

國立虎尾科技大學

機械設計工程系

計算機程式 ag8 期末報告

PyQt5 事件導向計算機

PyQt5 Event-Driven Calculator Project

學生：

設計一甲 40623140 韓希然

設計一甲 40623141 何立翔

設計一甲 40623142 郭益綸

設計一甲 40623144 林昭權

設計一甲 40623150 仝允丞

設計一甲 40623152 潘季宏

指導教授:嚴家銘

2017.12.18

摘要

了解 fossil 和 github 協同倉儲，並運用在計算機上

- 視窗命令列與 Fossil SCM 常用指令
- Python3 程式語法練習
- 客製化可攜環境及 PyQt5 程式編寫
- Github 協同倉儲
- 普通計算機程式設計，採用 PyQt5 與 Python3 建立，預計採用事件導向的方式

目錄

摘要.....	i
目錄.....	ii
第一章 前言	4
第二章 可攜環境介紹.....	5
2.1 啟動與關閉 1	5
2.2 啟動與關閉 2(客製化設定)	7
第三章 Python3 程式語法	9
3.1 變數命名	9
3.2 print 函式	9
3.3 重複迴圈	10
3.4 判斷式	10
3.5 數列	11
第四章 PyQt5 簡介	12
4.1 PyQt5 架構	12
4.2 PyQt4 與 PyQt5 區別	12
第五章 Calculator 程式	13

5.1 建立對話框	13
5.2 建立按鈕	15
5.3 建立程式碼	17
第六章 心得	19
第七章 結論	23
7.1 結論與建議	23

第一章 前言

第八組 機算機程式期末報告前言

1.1 前言內容：

何謂機械設計？

所謂機械設計，就是透過詳細思慮的考量，完成包括固體 流體和軟體等互動元件之精巧配置，使其展現特定功能之具體規劃表達。

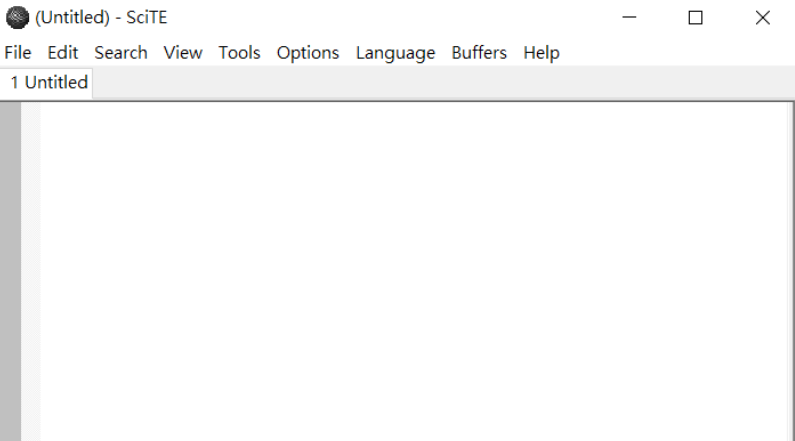
工程師有六種表達方式 → 口語 文字 2D 3D 數學理論與實作表達

創造力有三種基本要素：自學力 執行力 想像力

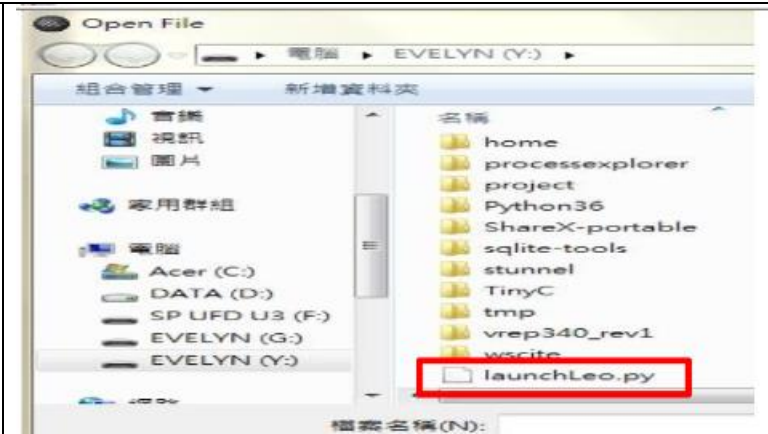
第二章 可攜環境介紹

可攜環境介紹

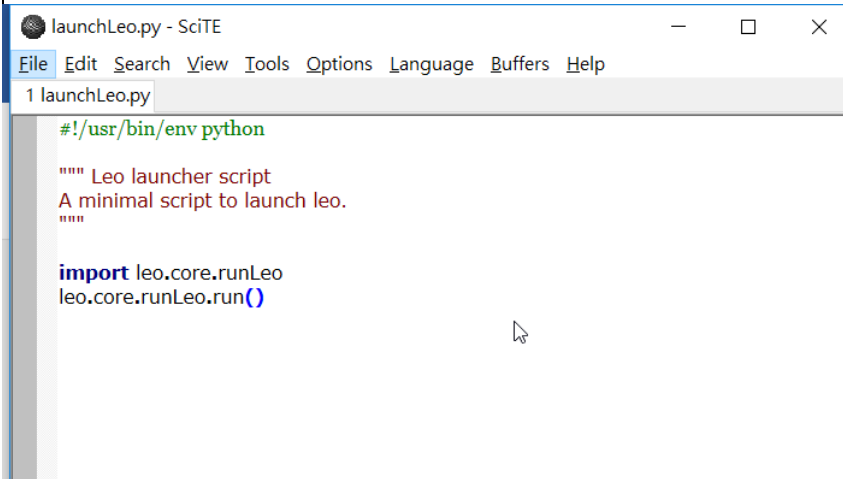
2.1 啟動與關閉 1

步驟	操作圖片
1. 開啟 Start.bat	
2. 打開 SciTe	

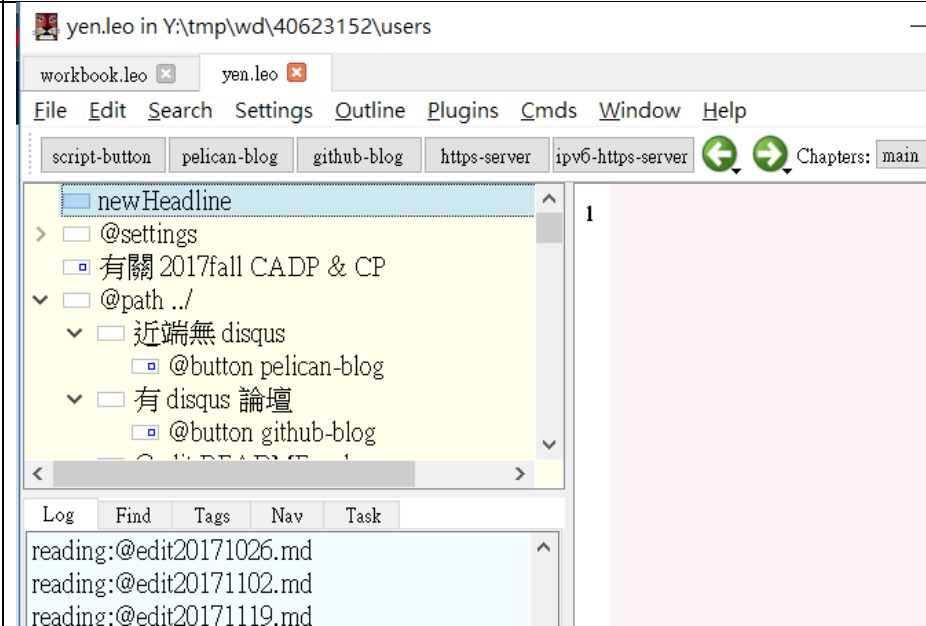
3. 開啟
launchLeo
.py




4. F5
go 它



5. 開啟
自己的
leo 頁面



2.2 啟動與關閉 2（客製化設計）

<p>將 start.bat 拉入 SciTE 更改其中的 數量</p>	 <pre>start2.bat SciTE File Edit Search View Tools Options Language Buffers Help 1 start2.bat REM 14592021 主要針對初學 Python3 與 C 語言所建立 REM 以備常用 fossil 管理資料庫, 並且定時將各 git 格式庫上傳到 github @echo off REM 設定 y 磁碟代號與 data 目錄對應 set Disk=y set %Disk%=%Disk%\data REM 設定 leo 相關對應 Home 位置 set HomePath=%Disk%\Home set HomeDrive=%Disk%\Home set Home=%Disk%\Home REM 讀取 python 環境的 is 型為 utf-8 set PYTHONENCODING=utf-8 REM 將此磁碟的指令執行, 以 %Disk% 為主 %Disk% REM 設定 PYTHONPATH set PYTHONPATH=%Disk%\python3 REM 設定 Leo 所開的編輯器 set Leo_EDITOR=%Disk%\wscite\SciTE.exe REM 設定 fossil http 連線設定 REM 當在遠端使用 fossil ui 則無效 REM set HTTP=no REM 將主機連線設定 REM set path=%PATH%;%Disk%\python3;%Disk%\git\bin;%Disk%\tunnel\bin;%Disk%\logite-tools;%Disk%\python3\Scripts;%Disk%\Git\bin;%Disk%\vcpkg; set path=%Disk%;%Disk%\python;%Disk%\git\bin;%Disk%\tunnel\bin;%Disk%\logite-tools;%Disk%\python3\Scripts;%Disk%\Git\bin;%Disk%\vcpkg; path=%path%; start /MIN %Disk%\wscite\SciTE.exe start /MIN cmd.exe start /MIN cmd.exe REM 啟動 Leo 編輯器 REM %Disk%\Maincode\python.exe %Disk%\apps\launchLeo.py REM 啟動 tunnel REM start /MIN fossil.exe server -P 127.0.0.1:8080 %Disk%\tmp\fossil_repo REM start /MIN tunnel.exe Exit</pre> <p>輸入幾個就會開出幾個DOS</p>
--	--

第三章 Python 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', 'non
local', '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 for 迴圈

print() 函式用法

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。

for 迴圈用法與 Python 的縮排規定

重複迴圈用法，使用者可以透過下列程式編輯區練習 for 迴圈與 print() 函式的用法。

```
1 #ex1 簡單的 for 迴圈範例
2 for i in range(10):
3     print(i)
4
```

Filename: .py

```
0
1
2
3
4
5
6
7
8
9
<completed in 25.00 ms>
```

指的是1到9

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
```

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)

True 為真
False 為假

3.5 數列

定義：數學上有個有趣的費式數列（Fibonacci series），頭兩個數字為 0 和 1，之後的數字為前兩個數字的和。底下程式計算小於 n 的最大費博納西數（Fibonacci number）

Python 舉例：

```
# coding=UTF-8

def fib(n):
    if n == 0 or n == 1:
        return int(n)
    else:
        return fib(n-1) + fib(n-2)

number = input('Please type an integer: ')
print(fib(number))
```

第四章 PyQt5 簡介

PyQt5 基本架構介紹

4.1 基本架構

大部分所見的圖形化介面程式語言都是由有物件導向的程式語言開發的，例如 C++、Java、C#、Python 等。直接使用 C++ 語言「寫出」圖形介面是一件滿費心的差事，因此有滿多圖形介面的函式庫（library）可以使用，例如 Qt、Tk、wxWidgets、GTK+ 等。使用簡單的函式（function）就可創造視窗介面，並且有很多函式庫盡力克服「跨平台」的障礙。由於可以包含的部件極多，稱得上圖形介面「框架（framework）」一詞。其中 Qt 是由 Qt Project 開發。Qt 支援平台種類眾多，除了常見的 Windows、Linux、Mac 以外，還有非 X Window System 的作業系統。授權方面也十分自由，採用 GNU 較寬鬆通用公共許可證（GNU Lesser General Public License, LGPL）、GNU 通用公共許可證（GNU General Public License, GPL）、商業授權三種模式，可以讓開發者應需求選擇。Qt 程式庫中甚至支援開發圖形介面的「周邊」功能，如網路通訊、OpenGL、OpenVG、SQL 與 XML 直譯器、圖片格式轉檔、Linux 的輸入法開發、瀏覽器引擎（使用 Google Chromium）、各

式圖表等。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，並且將較專門的功能另外分成 PyQt Chart (2D 圖表)、PyQt Data Visualization (3D 圖表)、PyQt Purchasing (應用程式購買功能)。另外 QScintilla 是一個將 Scintilla 連結至 PyQt 的套件 (在 C++ 可以直接用 Qt 和 Scintilla 即可)，用途是辨識文字中的程式語言，以亮顯 (highlight) 的方式呈現，可以用作程式語言的辨識功能。PyQt 的版本與 Qt 相同 (除了小版號)，採用 GPL 和商業授權。需要注意的是，若作為軟體釋出，沒有商業授權是需要公開原始碼的。

4.2 PyQt4 與 PyQt5 區別：

兩者的區別如下：

- 重新組合模塊，一些模塊已經被廢棄(QtScript)，有些被分為兩個子模塊(QtGui, QtWebKit)。
- 添加了新的模塊，比如 QtBluetooth, QtPositioning，和 Enginio。
- 廢棄了 SINGAL()和 SLOT()的調用方式，使用了新的信號和 xx 處理方式
- 不再支持所有被標記為廢棄的或不建議使用的 Qt API