

Guida per Cam: Github e VSC

1. Preliminari

1. 1. Cos'è Github?
1. 2. Cos'è git?
 1. 2. 1. Come si fanno questi salvataggi?
1. 3. Cos'è un IDE?
 1. 3. 1. Perché non Google Collab?

2. Guida all'installazione

2. 1. Github
2. 2. Visual Studio Code

3. Utilizzo

3. 1. Condividermi le tue modifiche
 3. 1. 1. Come fare il commit su VSC
 3. 1. 2. Come pusharlo su Github
3. 2. Scaricare le mie modifiche

Ciao Cam! Ho scritto questa guida semplice semplice per poter imbastire il progetto. Forse alcune cose le sai già, ma non sapendolo ti ho scritto in teoria tutto quello che dovrebbe servirci. Se hai qualche dubbio scrivimi!

1. Preliminari

1.1. Cos'è Github?

- è una specie di Google Drive dove le persone possono lavorare insieme ai progetti informatici
- Github per funzionare è basato su **git**
- la differenza è che, mentre git è in locale sul tuo pc, github è "online" e quindi condivisibile tra collaboratori

1.2. Cos'è git?

- è uno strumento per salvare degli "snapshot" del tuo progetto
- è un po' come nei videogiochi, in cui fai un salvataggio nuovo ogni volta e puoi ritornare ai vecchi salvataggi quando vuoi
- è uno strumento molto utile se a un certo punto il progetto impazzisce e devi tornare indietro
- quindi github sincronizza questi salvataggi di git in una cartella "online" che si può scaricare e aggiornare

1.2.1. Come si fanno questi salvataggi?

- ogni salvataggio è un **commit**
- i commit li deve fare il programmatore "manualmente" ogni volta che sente di

- aver fatto una modifica
- esempio: ho scritto un nuovo metodo, o una nuova classe, oppure ho eliminato degli import inutili, o delle funzioni inutilizzate... e così via
 - per fare un commit si può:
 - o usare la linea di comando da **terminale** → un po' scomodo ma a volte è l'unica
 - oppure usare l'IDE

1.3. Cos'è un IDE?

- IDE = Ambiente di sviluppo integrato
- è un software che permette di sviluppare in un linguaggio di programmazione
- usare un IDE per programmare è tipo usare Word per scrivere la tesi
- gli IDE ti permettono di eseguire il tuo codice senza troppi casini di configurazione file
- quando usi Google Collab in qualche modo stai usando un IDE
- ci sono un po' di IDE che però permettono di fare tutti le stesse cose, tra queste quelle che interessano a noi sono:
 - scrivere in Python (`.py`)
 - scrivere Jupiter Notebook (`.ipynb`)
 - organizzare i file del progetto
 - fare commit
 - eseguire il codice
 - scaricare librerie
- noi useremo un IDE gratis che si chiama **Visual Studio Code**

1.3.1. Perché non Google Collab?

- google collab serve per aprire Jupiter Notebook
- ma non permette di lavorare su progetti più grandi di un unico Jupiter Notebook
- a noi invece serve lavorare su più file per mantenere il codice organizzato

Cos'è un Jupiter Notebook

- se un file python è un file che contiene codice python...
- ...un jupiter notebook è un file che contiene python e markdown
- markdown è un modo di scrivere testo normale
- i notebook sono comodi perché ti permettono di eseguire singole parti di codice e di focalizzarti su un problema alla volta
- però in certi casi conviene usare un file python (per definire classi o per componenti più complesse)

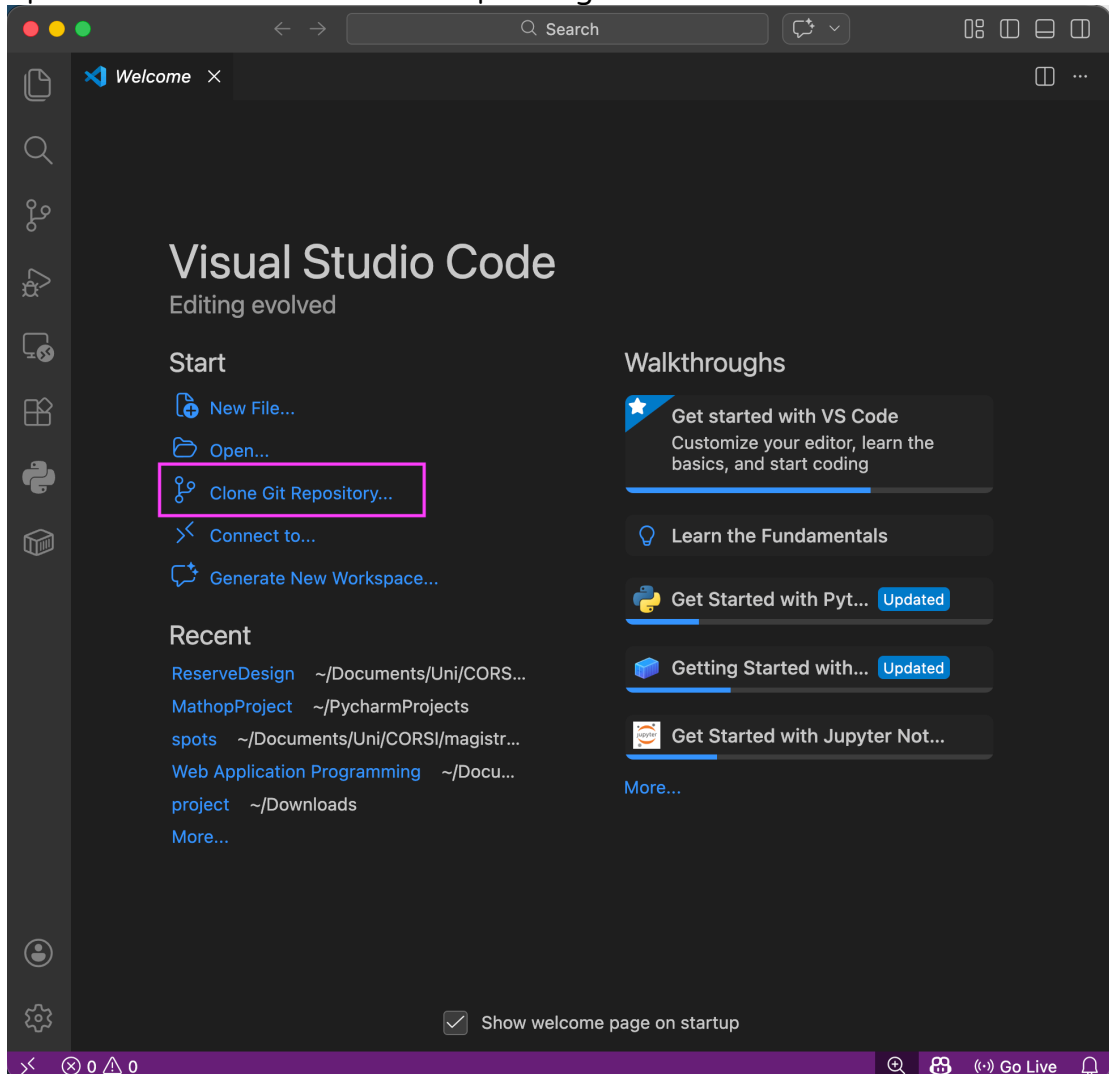
2. Guida all'installazione

2.1. Github

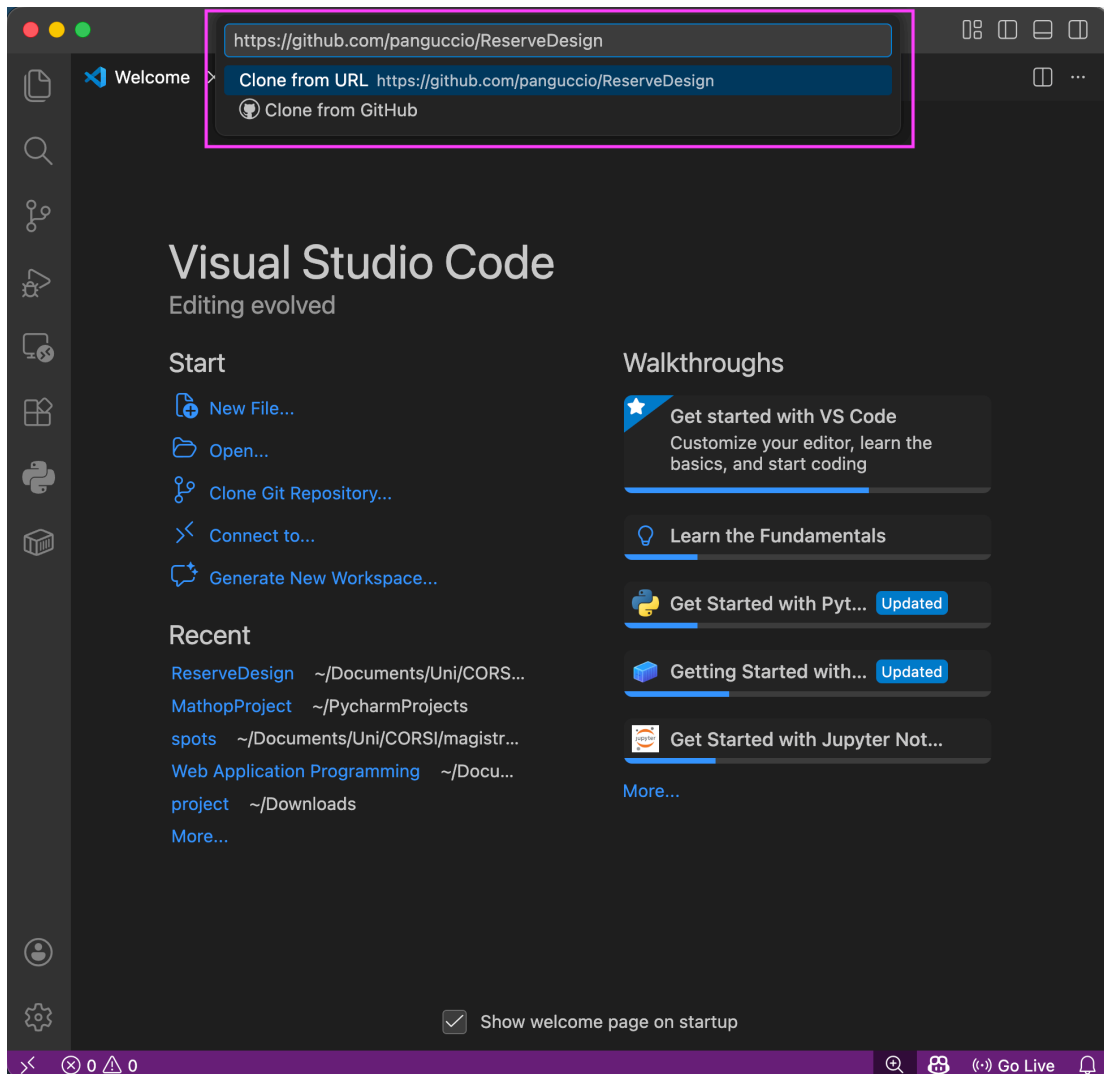
1. vai su <https://github.com>
2. fai l'account se non ce l'hai già

2.2. Visual Studio Code

1. vai su <https://code.visualstudio.com/download> e scarica VSC per il tuo sistema operativo
2. segui la procedura di installazione
3. apri VSC e seleziona "Clone Git Repository..."



4. scrivi <https://github.com/panguccio/ReserveDesign>



5. seleziona la cartella dove installare il progetto

6. apri il progetto

ora dovresti vedere tutti i file come sono su github, però sono sul tuo pc!

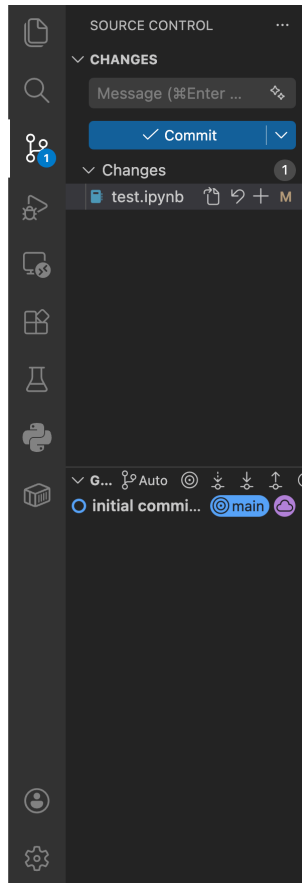
3. Utilizzo

3.1. Condividermi le tue modifiche

- adesso quello che tu modifichi nei file verrà modificato solo in locale
- è un po' come se tu avessi una tua versione del file
- per potermelo "mandare" devi semplicemente fare un **commit** e pusharlo su github

3.1.1. Come fare il commit su VSC

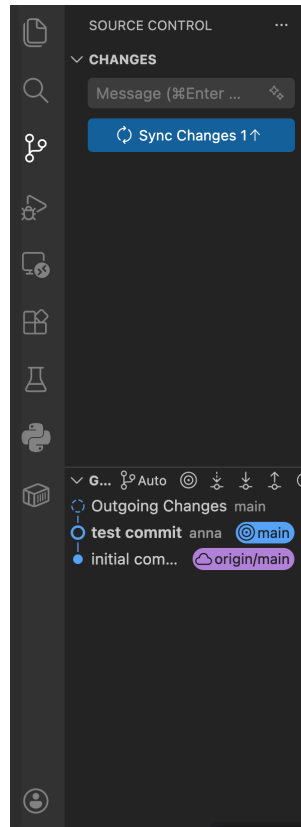
- tutte le modifiche le vedi in questo menù
 - nella parte sopra vedi le modifiche che hai fatto in quali file → per esempio nel mio caso ho modificato `test.ipynb`
 - nella parte sotto vedi lo storico dei commit in pratica i tuoi salvataggi → per esempio nel mio caso c'è solo il primo "initial commit"



- quando sei sicura di una mofica clicchi il tasto commit
 - questo equivale a fare un salvataggio con **git**
 - quindi è salvato solo sul tuo computer

3.1.2. Come pusharlo su Github

- una volta fatto questo, il commit comparirà nella parte sotto e verrà scritto "Sync Changes"



- se tu sei **sicurissima** del codice che hai scritto, puoi premere Sync Changes per condividerlo con me!

3.2. Scaricare le mie modifiche

- in teoria se io ho modificato i file, VSC dovrebbe avvisarti e chiederti se vuoi aggiornare con le modifiche in remoto
- se non abbiamo modificato gli stessi file dovrebbe andare tutto liscio
- se abbiamo modificato gli stessi file è un po' più complicato (bisognerebbe fare dei branch e poi dei merge)
- dato che non vogliamo le cose complicate cerchiamo di metterci d'accordo prima su cosa modificare
- se così non dovesse andare scriverò una parte 2 di questa guida :D