國立虎尾科技大學

機械設計工程系

計算機程式 bg5 期末報告

PyQt5 事件導向計算器

PyQt5 Event-Driven Calculator Project

學生：

設計一乙 40623222蔡博淮 (內文-可攜程式系統介紹、Python 程式語法)

設計一乙 40623223 任明彥 (摘要 fossil、github 倉儲)

設計一乙 40623224 鐘偉哲 (內文-Calculator 程式)

設計一乙 40623231周駿麟 (內文-PyQt5 簡介)

設計一乙 40623232 余建杰 (前言-討論與建議)

設計一乙 40623233 謝宗宏 (結論、討論與建議)

指導教授：嚴家銘

2018.01.08

目錄

|  |  |
| --- | --- |
| 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　．．．．．．．．．．．．．．．．．． | 4 |
| 第三章 Python 程式語法　．．．．．．．．．．．．．．．．． | 6 |
| 3-1 變數命名　．．．．．．．．．．．．．．．．．．．． | 6 |
| 3-2 prin函式　．．．．．．．．．．．．．．．．．．．． | 7 |
| 3-3 重複迴圈　．．．．．．．．．．．．．．．．．．．． | 8 |
| 3-4 判斷式　．．．．．．．．．．．．．．．．．．．．． | 9 |
| 3-5 數列　．．．．．．．．．．．．．．．．．．．．．． | 10 |
| 第四章 倉儲系統　．．．．．．．．．．．．．．．．．．．． | 11 |
| 4-1 Fossil SCM　．．．．．．．．．．．．．．．．．．．． | 11 |
| 4-2 Github 協同　．．．．．．．．．．．．．．．．．．．． | 14 |

|  |  |
| --- | --- |
| 第五章Calculator 程式　．．．．．．．．．．．．．．．．．． | 18 |
| 5-1計算機製作　．．．．．．．．．．．．．．．．．．． | 18 |
| 5-2計算機的命名 ．．．．．．．．．．．．．． | 20 |
| 第六章 結論與建議　．．．．．．．．．．．．．．．．．．．．． | 29 |

第一章 前言

本次報告的目的在於統整這學期的課程內容並加以檢視及複習，參照上課的日期依序介紹各教學內容:可攜程式系統介紹、Python程式語法、fossil網誌、github倉儲、PyQt5 簡介、Calculator 程式。為多加練習多人協同合作之形式，本報告由各組學員分段同時編寫，以利未來行事之效率。根據結果，多人協同有助於減少個人作業的壓力並能與協同者有較多的交流，有利於討論及改進錯誤，使時間能在有效的運用下縮短整體作業時間。最後，歸納報告之結論，並整合各學員討論及建議，做為後續相關課程的參考資料。

**關鍵詞：**檢視及複習、多人協同、行事之效率、討論及改進錯誤。

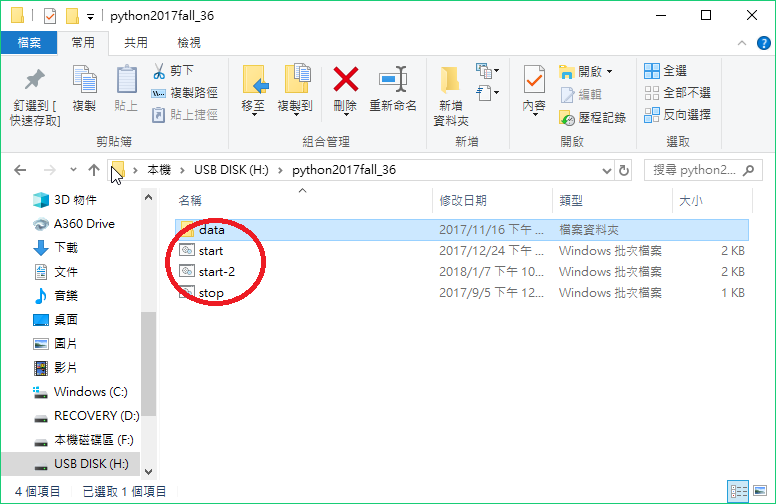
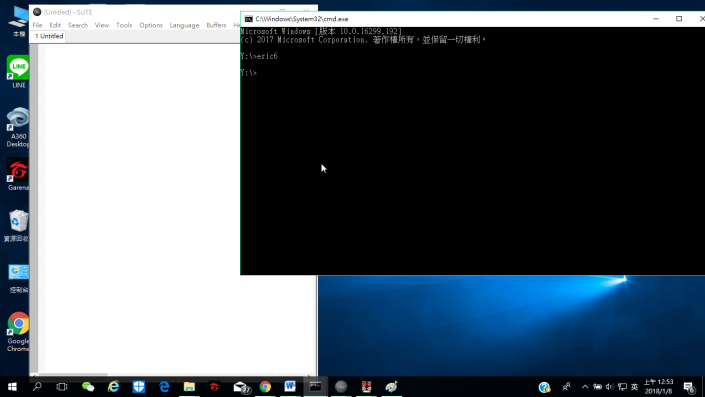
1

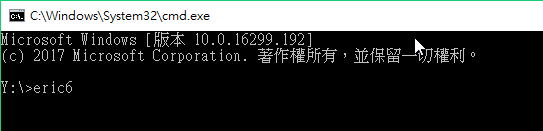
第二章 可攜程式系統介紹

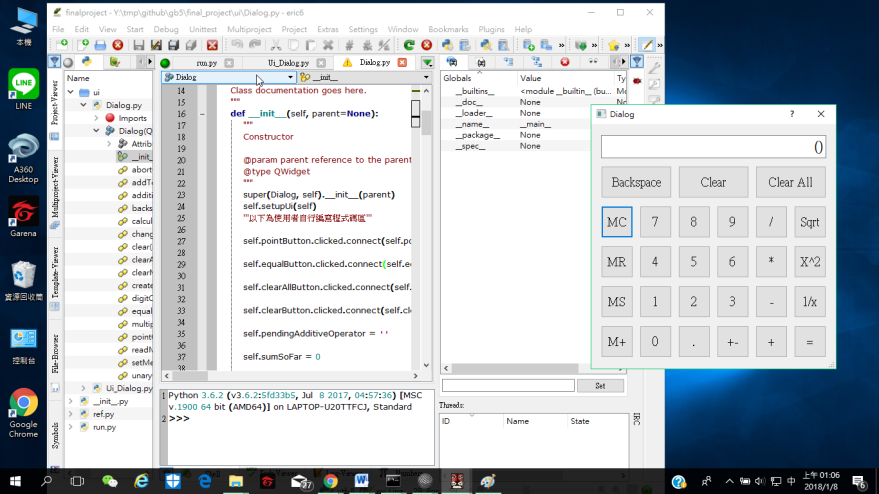
2-1 啟動與關閉

1.啟動

按下在python2017fall\_36裡的star.bat，使用它來開啟小黑盒及SciTE，來繼續後續的工作處理。



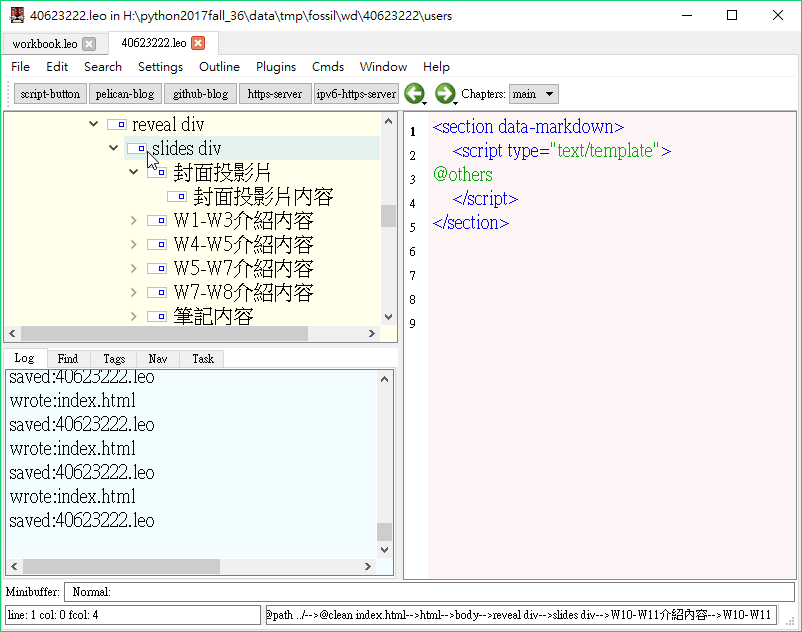
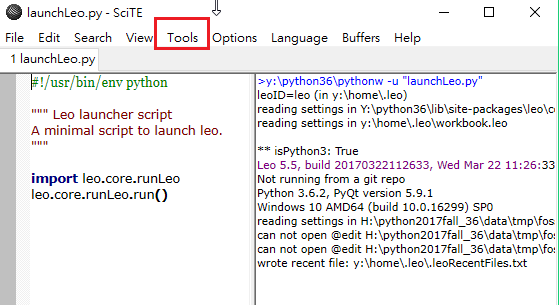
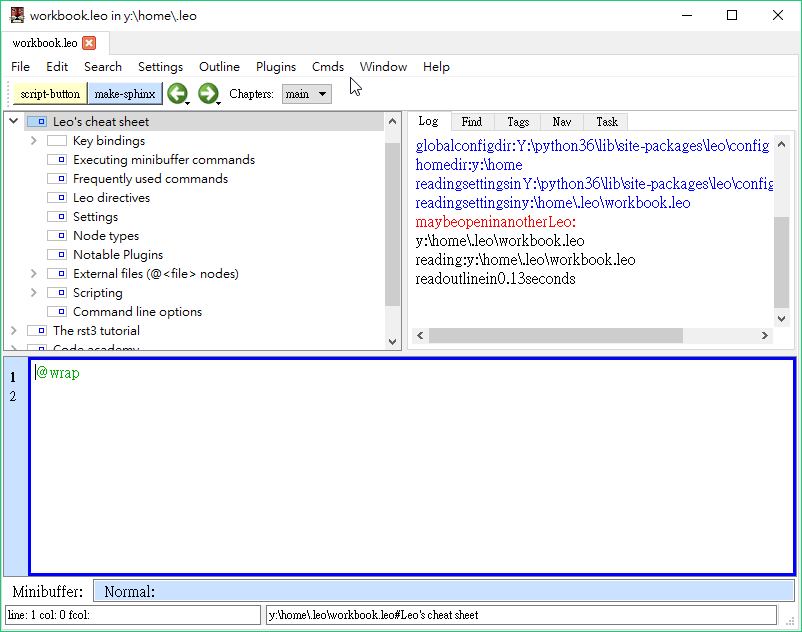
(1) 利用小黑盒叫出eric6來做計算機程式的編寫



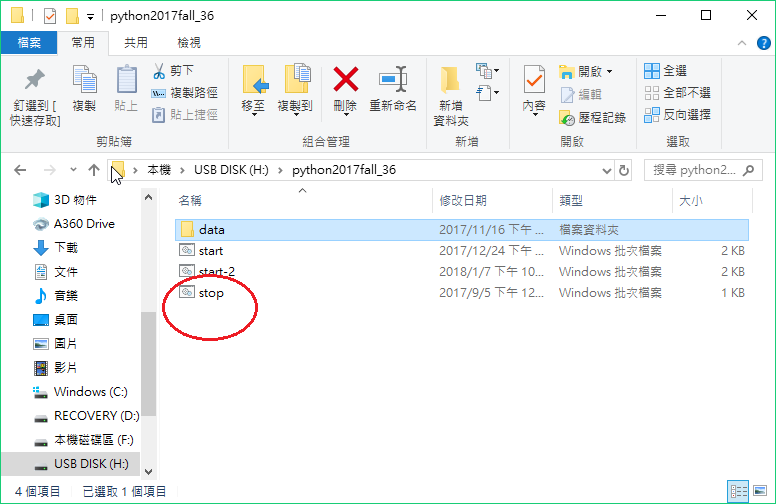
2

(2) 利用SciTE來開啟launchLeo，但開啟時需再把自己的檔案打開。

按下tools來驅動



打開自己的leo檔案

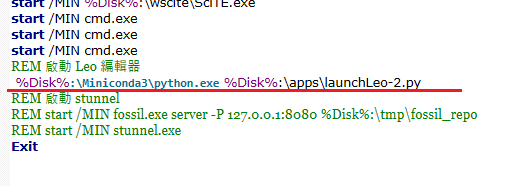
(3) 關閉時按下python2017fall\_36的stop鍵，停止所有用python來開的程式。

3

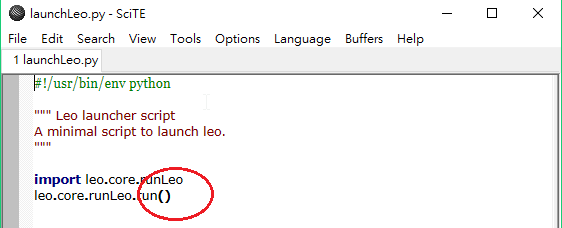
2-2 啟動與關閉-2

1.為了現代客製化的趨勢，我們把開啟時的步驟縮減一些，並且能達到快速的使用。而縮減步驟的部分為launchLeo的開啟。

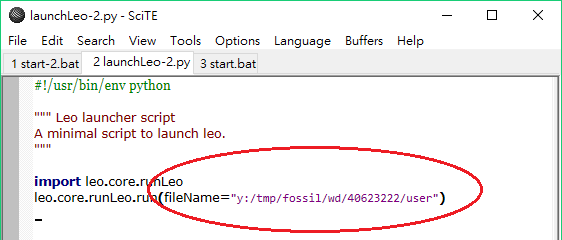




2.要讓程式知道我們要開的是哪個檔案，所以必須要在launchLeo.py中改寫開的檔案。

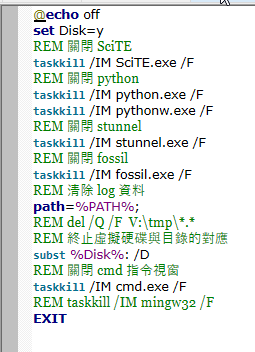


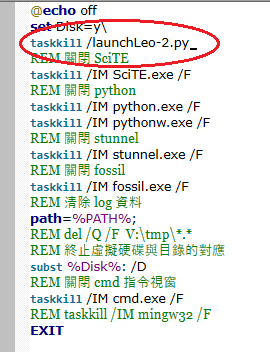
把節點設定在自己的檔名



4

3.關閉時為了要將所有的程式關閉，所以也必須改寫。





5

第三章 Python 程式語法

3-1命名

1. 通用命名規則:

函式命名，變數命名，文件命名要有描述性；少用縮寫，盡可能地使用描述性的字元來呈現，方便其他人來觀看，以及易於檢查。

2. 文件命名:

文件名要全部小寫，可以包含底線 (\_) 或連字符 (-)。如果是有專案約定的情況，則照專案約定來使用，如果沒有專案約定(­­­\_)會更好。

3. 變數命名:

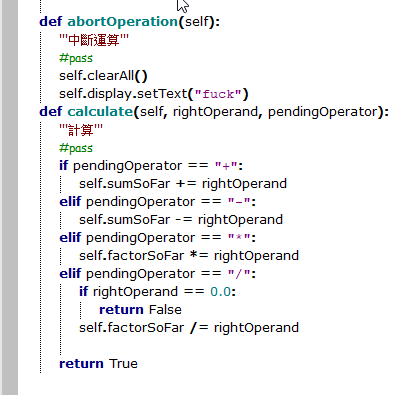
變數名一律小寫，單詞之間用底線連接。類別的成員變量以下劃線結尾，但結構體的就不用

6

3-2 print函式

函式為結構化程式，將相同功能的程式獨立出來，經由函式的呼叫，傳入資料與回傳處理後的結果，只要將函式寫好，可以不斷利用此函式做相同動作，可以達成程式碼不重複，要修改此功能，只要更改此函式。

函式的定義:

 以def開頭，空一個空白字元，接函式名稱後，串接著一對小括號，小括號可以填入要傳入函式的參數，當參數有多個的時候以逗號隔開，右小括號後面須接上「:」，函式範圍為縮行固定個數空白字元的程式碼，縮行相同空白字元的程式碼就是函式的作用範圍。當函式需要傳回值使用指令return，表示函式回傳資料給原呼叫函式，若不需要回傳值的函式就不需要加上return。

7

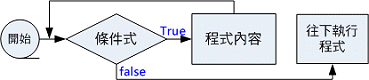
3-3 重複迴圈

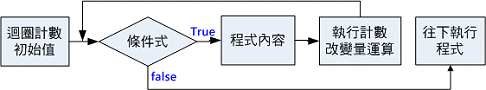
1.處理重複事件的敘述稱為迴圈，有[while]、[for]、[do…while]。

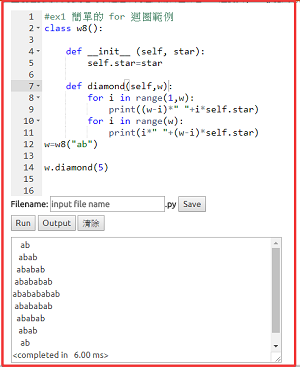
2.[for]是屬於固定次數的迴圈，[while]為不固定次數的迴圈。

3.[for]、[while]為測試在執行，屬於前測試迴圈；[do…while]為執行後在測試，屬於後測試迴圈。

While迴圈:

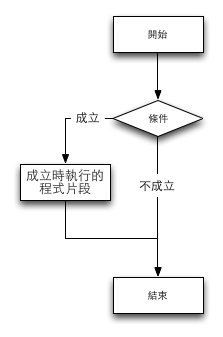


For迴圈:



8

3-4 判斷式

 １.定義:在程式中判別是否成立要或不成立的一種語法。

2.判斷的條件:

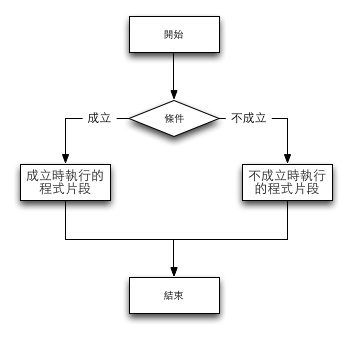
|  |  |
| --- | --- |
| 比較運算  (A、B比較) | 語法 |
| 等於 | A==B |
| 不等於 | A!=B |
| 大於 | A>B |
| 小於 | A<B |
| 不大於 | A<=B |
| 不小於 | A>=B |

9

3. if-else的使用

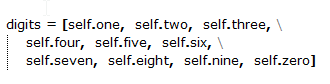
If的條件:成立時通過成立時要跑的程式。

Else的條件:不成立時通過不成立要跑的程式。



3-5. 數列

簡單的解釋就是說我把我所需要的資源全部放在一塊，然後讓我在後續的工作不需要再去一個一個找，而是直接重我所放資源的地方找。



[ ]中的部分就是數列

10

第四章 倉儲系統

4-1 Fossil SCM

Fossil scm(化石)是本學期為使學員熟悉管理版次及編寫blog所練習之程式。

以下為練習之過程:

1. 在cmd(提示指令字元)的fossil目錄下clone線上倉儲網址，並在https://後輸入綁定帳號之身分以利於之後推送。



11

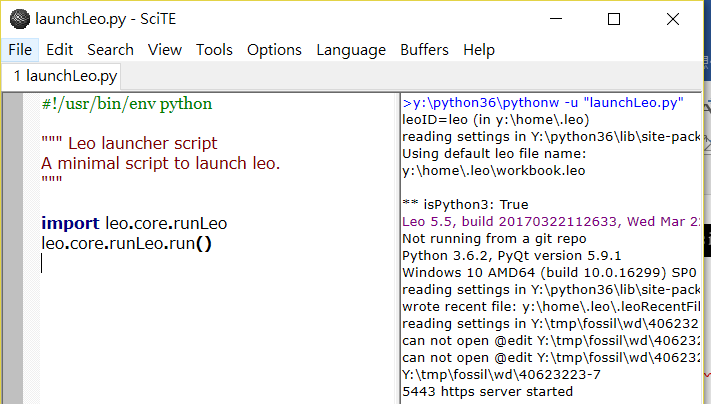
2. clone完後興建一個wd(工作目錄)以方便管理資料。



3. 在wd下興建一個自行設定的子目錄，並於子目錄open相對於fossil目錄的封包檔。

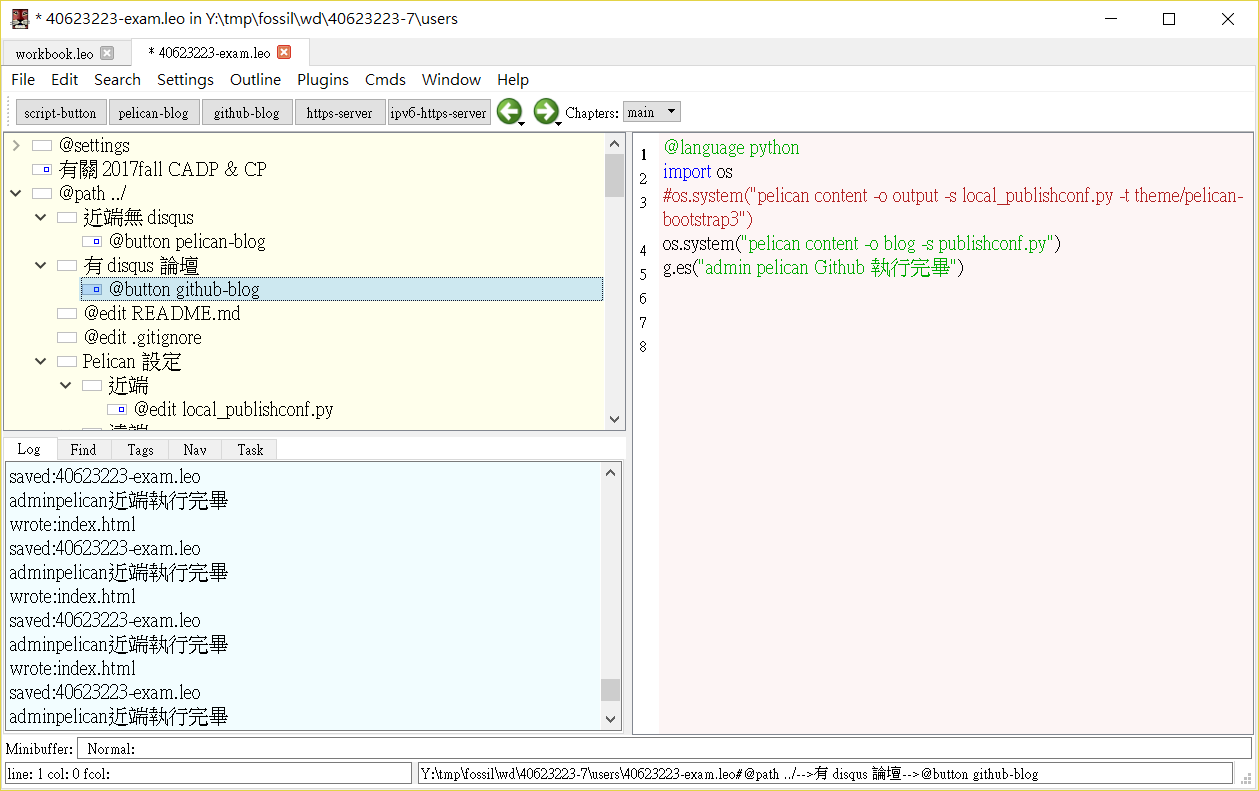


4. 在sciTE中開啟於Y:目錄下的launchLeo 啟動leo 。

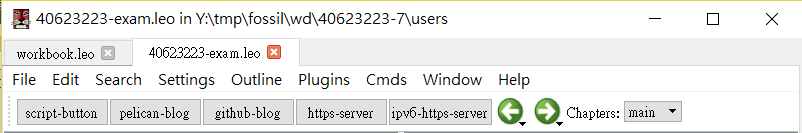


12

5. 開啟位於user的 .leo檔以編寫blog

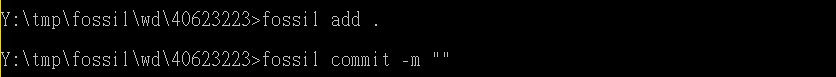


6. 利用https-server按鈕啟動近端檢視，並使用peliccan-blog按鈕將所編寫之文字轉換成近端程式碼，檢查無誤後再用github-sever按鈕轉換成遠端格式



13

14

7. 最後再用fossil add .及fossil commit -m “ ”指令將完成編寫之blog推送。

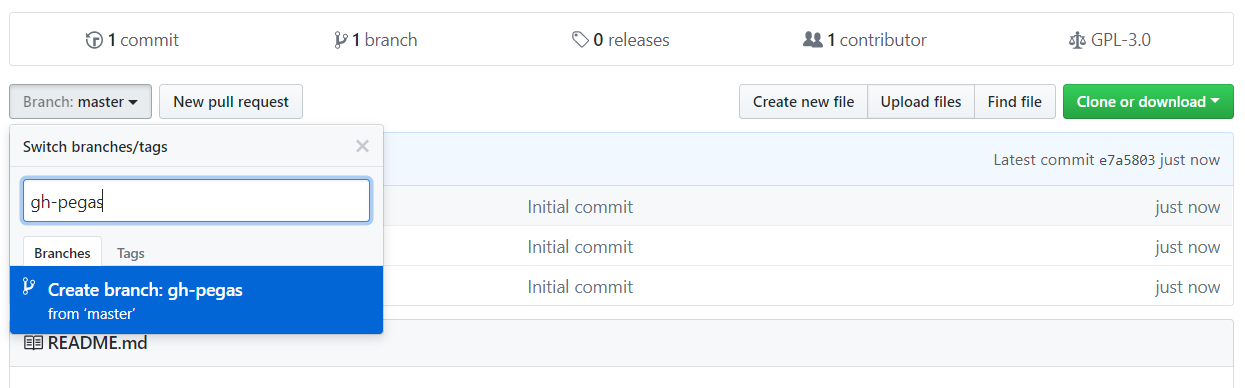
4-2 Github協同

Github為現在業界普遍的程式協同倉儲，課程中為學員練習協同計算機程式與報告，幫助各組同學熟悉未來開發程式或課程會遇到的作業方式。

以下為練習過程:

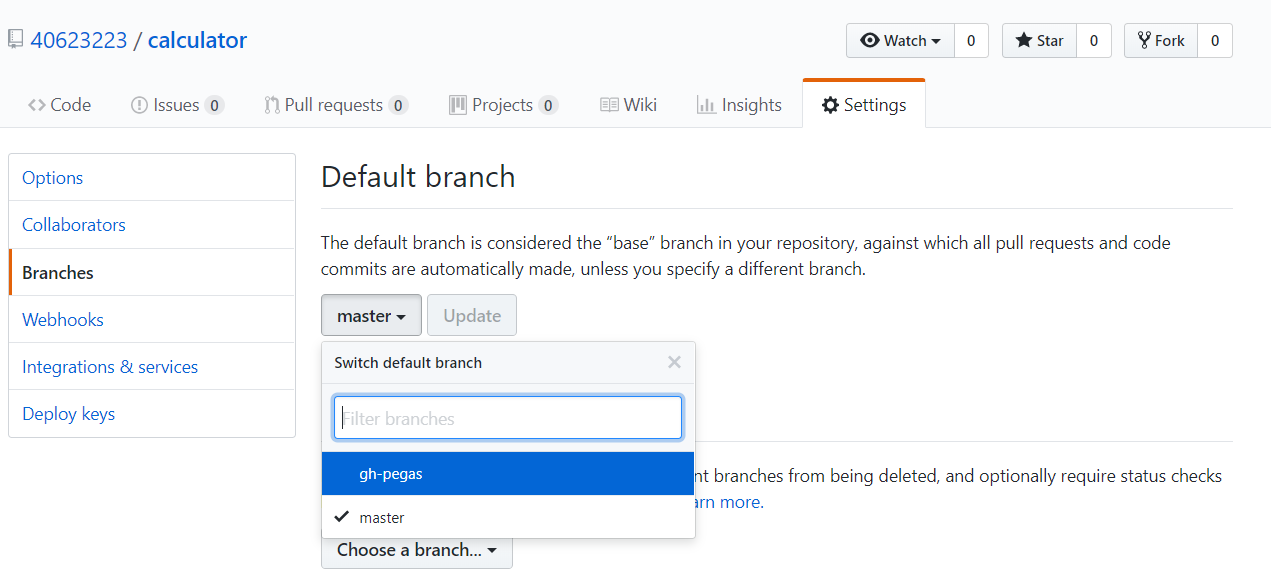
1. 建立一個新的專案並參照格式設定

2. 於Branch分支中新增gh-pegas



15

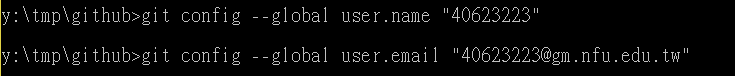
3. 進入Settings中的Branches將master修改成gh-pegas



4. 進入Collaborators邀請協同之學員

16

5. 完成分配之作業如要推送需先設定帳號email

****

6. 依序使用以下指令確認無誤後方可完成推送

****

****

****

17

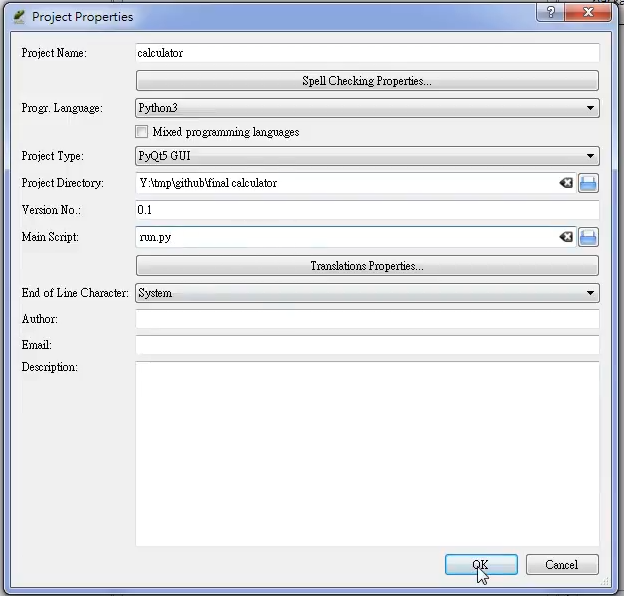
第五章 Calculator

5-1 計算機的製作

1. 先開啟eric6



2. 新增一個專案，完成後確認

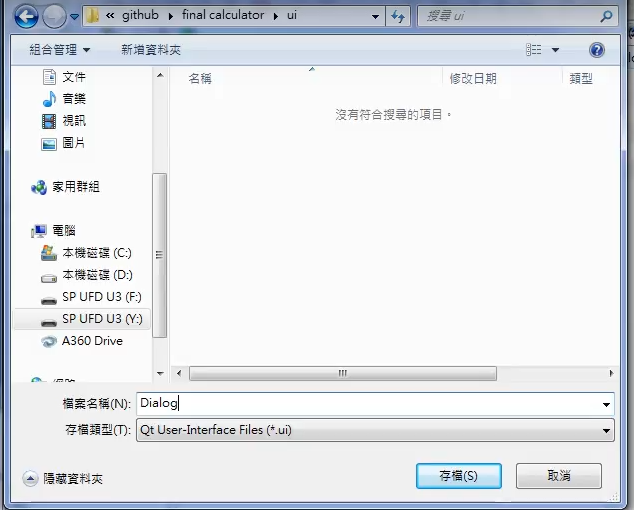
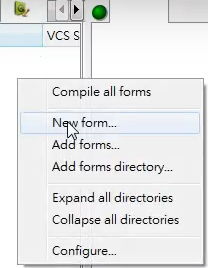
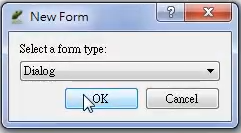


給予一個專案名稱

選取到對應的資料夾

主要跑的程式

18

3.建立一個表單

建立ui資料夾

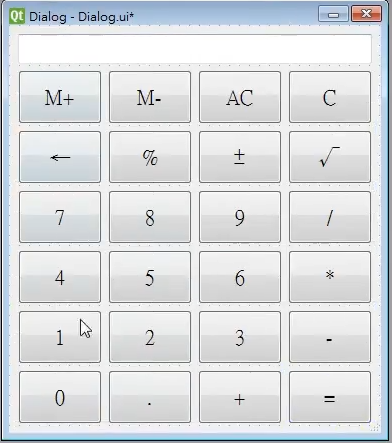
並存於底下

4. 用C:\Users\ADER\Desktop\未命名.png以及C:\Users\ADER\Desktop\未命名.png完成計算機表單並修改主詞和圖片大小

C:\Users\ADER\Desktop\未命名.png

修改主詞

表單名稱為Dialog

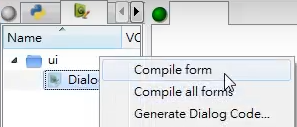


按鈕的長寬尺寸變更

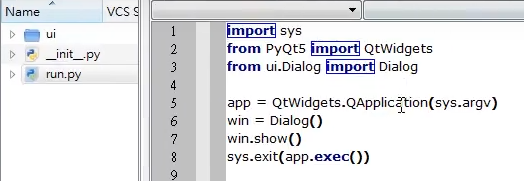
C:\Users\ADER\Desktop\未命名.png

19

5. 完成後把圖片的東西編寫上去



6. 在run.py的部分把程式撰寫上去



5-2 計算機的命名

1. 數字按鍵以zero、one、two、three、four、five、 six、seven、eight、nine命名、顯示幕以display命名、等號以 equalButton命名。

20

5-3 計算機邏輯

1. 與 MS, M+, 或 MC 按鍵相關的計算機記憶體數值, 存入 sumInMemory 變數對應的記憶空間，以 sumSoFar 儲存累計數值，使用者按下等號後， sumSoFar 重新計算結果，並顯示在 display 幕，Clear All 按鍵則重置 sumSoFar 為 0。

21

以 factorSoFar 儲存乘或除運算子運算過程所得的暫存數值； pendingAdditiveOperator 儲存使用者最後點按的加或減運算子字串；pendingMultiplicativeOperator 儲存使用者最後點按的乘或除運算子字串；waitingForOperand 界定使用者是否處理運算數輸入階段，若 waitingForOperand 為 True，表示計算機正在等待使用者"開始"輸入運算數，waitingForOperand 起始值為 True，只有重新進入 digitClicked 方法，display 才會 clear()，否則在顯

示幕非為 0 的情況下，將堆疊數字字串。

因為考量先乘除後加減運算順序，將加減與乘除處理方法分開，若使用者輸入加減運算子後，緊接乘除運算子，計算機會先乘除運算後再加減；直接運算元可以在使用者按下按鍵後, 直接對 display 中的數值進行處理，無需其他暫存需求；pendingAdditiveOperator、 pendingMultiplicativeOperator、 sumSoFar、factorSoFar 與 waitingForOperand，在 Dialog 類別建構子中設定起始值。

2.數字按鍵點按處理:

使用者點按數字按鍵，將會送出該按鍵clicked() 訊號**，**按鍵的 clicked() 訊號將會根據設定， 觸發 digitClicked() 方法槽，由於 PyQt5 的 PushButton 以 Qt5 中的 QObject::sender() 送出訊號，此函式會傳回 sender 作為 QObject 的指標，因為此一與 Push Button

22

配合的 sender 為 Button 物件，因此可以在 digitClicked() 函式中，利sender().text() 取得按鍵的 text 字串，假如使用者點按 0，display 顯示字串 0，但是若一開始輸入兩個以上的 0，digitClicked() 應該仍只顯示 0 字串，但是若計算機處於等待新運算數輸入時 (以 waitingForOperand 判定) ，新數字在顯示前，display 應該要清除先前所顯示的數字，最後，除了在顯示幕為 0 之後的 0 按鍵輸入，digitClicked() 方法槽

不會繼續判定是否清除顯示幕或堆疊數字字串外，所按的數字將會堆疊顯示。

3. 直接運算按鍵處理:

Sqrt, x^2 與 1/x 等按鍵的處理方法為unaryOperatorClicked()，與數字按鍵的點按回應相同，透過 sender().text() 取得按鍵上的 text 字串；unaryOperatorClicked() 方法隨後根據 text 判定運算

23

子後，利用 display 上的運算數進行運算後, 再將結果

顯示在display 顯示幕，若進行運算 Sqrt 求數值的平方根時，顯示幕中為負值，或 1/x 運算時， x 為 0， 都視為無法處理的情況，以呼叫 abortOperation() 處理；abortOperation() 方法則重置所有起始變數，並在 display 中顯示 "####"，直接運算子處理結束前，運算結果會顯示在 display 中，而且運算至此告一段落，計算機狀態應該要回復到等待新運算數的階段，因此 waitingForOperand 要重置為 True。

4.加或減按鍵處理:

使用者按下加或減運算子按鍵時，程式設定以 additiveOperatorClicked() 處理，進入 additiveOperatorClicked() 後，必須先查是否有尚未運算的乘或除運算子，因為必須先乘除後才能加減，先處理乘與除運算後，再處理加或減運算後，將 sumSoFar 顯示

24

在 display 後，必須重置 sumSoFar 為 0，表示運算告一段落。

5. 乘或除按鍵處理:

使用者按下乘或除運算子按鍵時，程式設定以 multiplicativeOperatorClicked() 處理，進入 multiplicativeOperatorClicked() 後，無需檢查是否有尚未運算的加或減運算子，因為乘除運算有優先權，先處理乘與除運算後，再處理加或減運算，將 sumSoFar 顯示在 display 後，必須重置 sumSoFar 為 0，表示運算告一段落。

6. 小數點按鍵處理:

使用者按下小數點按鍵後，以 pointClicked() 方法處理，直接在 display 字串中加上 "." 字串。

25

7. 數值變號按鍵處理:

使用者按下變號按鍵後，由 changeSignClicked() 處理，若顯示幕上為正值，則在 display 字串最前面, 疊上 "-" 字串，假如顯示幕上為負值，則設法移除 display 上字串最前方的 "-" 字元。

26

8. 退格按鍵處理:

使用者按下退格按鍵後，由 backspaceClicked() 處理，這時可以利用 Python 字串數列中的 [:-1] ，保留除了最後一個字元的字串，離開 backspaceClicked() 前，將顯示幕中原有字串的 [:-1] 字串，顯示在 display 上，若退格後 display 上為空字串，則顯示 0，並且將 waitingForOperand 起始設為 True，表示等待新運算數中。

9. 清除按鍵處理:

使用者按下 Clear 按鍵後，以 clear() 方法處理，進入函式後，將現有的運算數重置為 0，離開 clear() 前，將 waitingForOperand 起始設為 True，表示等待新運算數中，ClearAll 按鍵，則將所有變數全部重置為起始狀態。

10. 記憶體按鍵處理:

clearMemory() 方法與 "MC" 按鍵對應，清除記憶體中所存 sumInMemory 設為 0；readMemory() 方法與 "MR" 按鍵對應，功能為讀取記憶體中的數值，因此將 sumInMemory 顯示在 display，作為運算數；setMemory() 方法則與 "MS" 按鍵對應，功能為設定記憶體中的數值，因此取 display 中的數字，存入 sumInMemory；addToMemory() 方法與 "M+" 按鍵對應，功能為加上記憶體中的數值，因此將 sumInMemory 加上 display 中的數

27

值，因為 setMemory() 與 addToMemory() 方法，都需要取用 display 上的數值，因此必須先呼叫

equalClicked()，以更新 sumSoFar 與 display 上的數值。

11. calculate() 方法:

calculate() 方法中的運算，以 rightOperand 為右運算數當執行加或減運算時，左運算數為 sumSoFar，當執行乘或除運算時，左運算數為 factorSoFar，若運算過程出現除與 0 時，將會回傳 False。

**完成後，計算機就能使用了~**

28

第六章 結論與建議

有了這次的協同設計練習，讓我們體會到協同設計的重要性，在職場上不可能所有的東西都是你一個人自己做，一定是協同設計，每個人都負責不同的地方一點一點的拼湊出來的，至於為甚麼要用協同設計，主要的原因是因為人不可能都不休息和有問題都自己解決，有了這個協同設計當你在休息的時候換別人依序你所做的，有問題也可以提出來一起討論，以利提升行事的效率。

29