

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 40623201 王君庭

設計一乙 40623202 吳姍蓉

設計一乙 40623203 蔡宜芳

設計一乙 40623210 翁嘉宏

設計一乙 40623211 王得榮

設計一乙 40623212 魏有泉

指導教授：嚴家銘

2018.01.08

摘要

何謂機械設計?

所謂機械設計是透過詳細思慮考量後完成，其中包括固體、流體、軟體等互動元件間交互作用下的產物，使其展現特定功能的具體規劃表達。

工程師有那六種表達方式? 口語、文字、2D、3D、數學理論與實作表達。

創造力的三個基本要素：自學力、執行力與想像力。

自學力在提升自我解題能力；執行力則在運用所學解決問題，當累積一定經驗後，就能透過想像、自學與執行過程之整合運用，創造出具效能的創新產品。

本課程主要敘述做為一位機械設計工程師所需具備的基本能力，例如上面提到的六種表達方式；口語:能快速了解顧客需求並有效的溝通。文字:進行協同時能讓協同者了解專案內容。2D、3D:利用圖表、立體圖等幫助他人了解含意。數學理論:一位專業的機械設計師必須具備應有的數學能力，因機構相互作動時牽扯到許多數學理論。

有效的版次管理也是這門課程中有提到的，利用 Github 及 Fossil SCM 的功能來輔助，學期末更使用 Github 來與組員協同，近端、遠端要隨時同步更新，避免兩者版次差距過大。

目錄

摘要	i
目錄	ii
圖目錄	iv
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 Python 程式語法	3
3.1 變數命名	3
3.2 print 函式	3
3.3 重複迴圈	3
3.4 判斷式	3
3.5 數列	3
第四章 PyQt5 簡介	5
4.1 PyQt5 架構	5
第五章 Calculator 程式	6
5.1 建立對話框	6
5.2 建立按鈕	7
5.3 建立程式碼	7
第六章 心得.....	9
6.1 Github 協同倉儲	10
6.2 學員心得	10
第七章 結論.....	12
7.1 結論與建議	12
第八章 參考文獻	13

圖目錄	圖 2.1 Kmol.....	錯誤! 尚未定義書籤。
圖 3.1 Kmol		2
圖 3.2 for 迴圈.....		9
圖 5.1 建立對話框 1.....		12
圖 5.2 建立對話框 2.....		錯誤! 尚未定義書籤。
圖 5.3 建立對話框 3.....		錯誤! 尚未定義書籤。

第一章 前言

計算器程式期末報告前言:

電腦輔助設計 (Computer Aided Design)，係指運用電腦功能及特性協助使用者完成設計。

輔助係指輔佐、非主要的，意旨電腦只是相持物，而非設計主要核心，操作電腦的使用者才是主體。然而，電腦輔助設計中，並非要使用特定軟硬體才能完成工作，面對各種實際情況的考驗，使用對應的工具，甚至自行打造工具完成設計。

設計是一種表達運用: 口語、文字、2D、3D、數學、實體等表達方法交互運用下所完成的可交付內容，現今所謂機械設計中的互動元件泛指: 固體、流體與軟體元件。

本學期計算機程式是由 python3 與 PyQt5 建立簡易的計算機，其中使用基本 python3 程式語法 python3 物件導向和視窗事件驅動程式的基本概念。使用可攜隨身系統中的基本視窗命令提示指令元 (Command Prompt Commands) 編輯近端檔案及提交至遠端倉儲，第十四週分組使用 github 協同倉儲製作簡易計算機。

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



圖 2.1 : 打開後會叫出 SciTE 檔和 cmd 檔

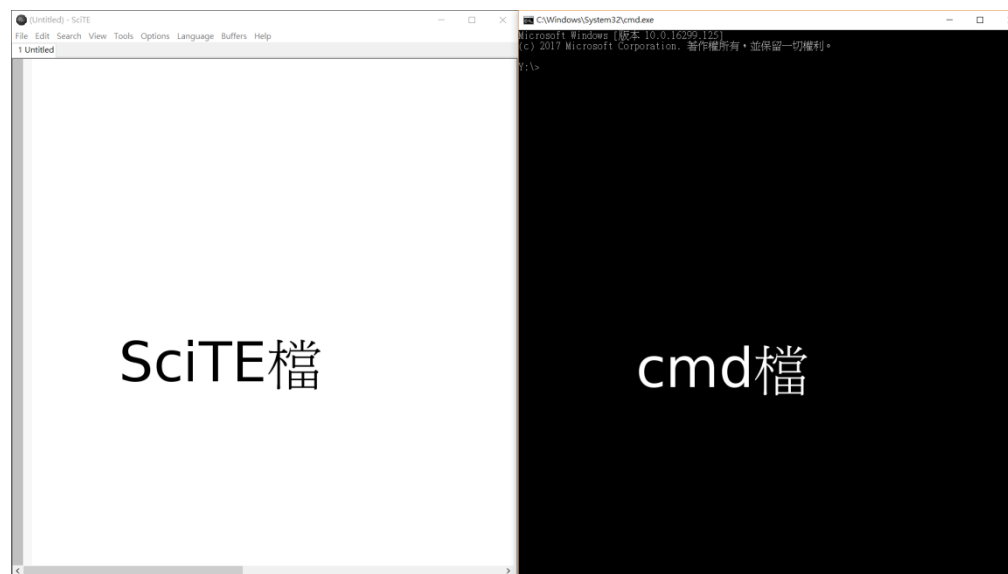


圖 2.2 : 接著進行客製化讓開啟隨身系統可以同時開啟 Leo 檔



圖 2.3 : 首先更改 start.bat 的執行碼



圖 2.4 : 一樣使用 SciTE 檔開啟

2.2 啟動與關閉 2

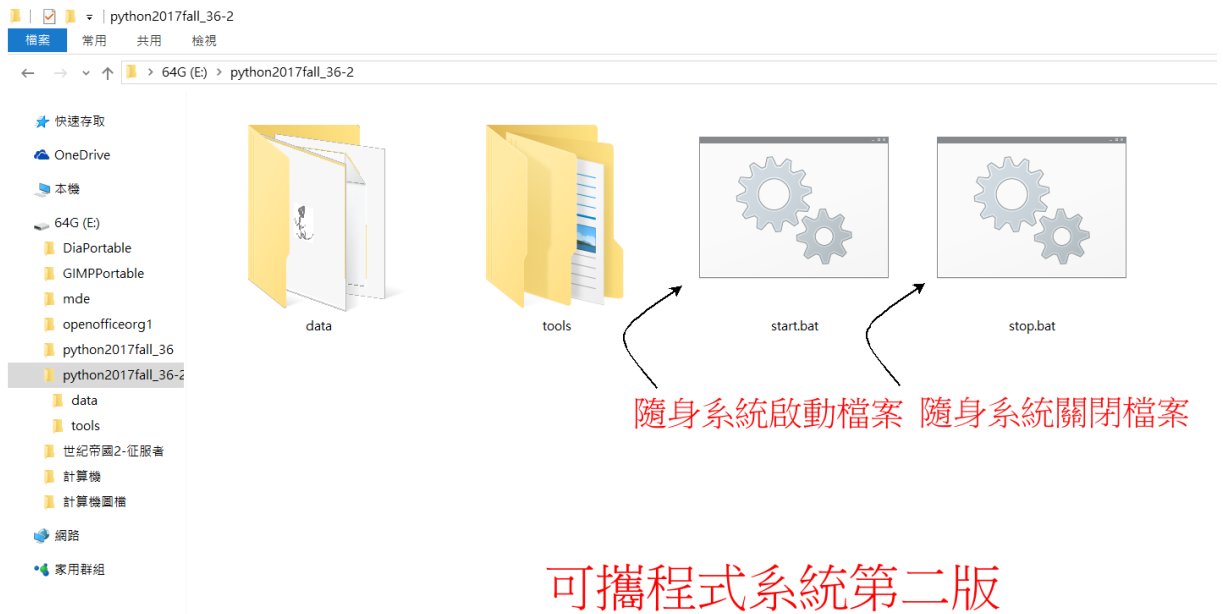


圖 2.7 : 步驟同上一樣進行客製化處理即可

第二版與第一版之差異：



圖 2.8 : 第二版多內包含了 Pandoc 與 MiKTeX 能進行倉儲上的報告協同

第三章 Python 程式語法

Python 程式語法

3.1 變數命名

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

Python3 的程式關鍵字, 使用者命名變數時, 必須避開下列保留字.

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']

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

3.2 print 函式

print() 函式用法

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

3.3 重複迴圈

For 語句是使用最為靈活的

- (1) 先執行運算式
- (2) 在執行運算式 2，若其值為真(非 0)，則執行 for 語句中指定的內嵌語句，然後執行下面第 3 步；若其值為假(為 0)，則結束迴圈，轉到第 5 步
- (3) 執行運算式 3
- (4) 轉回上面第 2 步繼續執行
- (5) 迴圈結束，執行 for 語句下面的一個語句
- (6) 執行過程由右圖表示

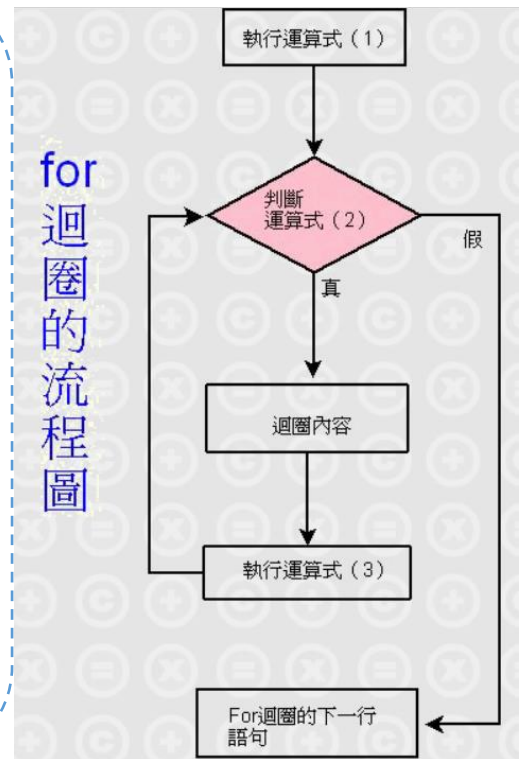


圖 3.1 : for 迴圈的流程圖

```
1 #ex1 簡單的 for 迴圈範例
2 def dia(w):
3     for i in range(1, w):
4         print((w-i)*" "+i*"")
5     for i in range(w):
6         print(i*" "+(w-i)*"")
7 dia(5)
8
```

Filename: .py

```
*
**
***
****
*****
*****
****
***
**
*
```

依指定的次數重覆執行一段程式

圖 3.2 : for 迴圈的簡單運算

```

1 #ex1 簡單的 for 迴圈範例
2 class w8():
3
4     def __init__(self, star):
5         self.star = star
6
7     def diamond(self, w):
8         for i in range(1, w):
9             print((w-i)*" " + i*self.star)
10        for i in range(w):
11            print(i*" " + (w-i)*self.star)
12
13 w = w8("a")
14
15 w.diamond(5)
16

```

Filename: .py

```

a
aa
aaa
aaaa
aaaaa
aaaaa
aaaa
aaa
aa
a

```

這兩個迴圈是以文字或圖形製作出類似鑽石 (diamond) 形狀的圖形, a 與 ab 需依照字數增減空格已達到圖形對稱, 形成正確的圖形

```

1 #ex1 簡單的 for 迴圈範例
2 class w8():
3
4     def __init__(self, star):
5         self.star = star
6
7     def diamond(self, w):
8         for i in range(1, w):
9             print((w-i)*" " + i*self.star)
10        for i in range(w):
11            print(i*" " + (w-i)*self.star)
12
13 w = w8("ab")
14
15 w.diamond(5)
16

```

Filename: .py

```

ab
abab
ababab
abababab
ababababab
ababababab
abababab
ababab
abab
ab

```

圖 3.3 : 兩個 for 迴圈差異

3.4 判斷式

運用 if 判斷冒號以下的函式是為真還是為假，而符合 True 的條件則為真；符合 False 的條件則為假，因而產生不同的結果。

```

def reverse(ev):
    global anim, direction
    # 當 anim 為 hold 時, 按鈕無效
    if anim != "hold":
        if direction == True:
            direction = False
        else:
            direction = True

```

True: 為真

False: 為假

用if判斷式判斷函式符合True的條件或False的條件

True的T及False的F一定要大寫(語法規定)

圖 3.4 : 判斷式

3.5 數列

利用 `range()` 能產生重複數列，而括號中的數值需依照需求指定起始數字或增量 (increment) 做變更

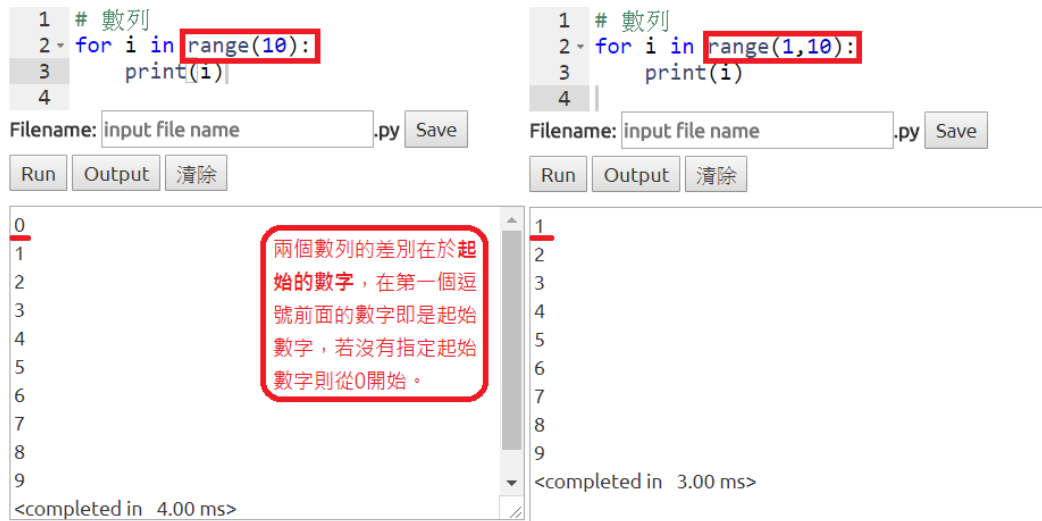


圖 3.5 :起始位置差別

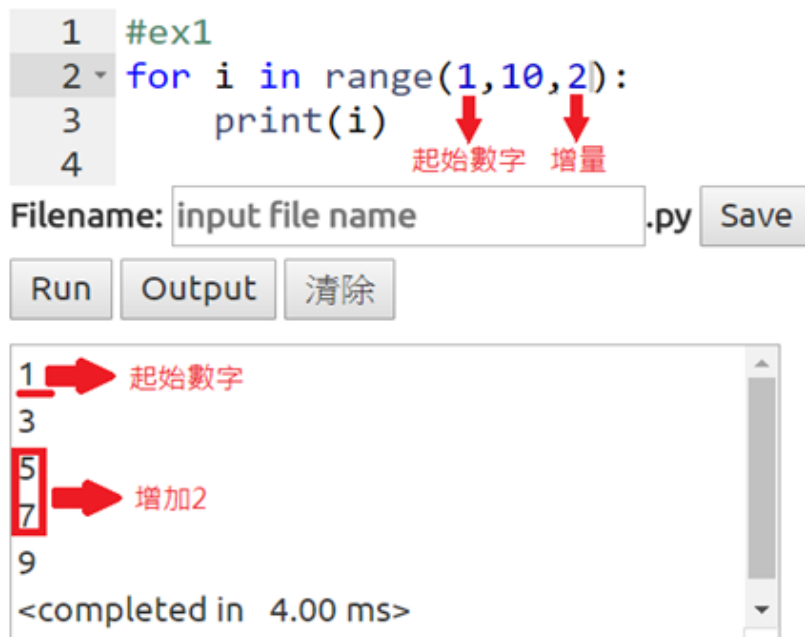


圖 3.6 : 增量數值

第四章 PyQt5 簡介

PyQt5 簡介

PyQt 包含了大約 440 個類型、超過 6000 個的函式和方法。

「QtCore」模組主要包含了一些非 GUI 的基礎功能，包含事件迴圈與 Qt 的訊號機制。此外，還提供了跨平台的 Unicode、執行緒、記憶體對應檔案、共享記憶體、正是表示式和用戶設定。

PyQt5 流程

以下是在網路看到別人使用 PyQt 開發 python GUI 程序的主要流程：

- 1、安裝開發環境：python 3.5.2 + PyQt5 5.6
- 2、利用 Qt Designer 畫介面，也就是控制項、菜單、狀態欄等等，重點在於了解 PyQt5 的信號/槽機制
- 3、使用 pyuic5 命令將 2 中生成的.ui 文件轉換為.py 文件
- 4、通過 python 的繼承功能實現介面與業務邏輯的分離
- 5、將耗時操作放入子線程解決介面假死問題，重點在於如何實現子線程以及線程間通信
- 6、開闢多線程並行工作，這裡涉及到線程間同步問題
- 7、實現子線程的停止功能，如何安全退出子線程
- 8、將代碼打包發布為 exe 應用程式，主要涉及 pyinstaller 的安裝、使用等

Qt 是由 Qt Project 開發。Qt 支援平台種類眾多，由於 Qt 的功能極為強大，英國的 Riverbank Computing 公司率先為其撰寫 Python 語言的套件，甚至開發了 SIP 這套工具將 C 與 C++ 程式庫包裝為 Python 套件。

PyQt 幾乎支援 Qt 大部分的功能

第五章 Calculator 程式

Calculator 程式細部說明

5.1 建立對話框

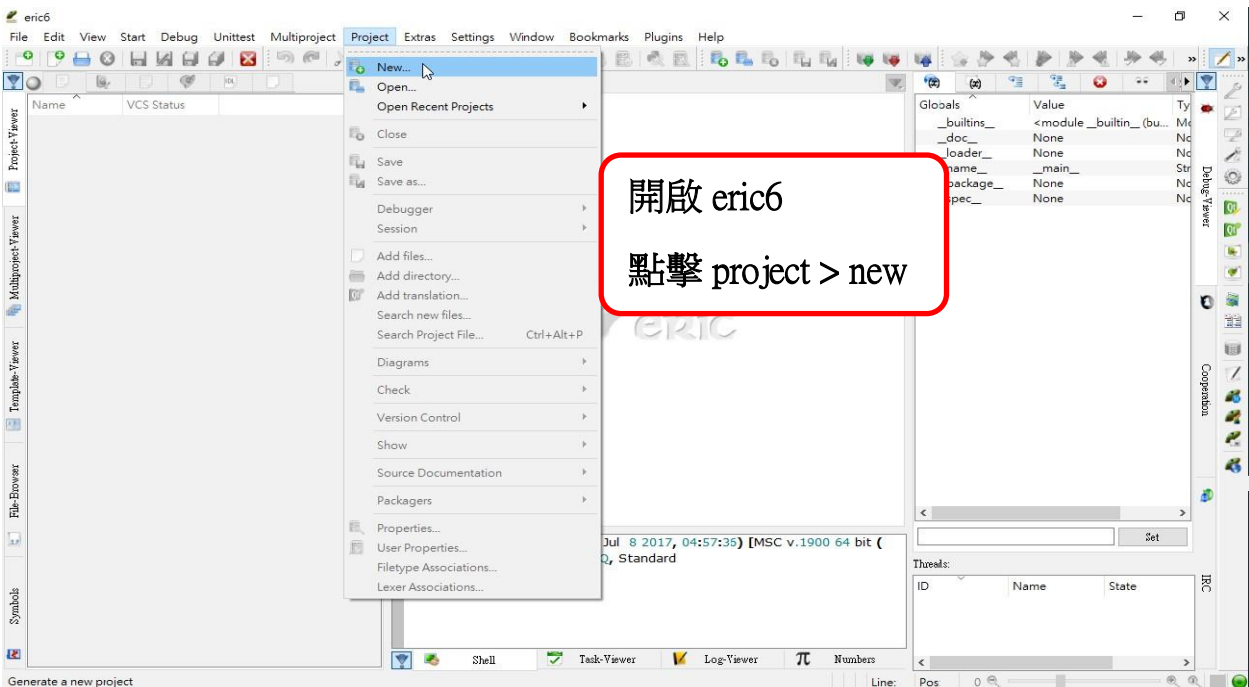
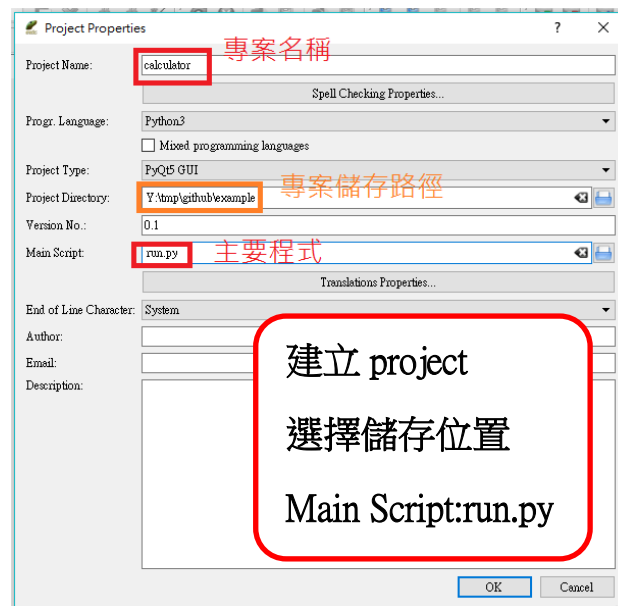
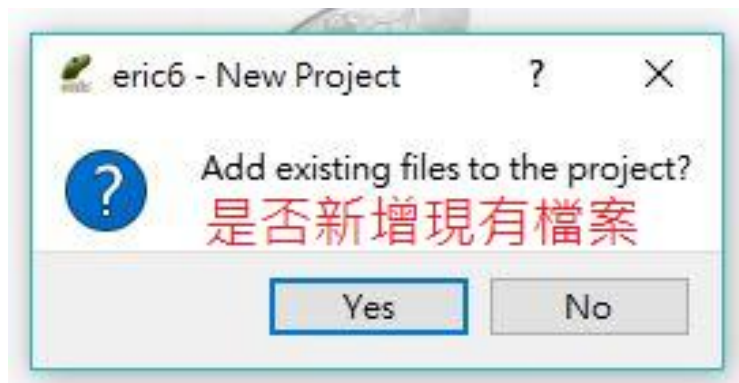


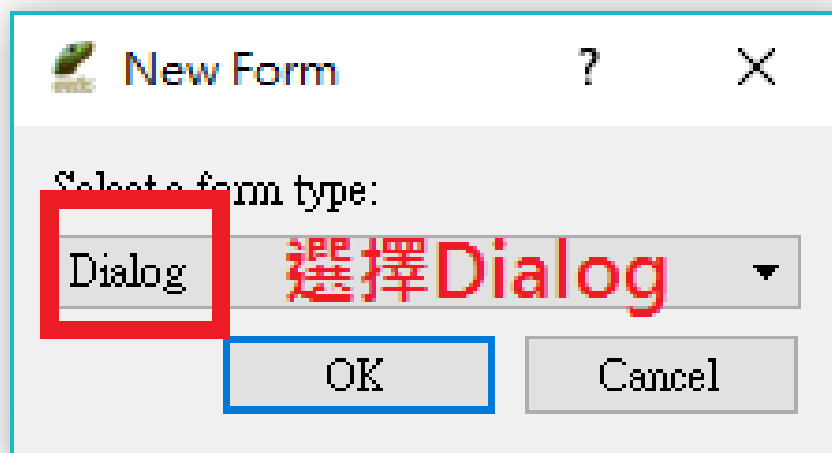
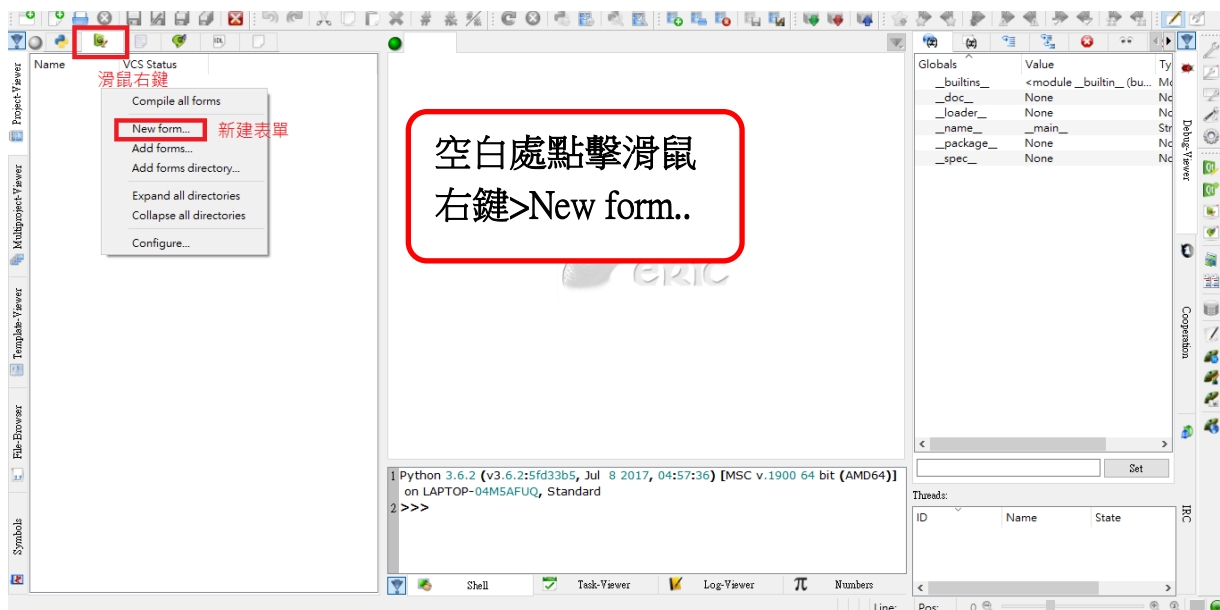
圖 5.1: 建立對話框 1



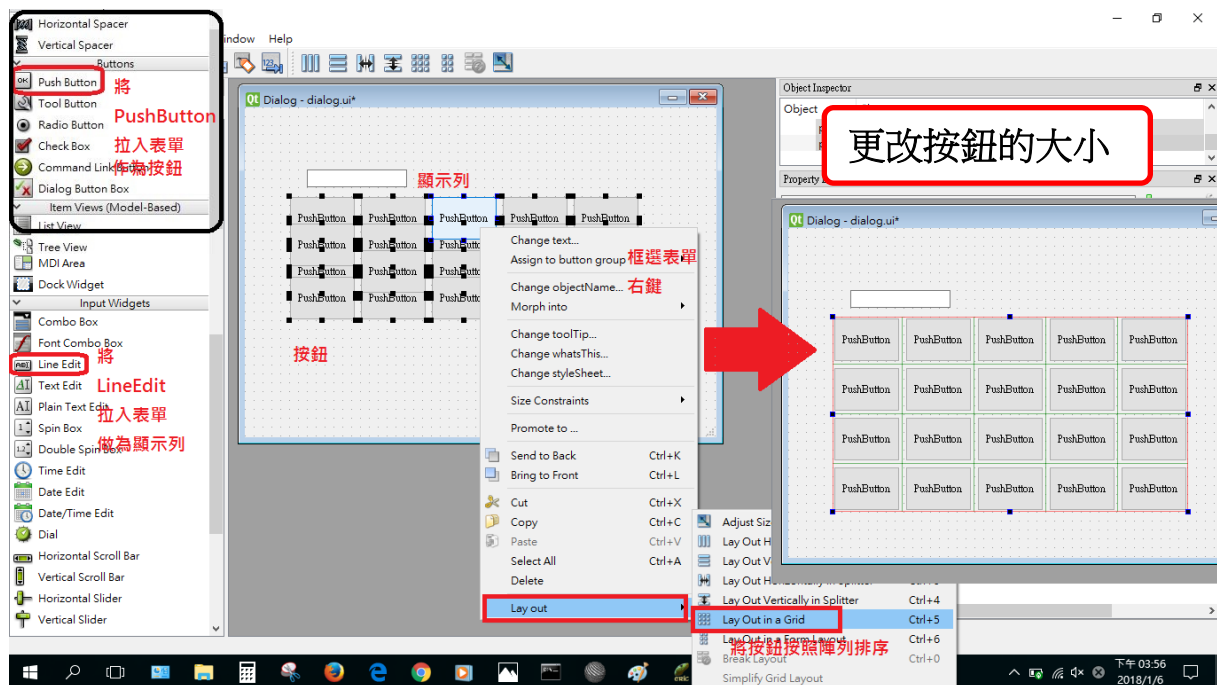
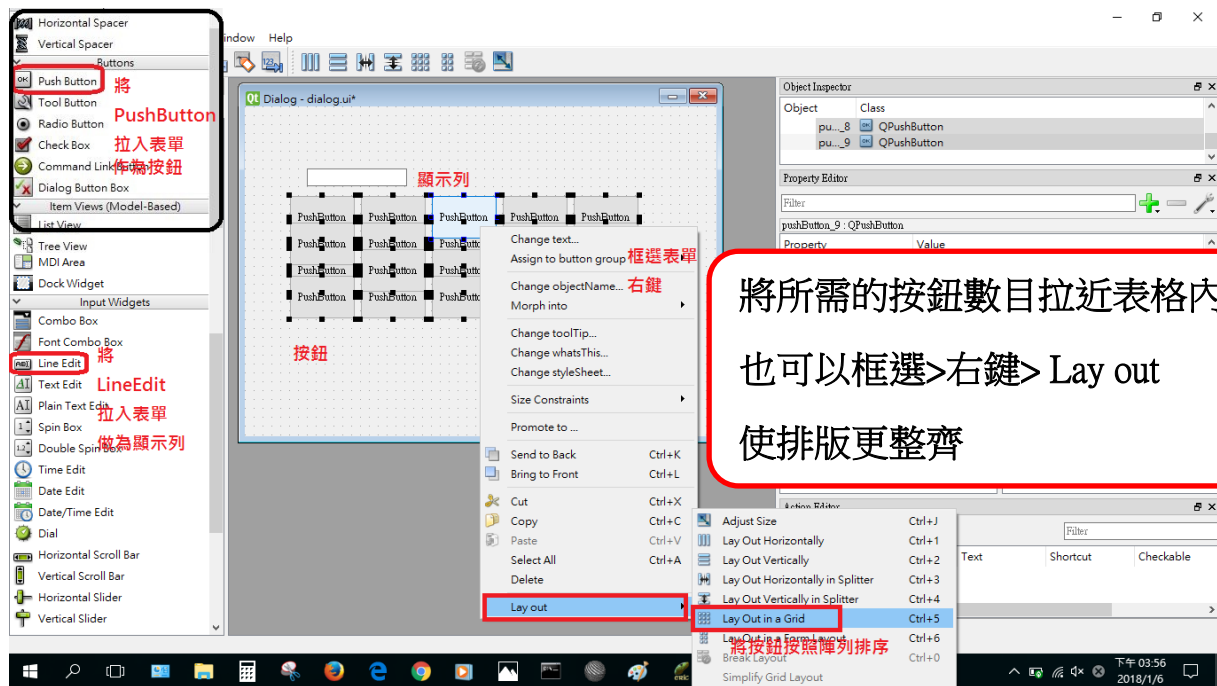


若有現有檔案點選 Yes
反之選 No

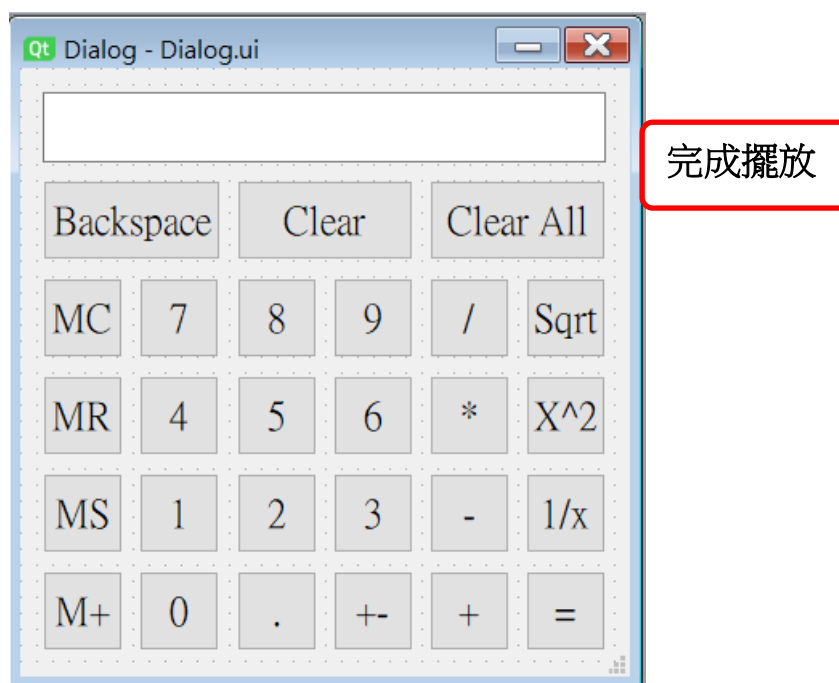
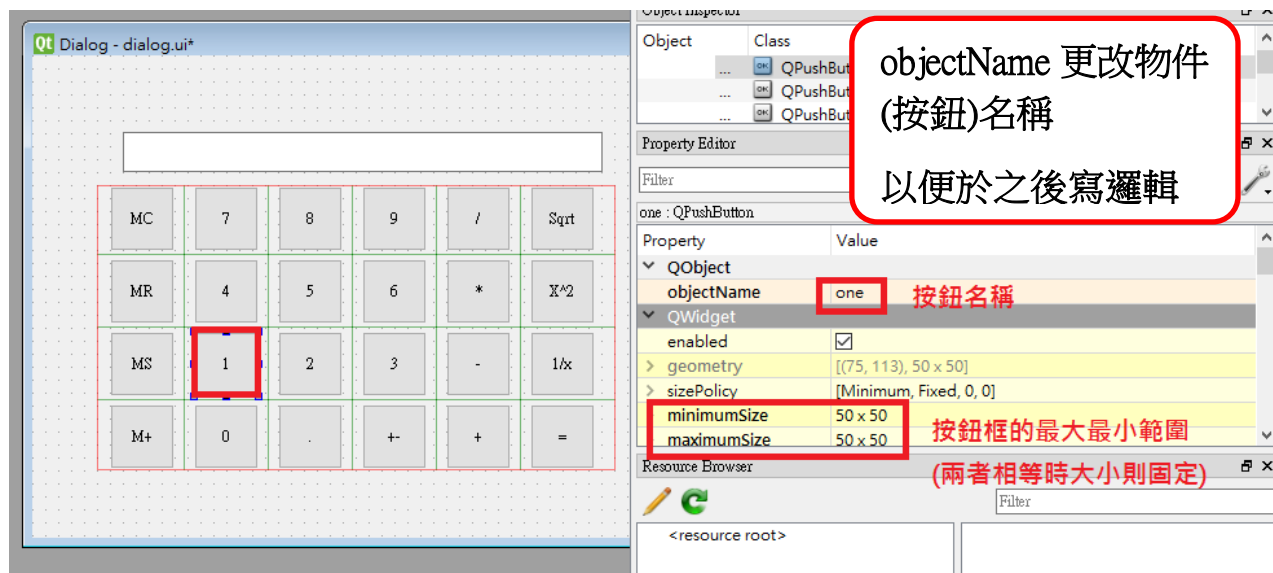
5.2 建立按鈕



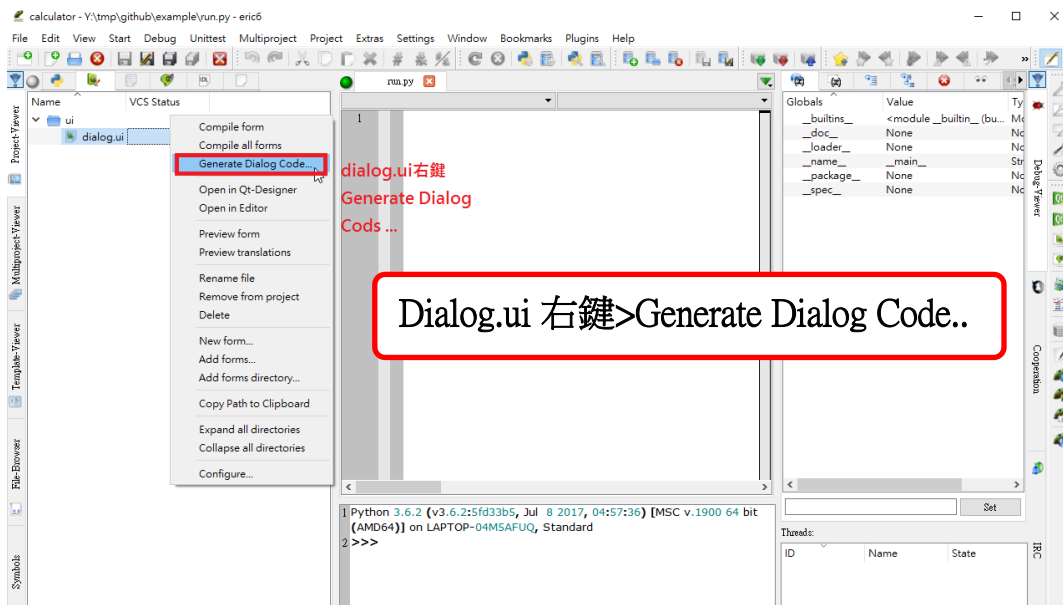
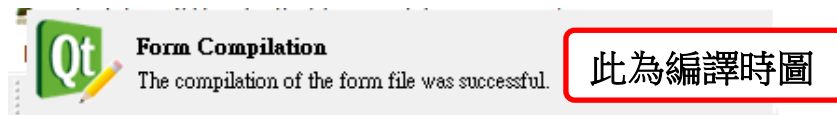
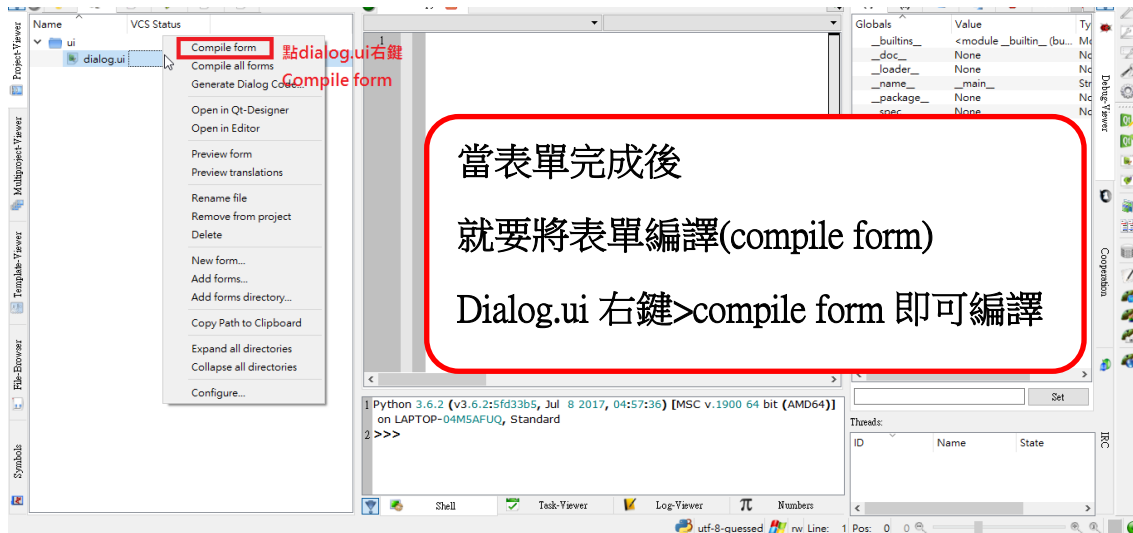
選擇 Dialog
後，按下 OK

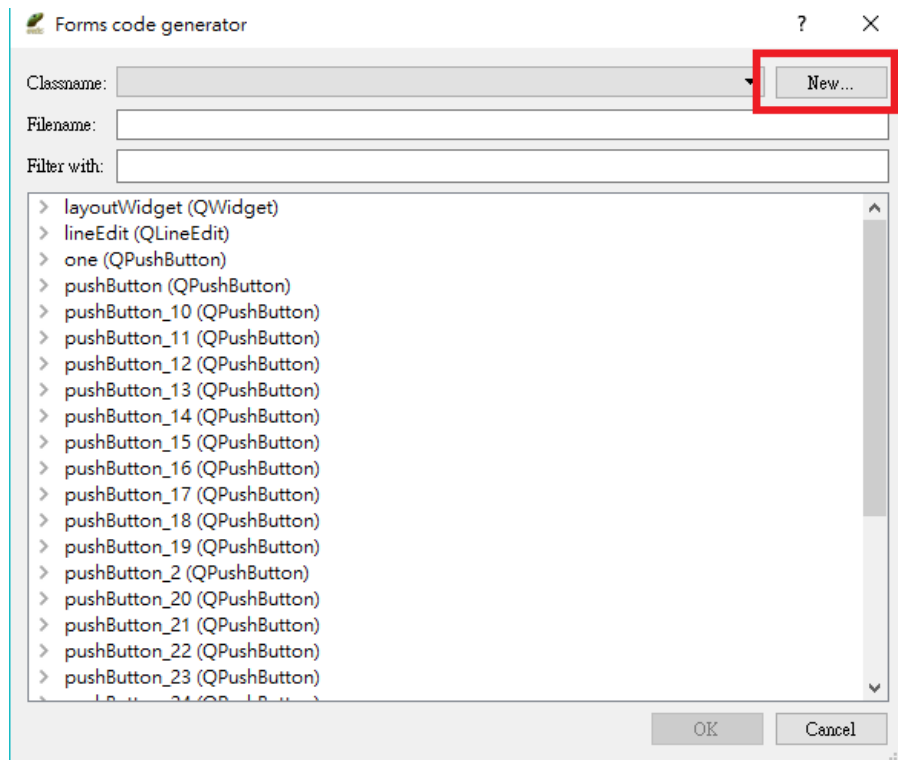
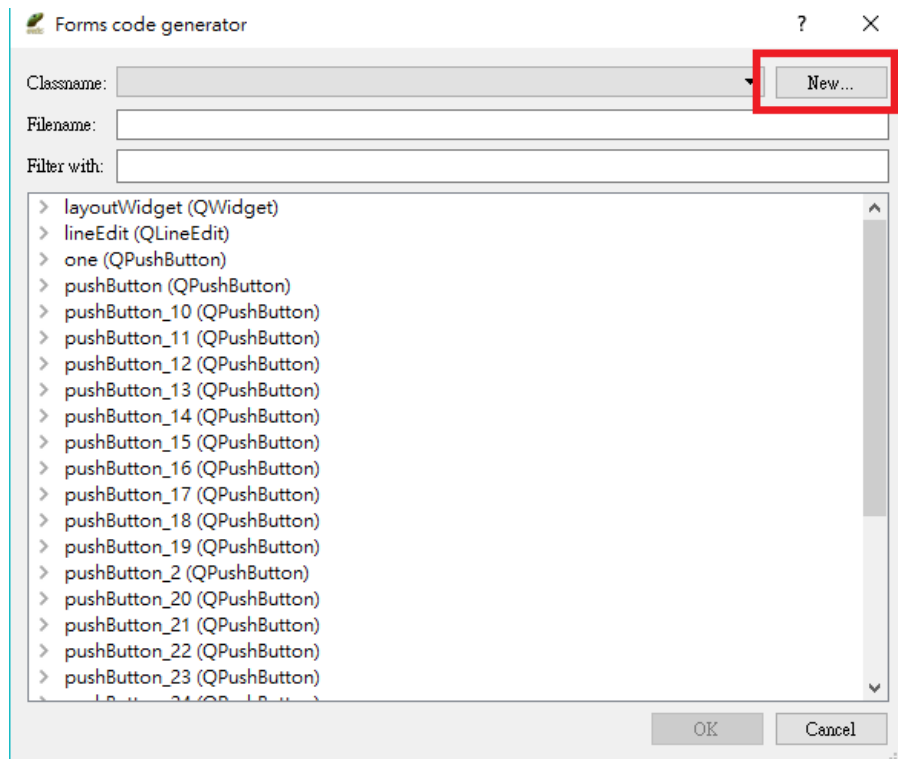


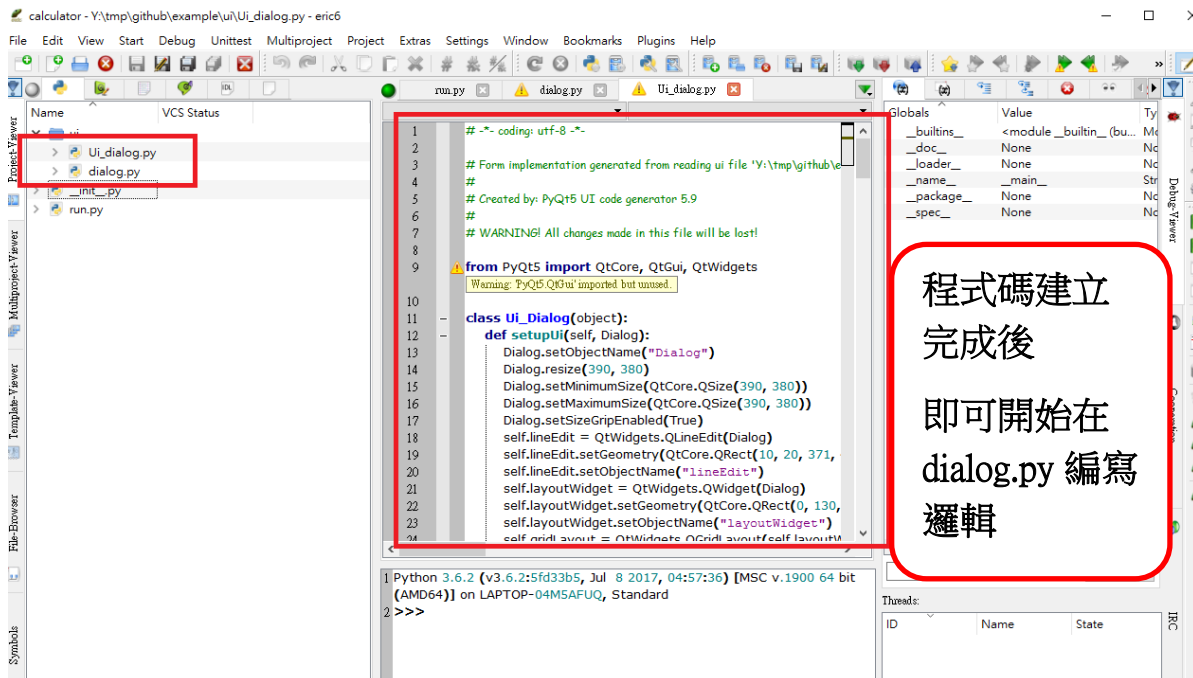
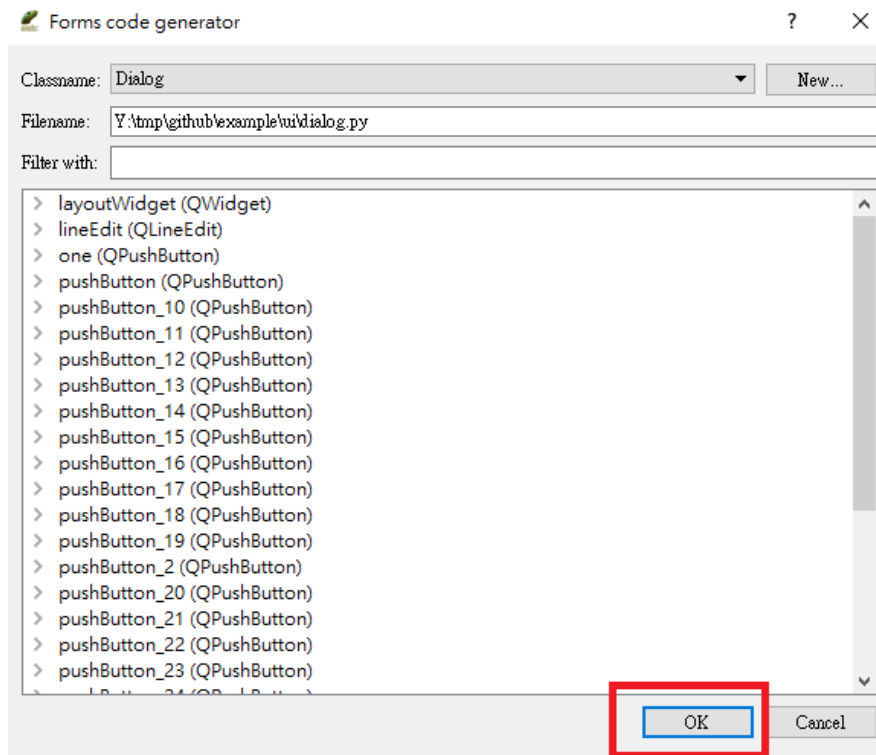
Layout 過後的排版如上

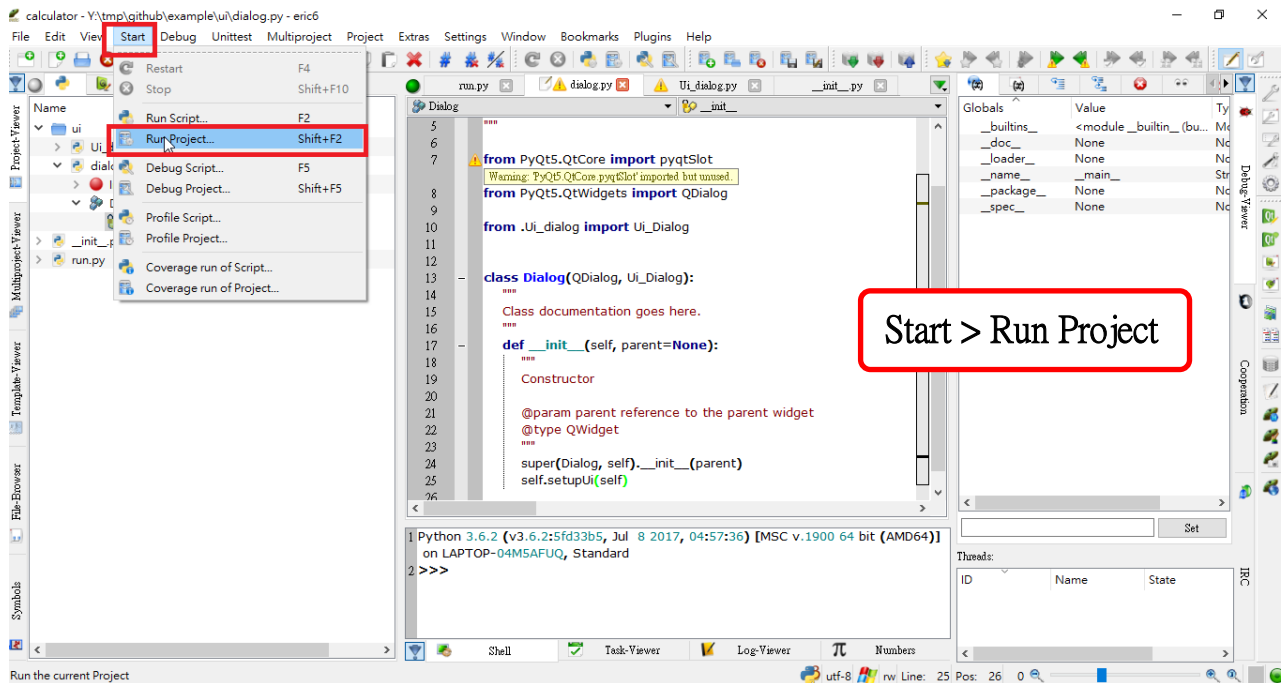
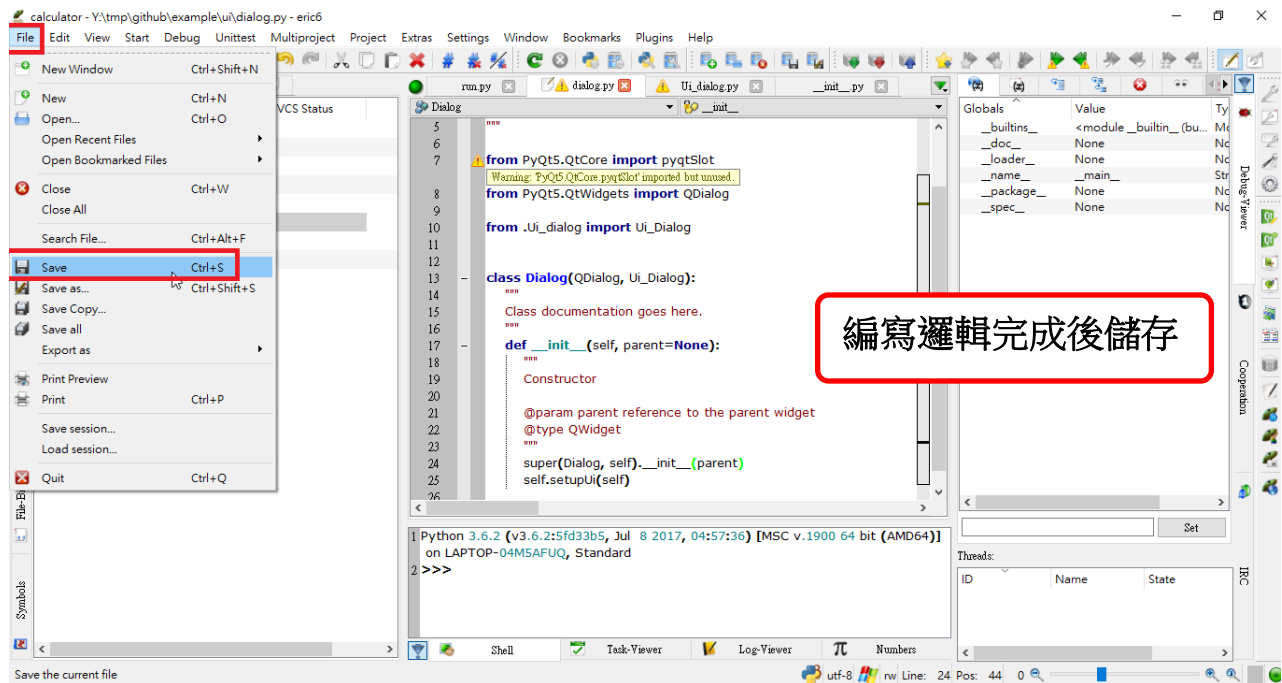


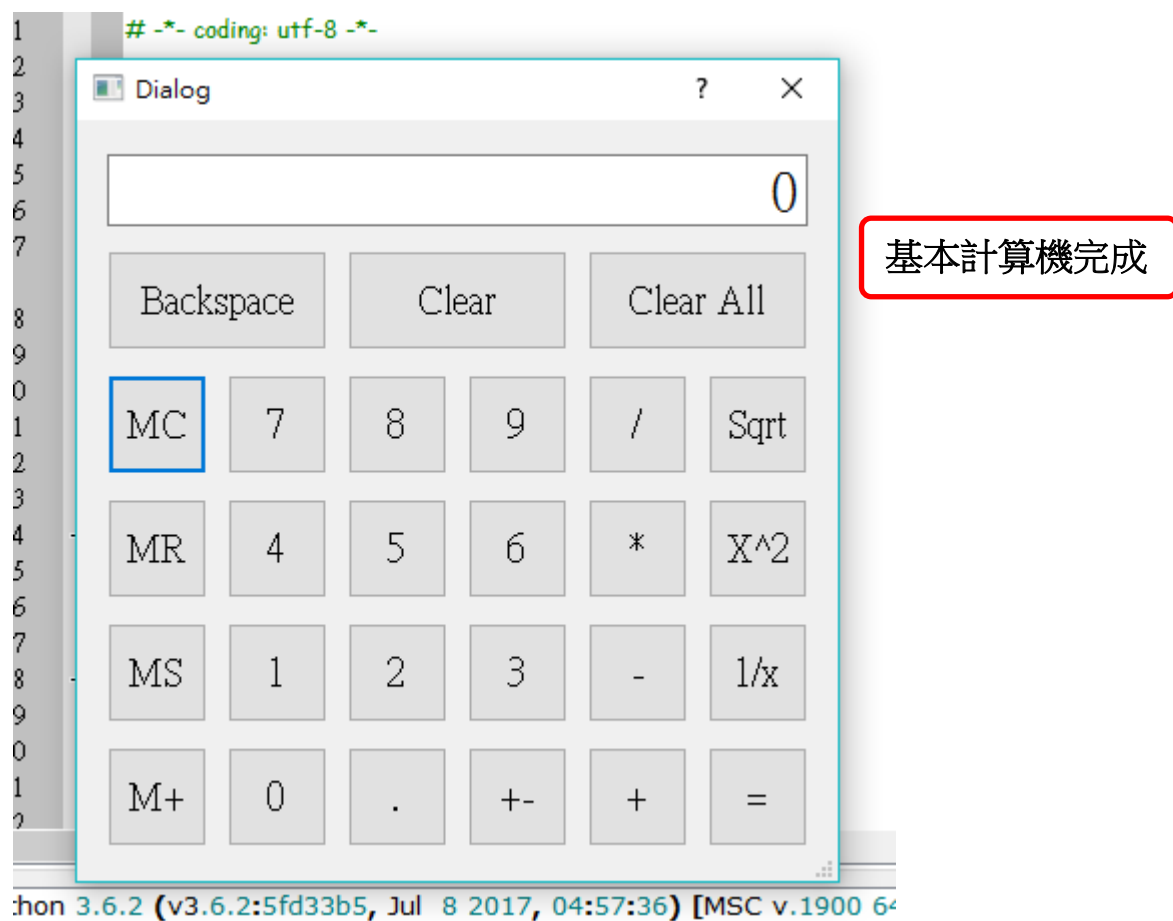
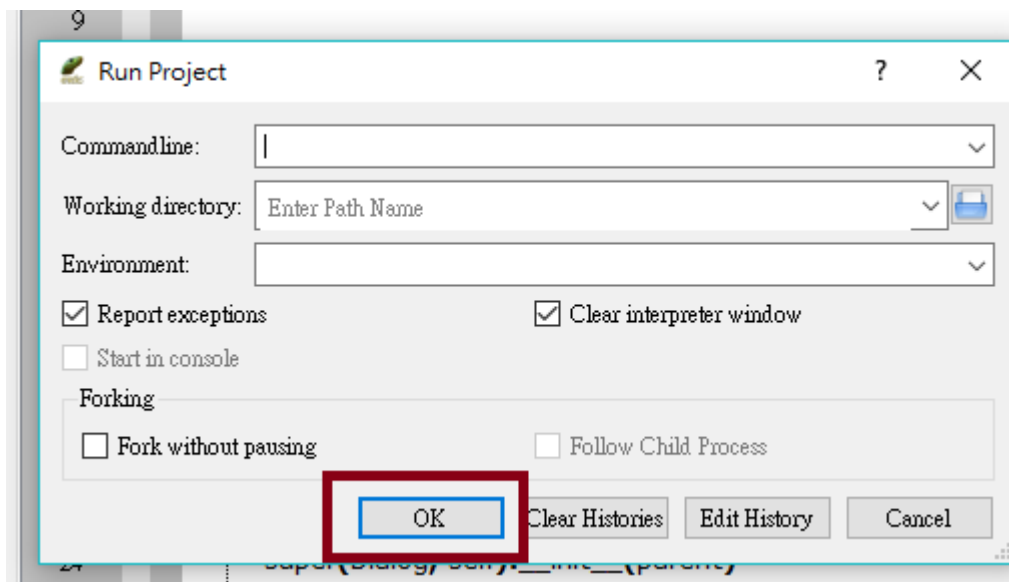
5.3 建立程式碼











第六章心得

期末報告心得

Fossil SCM

Fossil 是一個分散式版本控制系統、缺陷跟蹤管理系統以及在軟體開發中使用的 wiki 軟體伺服器。

對於設計師而言，日常工作中最常使用的工具，可能會是編輯器，或專為某種程式語言所設計的整合開發環境；而對負責軟體開發工作的軟體團隊成員來說，版本控制系統則是另一套相當重要的軟體工具。如果沒有版本控制系統，大型軟體的開發團隊成員將難以有效控制軟體版本，並可能導致程式臭蟲增加。

一般人對於軟體本身的使用需求，多半是希望操作越簡單越好，並有相當程度的穩定性與可靠性。而操作簡單與系統本身穩定性高，正是 Fossil 所強調的二大重點。一般人即使沒有使用版本控制軟體的經驗，也能在閱讀 Fossil 提供的簡單文件之後立即上手。Fossil 之所以可以作為官方網站的平台，是因為除了版本控制系統相關的功能以外，亦提供了程式臭蟲追蹤 (Bug Tracking) 與維基共筆系統的支援能力。與本身的版本控制系統功能類似，Fossil 的程式臭蟲追蹤功能與維基共筆系統採用分散式的處理方式。

利用 Fossil 作為 Blog 平台的架設解決方案，所以無論使用者需要的是單純的版本控制，或是希望架設網站作為資訊分享的平台，都能利用 Fossil 一併解決。

Fossil SCM 心得

40623201 - 第一次接觸網路倉儲才知道原來倉儲有那麼多好處,能夠用 Timeline 看每個東西推上去的時間,也能透過網址去 clone 東西,當我隨身碟不小心中毒時,重新 clone 東西就都回來了

40623202 - 第一次接觸到 Fossil SCM 的時候可以說是完全搞不懂,而經過幾次上課後也慢慢懂得各個指令的用處,也了解它的功能有甚麼,可以說是基礎,對於往後課程的理解也較有幫助

40623203 - 剛開始上課時,一直聽不懂老師在教什麼,也不知道 fossil 要怎麼使用,但是隨著老師和同學的教導,慢慢地明白課程的內容,在上課不懂的地方我都會回去用電腦重複看幾次老師拍的影片,不懂就再問同學,來來回回幾,我已經越來越熟悉並且漸漸地跟上老師的進度了,在錯誤中學習經驗

40623210 - 使用過後讓我深深了解,一個設計者就算沒網路也能使用近端隨身系統作業,使用 Fossil SCM 可以有效率的版次管理。就算隨身系統遺失也可以馬上 clone 遠端版本。

40623211 - 第一次接觸 fossil,學習到如何靈活應用近端和遠端,usb 管理和多元的指令,是一個很棒的經驗

40623212 - 再接觸 fossil 時一開始甚麼都不懂,指令的功用也不懂,但經過這幾周下來我學到非常多東西也了解到倉儲是多麼便利的東西,不僅僅只是把東西存起來,即使改版後也會留下先前的資料。

網誌心得

40623201 - 網誌 : <https://cpb.kmol.info/40623201/doc/trunk/blog/>

心得 : 能夠透過 leo 編輯網誌將每個禮拜所學到的東西紀錄上去; 也能編輯投影片做重點摘要, 雖然一開始對推送過程不是很理解, 常常近端推上去遠端推不上去, 但經過幾次的推送後已經熟能生巧了

40623202 - 網誌 : <https://cpb.kmol.info/40623210/doc/trunk/blog/>

心得 : 網誌可以記錄上課的內容, 雖然一開始不懂近端跟遠端的差別, 也不知道如何推送, 但後來越做越熟悉了, 已經可以輕鬆地使用網誌紀錄

40623203 - 網誌 : <https://cpb.kmol.info/40623203/doc/trunk/blog/>

心得 : 在老師每週上課結束後, 撰寫網誌能夠有效地複習課程, 可以在網誌上編寫遇到問題和解決辦法, 讓有需要的人可以看網誌解決所發生的問題。一開始透過 Leo 編輯網誌時, 遇到很多問題, 使得我每次製作時都心浮氣躁, 但是當越來越熟悉時, 我慢慢地覺得有趣, 製作起來也比之前還要順手, 還能幫助別人解決問題, 現在已經能夠輕輕鬆鬆地使用 Leo 編輯網誌了

40623210 - 網誌 : <https://cpb.kmol.info/40623210/doc/trunk/blog/>

心得 : 每週撰寫網誌心得不但能有效地複習課程, 並且在遇到問題時可以撰寫上去, 未來遇到相關問題時才有資料可以回朔

40623211 - 網誌 : <https://cpb.kmol.info/40623211/doc/trunk/blog/>

心得 : 第一次接觸 fossil, 學習到如何靈活應用近端和遠端, usb 管理和多元的指令, 是一個很棒的經驗

40623212 - 網誌 : <https://cpb.kmol.info/40623212/doc/trunk/blog/>

心得 : 我們或許會每天寫日記, 但是只能寫在書上, 還未必能隨要隨看, 但有了網誌, 現今手機的普遍以及網路資源發達, 已經能再網誌上將課程學習內容..等之類的東西記錄下來了。即便不用網路也能在近端作業。

6.1 Github 協同倉儲

bg1 協同倉儲: https://github.com/40623203/bg1_pyqt5_calculator

40623201 - github 是個很好用的協同工具，即使很多人一起改東西，同時推東西上去，就能發生衝突，只要排除衝突都能夠把彼此所做的東西推上去

40623202 - 使用 github 協同不僅可以記錄修改的版次，協同者們也可以在各處一起協同，學會如何解決衝突，以及用 `git pull` 更新從遠端 clone 下來的資料，操作幾次過後，了解多人協同者的好處也能好好的利用這項功能，受益良多。

40623203 - Github 在處理分組協同十分的方便，不需要一群人待在同一個空間製作，每個人做完自己的部分後，再推上遠端，需要進行之後的步驟就在 `git pull` 拉下最新的資料，就能夠開始編輯了。在這當中我學到了如何解決衝突，操作幾次後，了解 Github 在製作這方面的便利性，也能夠好好的運用

40623210 - 在分組協同期末專案後我發現 Github 的方便性，他會整合每個協同者新增的資料並且能看到新增了哪些資料。

40623211 - 和組員協同倉儲更有效的利用時間，不用透過繁瑣的討論，各自把個人東西完成，便能了解彼此之間完成的進度。

40623212 - 再學習 github 倉儲中第一次接觸到協同，只是單方面的把東西傳上倉儲裡，但沒有處理協同問題，就會因為別人有推送過，而導致衝突發生，這時候就要使用 `git pull` 來處理衝突，然後再上傳。協同並不只是對單一伊人而是對所有的開發者，所以也需要其他人的幫忙才得以完成。

6.2 學員心得

- 40623201 -這學期我學到了如何用倉儲管理以及運用 github 網路倉儲與組員協同合力做出完整的計算機，也學到了分配工作以及團隊合作的重要性，每個人會的領域都不同，能互相學習大家的優缺點，雖然還是會有些不太熟悉或不理解的地方，但透過組員一起提出想法解決問題，能學到許多東西。
- 40623202 -這幾週以來,從一開始的拉計算機表單到開始寫計算機運算的邏輯,利用 github 協同開始分組完成計算機,在寫乘除運算的邏輯時遇到不少瓶頸,還好有組員們幫忙找出問題並解決,此時我真的了解到團隊的重要。
- 40623203 - 在製作計算機的過程中，我從什麼都不會到漸漸地越來越熟悉，可以解決不同的問題，從問題中找出不同的解決辦法。要讓計算機能夠運作，需要用很多心思去編寫，多一個字或少一個字都會造成計算機的程式無法執行，所以在製作過程中必須兼具細心與智慧，才能完成計算機程式的編寫，雖然過程很艱苦，但是完成後會感到收穫滿滿，很有成就感。在這個學期，我學到了很多以前沒學過，甚至從來沒想過會接觸程式這方面的知識，讓我受益良多，並且提高自身的基礎能力。
- 40623210 -這幾週在製作過程中，我學到如何利用 eric6 開啟 Qt designer 製作表單，並在製作數字及小數點中，知道如何產生 signal 和 slots 以及些微瞭解邏輯中的意思。分組的好處在於有問題能及時尋求幫助並解決，這也讓我了解到團隊分工的重要性。
- 40623211 - 經過這幾週的計算機製作,我學到如何應用 PyQt 和 eric6,更學到分工合作的重要,有效的分工,能讓工作流程更順暢,各個組員盡守本分,使我們更團結一心,其中我在製作根號時,因為一個英文字母漏掉,導致系統無法運算,所以我看到電腦打字準確性的重要,第一次自己製作計算機是一個好棒的體驗。
- 40623212 - 從計算機程式，課程上到現在，已經對計算機有初步的了解，小時候常常看到的計算機，原以為只是個普通又簡單的東西，但現在學到計算機程式才知道，原來後面的邏輯是多麼的複雜，聽老師說開發計算機的人連按鍵都是用程式寫，我們只是略懂皮毛一二，希望能有更多的了解。

說明各學員任務與執行過程

40623201 - 任務 - 計算機 (Clear All、Backspace)、報告(數列、判斷式)

執行過程 - 計算機(先查詢老師的步驟內容的意思，理解後，再將東西打上去，看是否能執行，再進行推送。)、報告(先將圖片選取好後，將圖片進行編輯加文字及註解上去，再用 Microsoft Word 將內文及編輯好的圖片放置上去。

40623202 - 在製作計算機時常因為一個字母的大小寫就影響了整個運作，再提交以及推送時也常常因為一個路徑錯誤搞老半天，當一個設計工程師真的需要非常細心、有耐心。

40623203 - 任務 - 計算機：+、-、=、變號鍵、中斷運算、計算及最後檢查

報告：Python 程式語法 (變數命名、print 函式、重複迴圈及排版檢查)

執行過程 - 計算機：一開始完全看不懂計算機的程式碼，再看老師的網誌及上網 google 查詢，才漸漸地編寫出程式碼，在製作計算時，沒有修改到 * 與 /，導致其他組員在製作時，一直跑出錯誤。

報告：一開始，無法將老師上傳的檔案轉至 PDF 檔，發現原來是多了一行 import chapter，導致錯誤發生。原本要使用 Pandoc 與 MikTeX 協同編寫期末報告，但之後出現了問題，所以臨時改成用 word 製作。

40623210 - 一開始自行練習時發現 eric6 打不開 Qt designer,尋求助教幫助後發現 eric6 中的設定跑掉重設後才恢復完整。在執行期末專案的過程中忘記導入 signal,後來尋求組員幫助才了解每段邏輯的重要性。

40623211 - 學員任務 - 我負責開根號和等於運算結果，了解 for 迴圈後其實看著老師給的檔案學習，還不算太難

執行過程 - 我在製作根號時，因為一個英文字母漏掉，導致系統無法運算，所以我看到電腦打字準確性的重要，第一次自己製作計算機是一個好棒的體驗。

40623212 - 我被分配到記憶按鍵，起初我連記憶按鍵是甚麼東西都不知道，所以起初很難了解邏輯上的建設怎麼開始，後來使用老師完成範本的計算機嘗試使用記憶按鍵來了解其功用最後終於了解記憶按鍵是甚麼了，也了解邏輯如何寫起，起初以為記憶按鍵中的 M+要配合加號的邏輯才能使用，但最後了解到 M+是跟加號分開的，是可以獨立進行，不

因為加號的邏輯而被干擾。而是直接把記憶按鍵中的儲存數設為真數再加上對話框所顯示的數字，因此無需使用到加號的邏輯。

第七章結論

期末報告結論

7.1 結論與建議

結論:經過這幾個禮拜的分組協同、製作報告，雖然遇到很多問題，但是經過彼此的幫忙，努力完成。