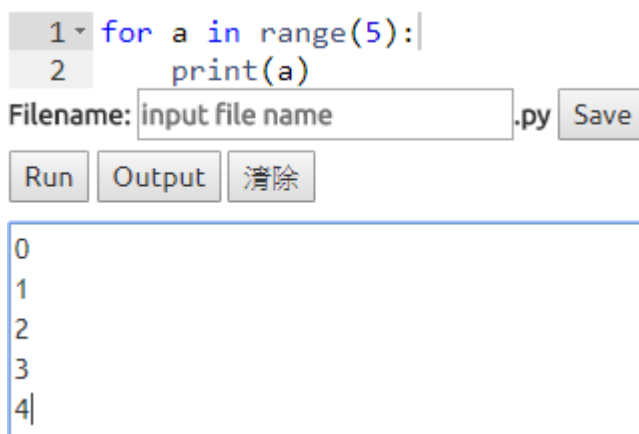


圖 3.2: for 迴圈簡易範例

while 迴圈:

while 迴圈與 for 迴圈最大的差異是 for 迴圈有固定會跑幾次，但 while 迴圈沒有特別限制會跑幾次，因此通常沒有固定跑幾次的迴圈常用 while 迴圈，while 迴圈可以搭配 continue、break、pass。

continue→ 使程式從迴圈的開始繼續執行

break→ 跳出迴圈

pass→ 什麼事都不做

```
1 i=1
2 ans=0
3 while i<=15:
4     ans+=i
5     i+=2
6 print(ans)
```

Filename:  .py

64

圖 3.3: while 迴圈

### 3.4 判斷式

判斷式就是我們製作計算機時所使用的 `def calculator` 的內容裡有寫到當達成某一

```
def calculate(self, rightOperand, pendingOperator):
    if pendingOperator == "+":
        self.sumSoFar += rightOperand

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

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

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

        self.factorSoFar /= rightOperand

    return True
```

條件時就會執行，反之則不執行

### 3.5 數列

數列有分為 `str`、`unicode`、`list`、`tuple`、`buffer`、`xrange`

`str` → 把括號內轉換成適合閱讀的形式

`unicode` → 有點像程式間的翻譯機每一種不同的語言都可以跟 `unicode` 互相轉換

`list` → 把一串你要的資料輸入中括號中，就可以幫你記憶你只有輸入相應位置的數字就會幫你輸入了

`tuple` → 跟 `list` 功能一樣，只差在不能夠修改

`buffer` → 在 `python3` 中改為 `memoryview`，有點像緩存功能

`xrange` → 用法跟 `range` 一樣，不同的是可以生成等差數列

## 第四章 PyQt5 簡介

說明 PyQt5 基本架構與程式開發流程

### 4.1 PyQt5 優點

PyQt5 有幾項特點

1. 容易撰寫

2. 功能強大

3. 跨平台

4. 容易擴充

5. 易於學習. 閱讀. 維護

### 4.2 PyQt5 基本撰寫知識

Python 程式會存成.py 檔，在 Windows 下的 Python 安裝程式會自動把這種副檔名和 Python 直譯器程式關聯起來，所以在指令行介面 (cmd.exe) 下輸入 “blahblah.py” 這樣的字眼，就會執行這個程式

### 4.3 程式開發流程

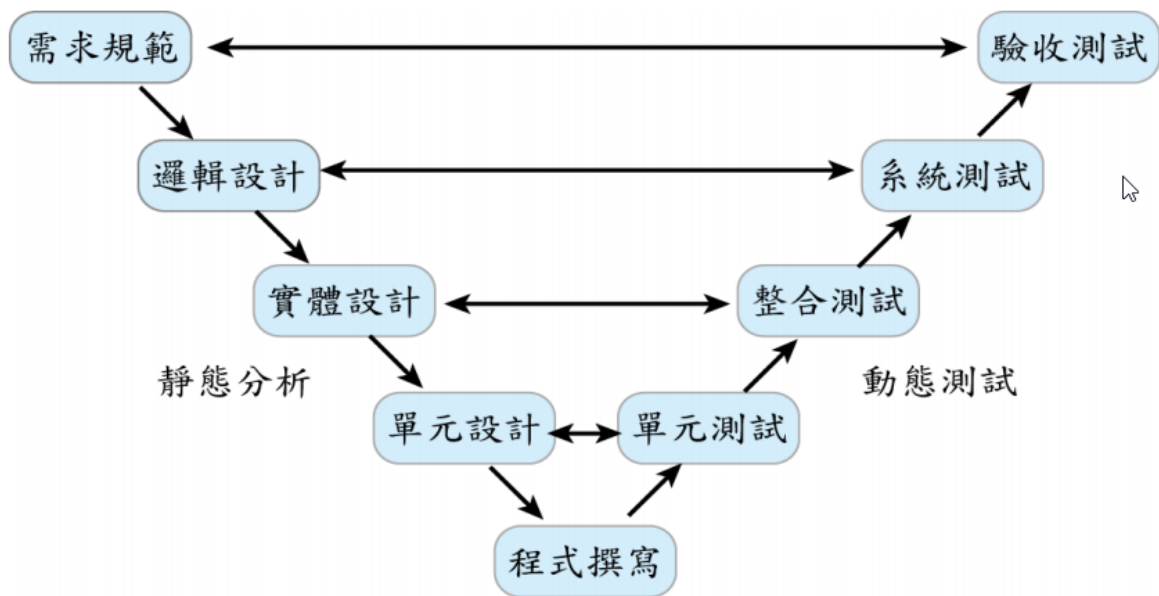


圖 4.1: 程式開發流程圖



## 第五章 Calculator 程式

Calculator 程式細部說明

### 5.1 建立對話框

自動控制的内容

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](http://www.tablesgenerator.com/markdown_tables)

## 5.2 建立按鈕

## 5.3 建立程式碼

## 第六章 心得

### 期末報告心得

#### 6.1 Fossil SCM

40623128 張華偉: fossil SCM 是我們在接觸 github 之前先接觸的網路倉儲，這讓我了解到網路倉儲的方便性，用別人電腦也可以更改倉儲資料，還可以檢視我在何時改了什麼，因此更加容易找到錯誤，只要對照之前的就知道我哪裡多改了。

#### 6.2 網誌心得

40623128 張華偉: 這應該是我人生第一次寫網誌，記錄下每一堂課上了什麼，雖然一開始使用 Markdown 編寫時很多不懂，但經過很長時間的努力，我應該可以算是小高手，知道出問題如何解決，我花了很多時間在上網尋找 Markdown 的編寫語法，雖然還有很多不足，但我會再接再厲的。

#### 6.3 Github 協同倉儲

40623128 張華偉: github，我們協同作業的開始，一開始使用 github 時，時常打成 fossil 的指令，經過了一段時間的適應後，我才不會一直打成 fossil 的指令，用 github 來協同作業真的很方便，可以看誰做了什麼，誰是豬隊友什麼都沒做，誰亂改程式碼，真的很方便，讓被抓到的人百口莫辯，上面寫的一清二楚，刪了什麼，加了什麼，一看就知道，我相信 github 可以幫我很多，上面也有很多別人協同製作的程式，可以幫助我自我學習。

#### 6.4 學員心得

40623128 張華偉: 我在製作 pdf 時我負責的任務是編輯一部分資料與教導組員如何使用 Markdown 編輯 pdf，因為有很多東西不懂，我花了一點時間上網解決問題，我發現使用 Markdown 編輯 pdf 有個很不方便的事情，圖片沒辦法調整大小，只能去調整原圖的大小，真的很麻煩，希望如果有辦法調整的話可以教教我。

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:

建議:

## 第八章 參考文獻

超連結:

淺談 Python 的 for 迴圈

Markdown 表單製作

<https://zh.wikipedia.org/wiki/Wikipedia:%E9%A6%96%E9%A1%B5>

<https://ithelp.ithome.com.tw/articles/10158587>

<https://docs.python.org/2.4/lib/typesseq.html>

<http://www.runoob.com/python/python-func-str.html>

<https://www.programiz.com/python-programming/methods/built-in/str>

<http://python.ez2learn.com/basic/unicode.html>

[PyQt5 程式簡介] [ilms.csu.edu.tw/sys/read\\_attach.php?id=814178](http://ilms.csu.edu.tw/sys/read_attach.php?id=814178)

[https://www.uuu.com.tw/Public/content/Edm/171115\\_brochure/pdf/Python.pdf](https://www.uuu.com.tw/Public/content/Edm/171115_brochure/pdf/Python.pdf)