

第1次練習-練習-PC1

學號：112111207

姓名：陳品霖

作業撰寫時間：60 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2024/11/28

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 請執行下面Git操作 (請參照題目pdf)

Ans:

2.請整理課堂上所提的Git指令，顯示其語法語給予明顯的例子

Ans:

1. `git add` (加入檔案到暫存區)

語法：

```
git add <檔案名稱>
```

範例：

```
git add ty
```

結果：將檔案 `ty` 加入到 暫存區，準備提交。

2. `git commit` (提交暫存區內容到本地倉庫)

語法：

```
git commit -m "<提交訊息>"
```

範例：

```
git commit -m "新增 ty 檔案"
```

結果：

將暫存區的內容提交至 本地倉庫，並附加一則訊息。

3. `git push` (推送提交至遠端倉庫)

語法：

```
git push <遠端名稱> <分支名稱>
```

範例：

```
git push origin main
```

結果：將提交的內容從 本地倉庫 推送至遠端倉庫的 main 分支。

4. `git fetch` (從遠端倉庫獲取資料)

語法：

```
git fetch <遠端名稱>
```

範例：

```
git fetch origin
```

結果：從遠端倉庫下載更新到 本地倉庫，但不會自動合併。

5. `git pull` (拉取並合併遠端更新)

語法：

```
git pull <遠端名稱> <分支名稱>
```

範例：

```
git pull origin main
```

結果：相當於執行 `git fetch` 和 `git merge`，從遠端倉庫拉取更新並自動合併。

6. `git branch` (管理分支)

語法：

```
git branch <分支名稱>
```

範例：

```
git branch new
```

結果：建立一個名為 `new` 的新分支。

7. git checkout (切換分支)

語法：

```
git checkout <分支名稱>
```

範例：

```
git checkout new
```

結果：切換到名為 new 的分支。

8. git merge (合併分支)

語法：

```
git merge <分支名稱>
```

範例：

```
git merge new
```

結果：將 new 分支的內容合併到目前所在的分支。

3. 請先寫出同學認為資料結構一門課每周需要花多少時間？並提出理由。並將「請問要學習大學的資料結構每一週需要對於一般大學生需要多少時間」貼入chatgpt，並將其所建議的敘述貼出後，比較同學所認為若是每周低於chatgpt所評估的時間，兩者主要差異為何？應該怎麼安排自己時間了解資料結構？

Ans:

1. 我個人認為四個小時差不多，因為理論類的東西，不應當花太長的時間學習，可以另可把時間拿去學習實作類的東西。
2. ChatGPT：學習大學的資料結構，每週建議的時間分配如下：
 - 課堂時間：2~4 小時
 - 課後閱讀與複習：2~3 小時
 - 程式實作：3~5 小時
 - 作業或專案：2~4 小時
 - 預習與延伸學習：1~2 小時
 - 總計：每週 10~15 小時，若需要更深入學習或補強基礎，則延長至 15~20 小時。
 - 學習技巧：每天分散學習，避免集中熬夜。練習解題以鞏固概念。善用資源與求助渠道（教授、助教、線上社群）。這樣規劃能有效掌握資料結構的核心內容和實作技能！

3. 我認為時間上有差異就是GPT把課堂的時間算進去，以及而外算到實作的部分，當然GPT給的時間我覺得過於長了，我是覺得可以縮短複習的時間為1小時，還有預習的時間為40分鐘，總計大概一周如果算進實作的話大概就是8~10小時的學習時間。

個人認為完成作業須具備觀念

本次練習的重點在於熟悉 `Git` 指令的基本操作與概念，學會如何管理版本控制。需掌握的指令包括 `git add` 用於將檔案加入暫存區、`git commit` 用於提交暫存區內容到本地倉庫、`git push` 用於將提交的內容推送至遠端倉庫。此外，需理解 `git fetch` 與 `git pull` 的差異與用途，學會分支操作如 `git branch`、`git checkout`、`git merge`，並能將理論應用於實際版本管理場景。