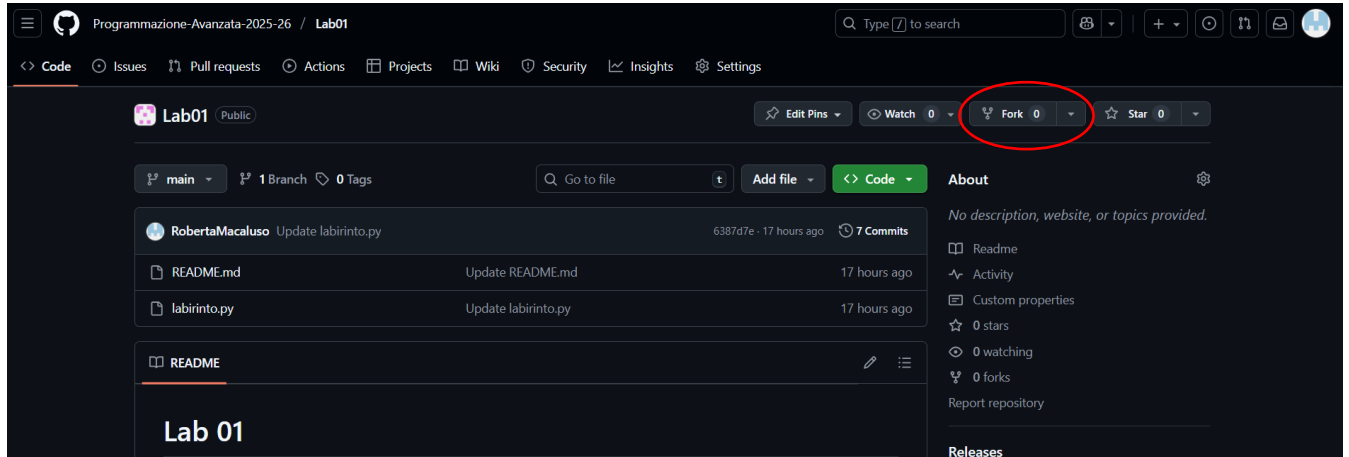


# GUIDA PER USARE GITHUB E PYCHARM

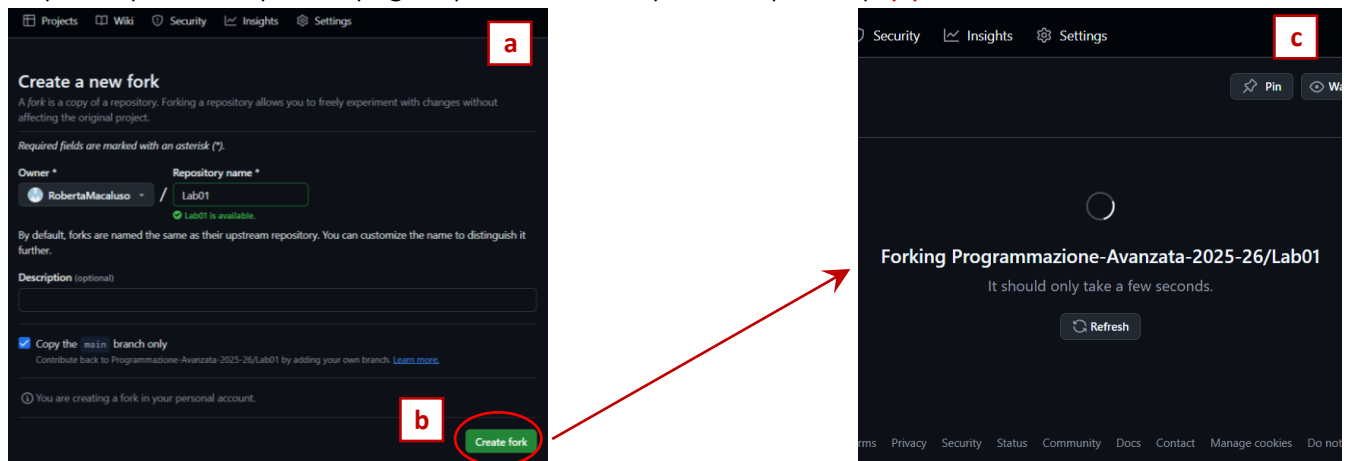
## FORK:

1. Effettuare il login su GitHub (<https://github.com/login>) utilizzando il proprio username e password.
2. Scaricare Git sul proprio PC dal link <https://git-scm.com/downloads>, se necessario.
3. Aprire su GitHub il repository sul quale si vuole lavorare (es. Lab01).
4. Utilizzare il pulsante *Fork* in alto a destra per creare una propria copia del progetto.



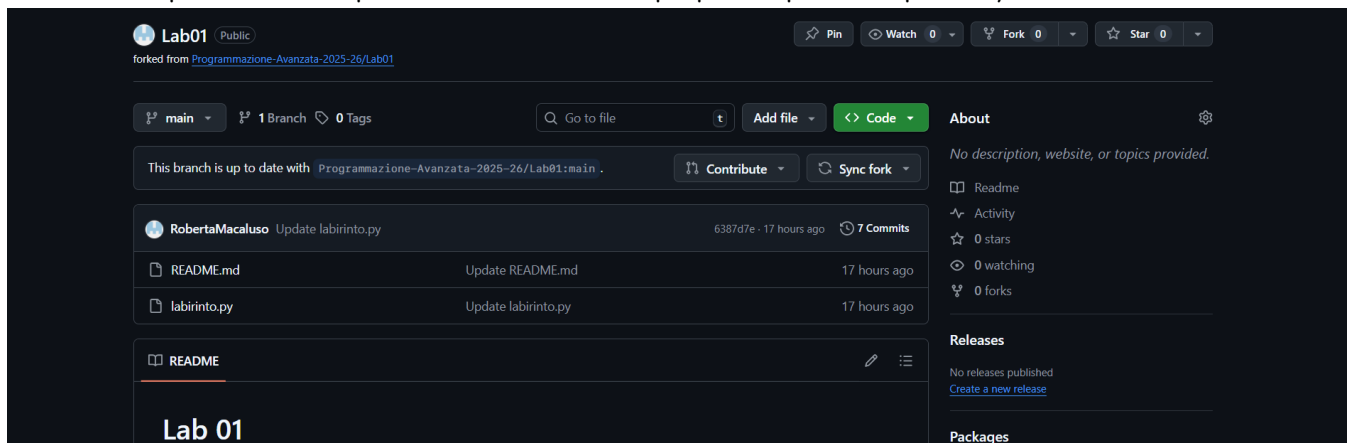
L'azione di Fork crea un nuovo repository nel proprio account GitHub con una copia dei file necessari per lo svolgimento del laboratorio.

5. A questo punto si aprirà la pagina per creare la copia del repository (a).

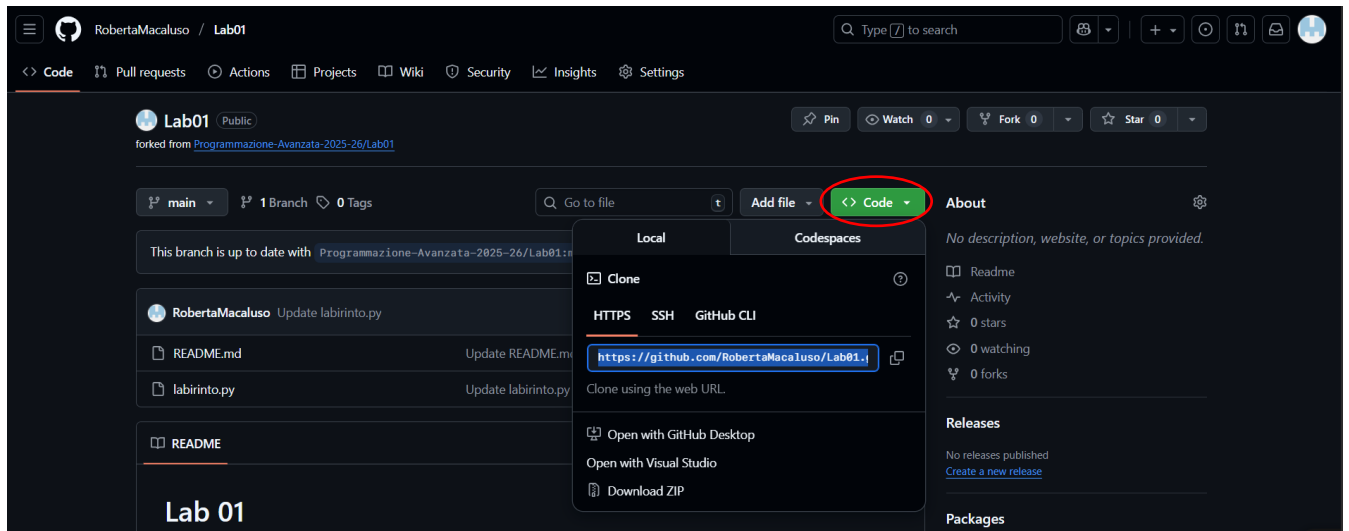


Non è necessario fare alcuna modifica: è possibile mantenere tutte le impostazioni predefinite e cliccare sul pulsante "**Create fork**" (b), quindi aspettare che la fork venga effettuata (c).

6. Terminato il processo sarà possibile accedere alla propria copia del repository.



7. Premere il pulsante "<> **Code**" e copiare l'URL evidenziato. Ciò servirà per clonare la propria copia del repository su Pycharm (non quello di *Programmazione-Avanzata-2025-26*)



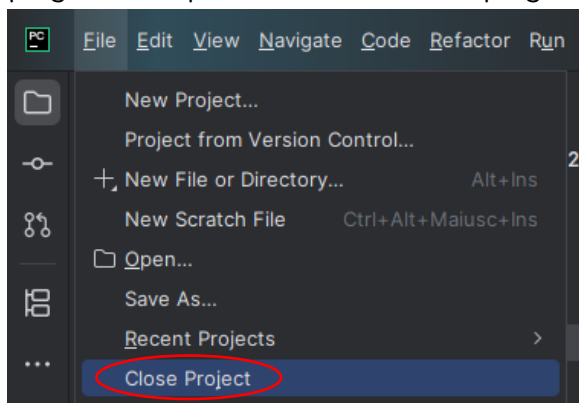
**ATTENZIONE:** Prima di copiare l'URL, verificare sempre di essere nel fork personale. Come mostrato nell'immagine, l'URL del repository creato per il fork personale avrà questo formato (lab01 nell'esempio): [https://github.com/Username\\_del\\_mio\\_GITHUB/Lab01.git](https://github.com/Username_del_mio_GITHUB/Lab01.git)

Invece, l'URL del repository principale è il seguente e **NON VA MAI USATO**:

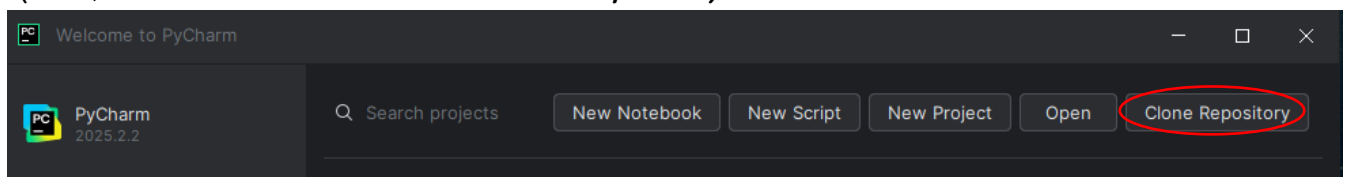
<https://github.com/Programmazione-Avanzata-2025-26/Lab01.git>

## PYCHARM:

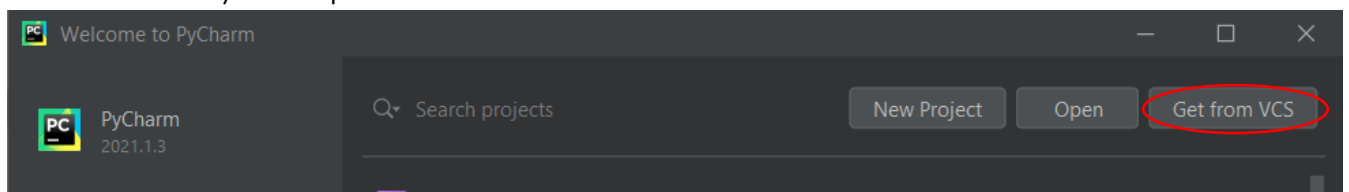
8. Scaricare e aprire PyCharm (versione Community 2025.2.2 consigliata) e, assicurarsi che nessun progetto sia aperto. Per chiudere un progetto selezionare **File** → "Close Project"



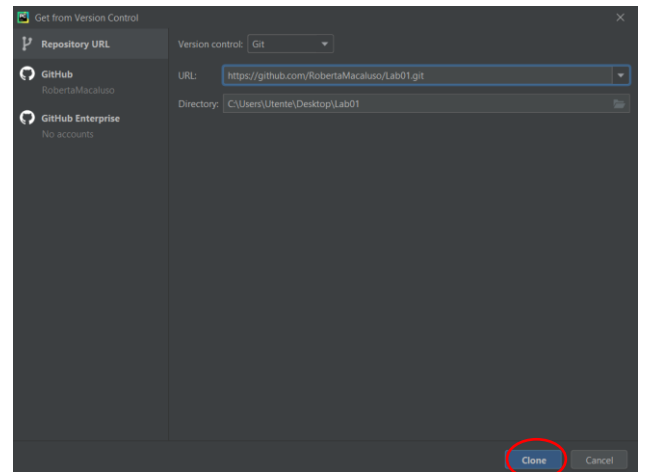
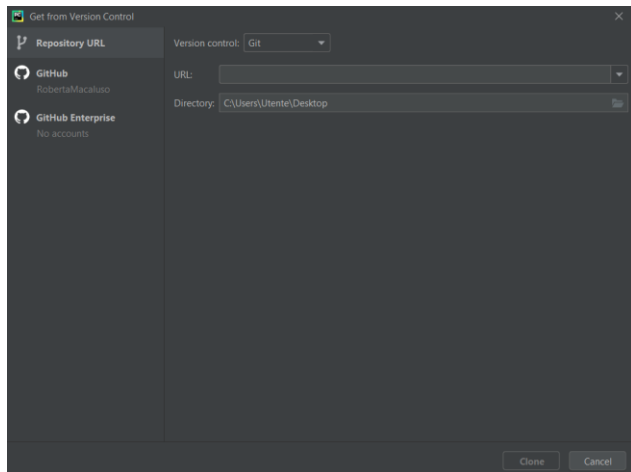
9. Quindi, selezionare nel menu in alto "**Clone Repository**"



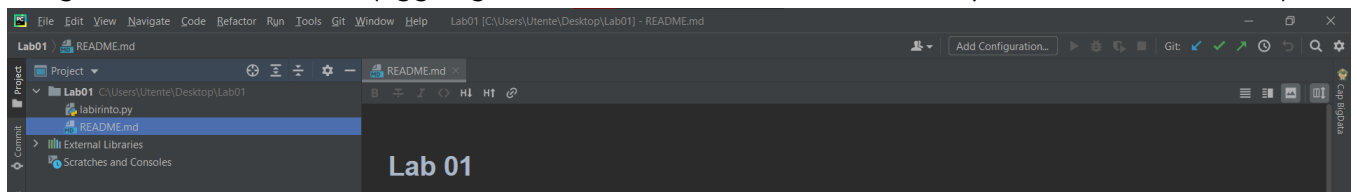
10. Per versioni di PyCharm precedenti alla 2025 è necessario selezionare "**Get from VCS**"



11. Si aprirà il seguente pannello, in cui si dovrà inserire l'URL copiato precedentemente: ad esempio [https://github.com/Username\\_del\\_mio\\_GITHUB/Lab01.git](https://github.com/Username_del_mio_GITHUB/Lab01.git) quindi, decidere in quale directory salvare il progetto sul PC personale (va bene anche quella che appare di default) e premere **"Clone"**.

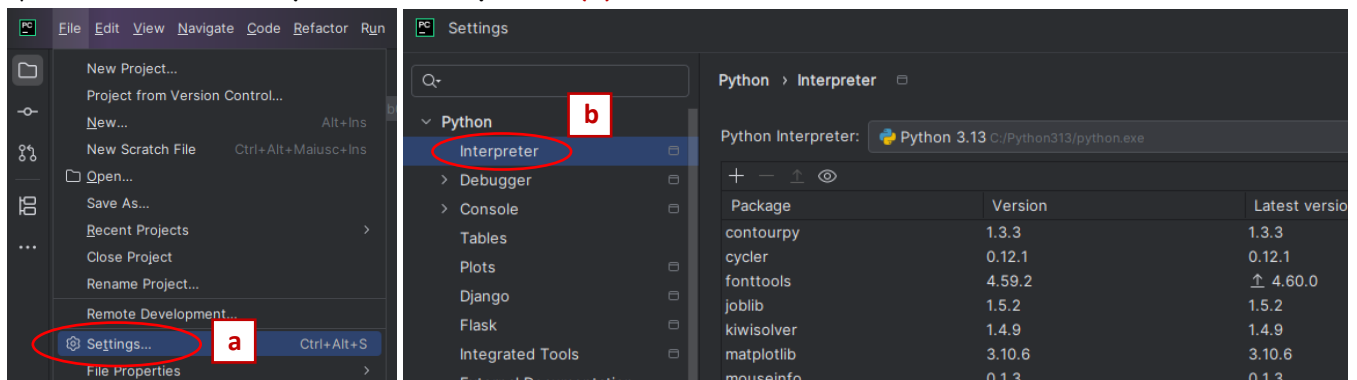


12. Dopo aver effettuato il login su GitHub con le proprie credenziali, un nuovo progetto verrà creato con tutti i file del repository al suo interno. Ora si potranno effettuare tutte le modifiche necessarie allo svolgimento del laboratorio (aggiungere nuovi file, modificare o eliminare quelli esistenti, e così via).

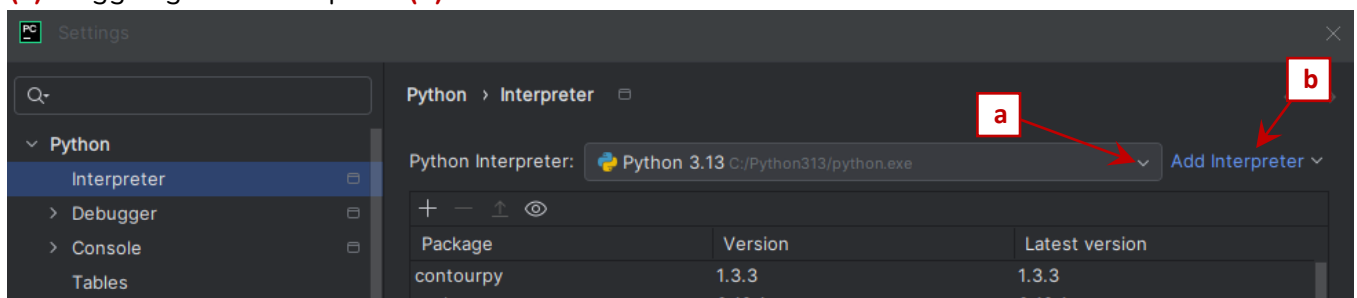


## SETTARE L'INTERPRETE PYTHON:

13. Per settare l'interprete, ove necessario, basta andare in **File → Settings (a)**, quindi selezionare **"Python → Interpreter" (b)**.

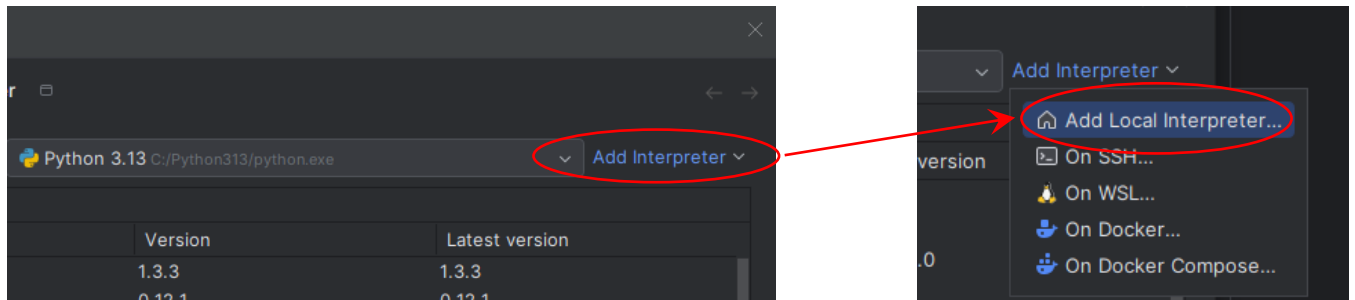


14. A questo punto è necessario scegliere (aprendo il menu a tendina si possono vedere quelli già installati) **(a)** o aggiungere un interprete **(b)**.

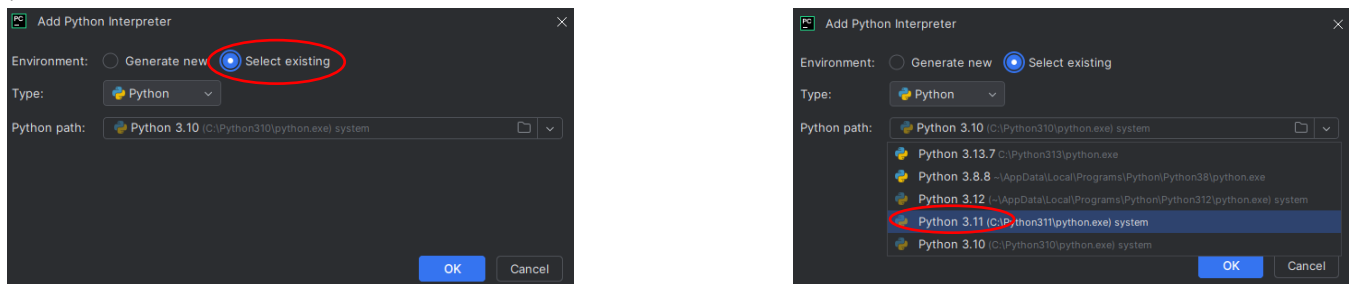


La versione utilizzata in laboratorio è la **3.11**.

15. Se una determinata versione non fosse già configurata, sarà possibile aggiungerla selezionando “**Add Interpreter**”, quindi, “**Add Local Interpreter**”.



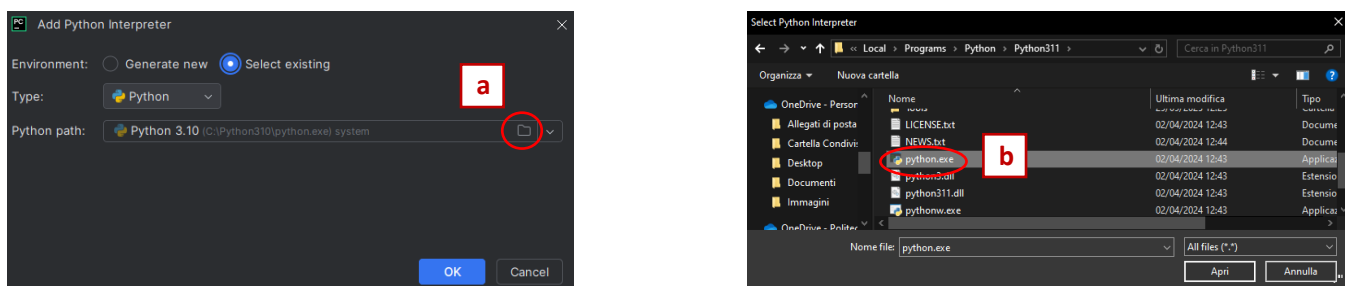
16. A questo punto saranno possibili varie opzioni: creare un virtual environment, usare un interprete di sistema, ecc. La scelta consigliata, sui PC dei lab, è quella di usare l'interprete di sistema con la versione 3.11. Quindi, selezionare “**Select Existing**”, aprire il menu a tendina e selezionare una versione  $\geq 3.11$  e premere “**OK**”.



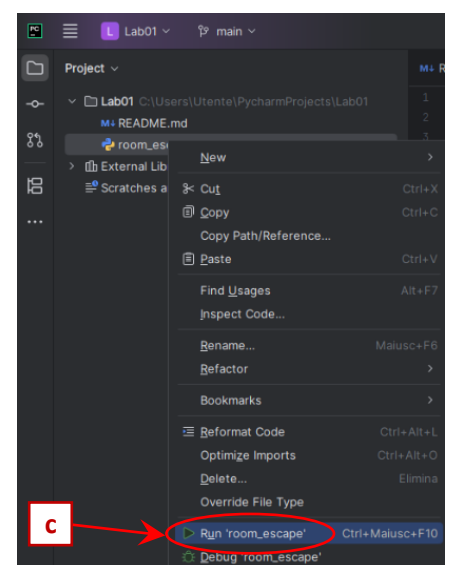
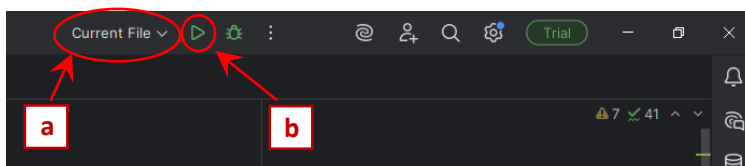
17. Nei PC personali, per la prima configurazione, potrebbe essere necessario cercare l'interprete installato indicando il path. Selezionare l'icona (a), quindi cercare il file python.exe della versione di python che si vuole usare (ad esempio la 3.11) (b). Solitamente il path di default è qualcosa di questo tipo:

**C:\Users\<TUO\_UTENTE>\AppData\Local\Programs\Python\Python3X\python.exe**

Dove **Python3X**, nel caso della 3.11, sarà **Python311**.

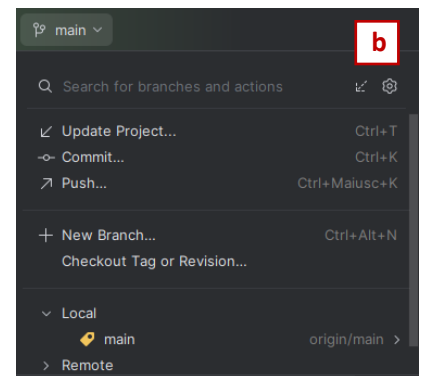
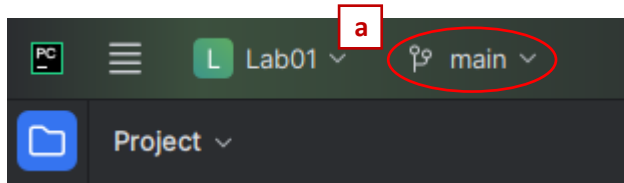


18. A questo punto, è possibile impostare su “**Current File**” il menu di esecuzione (a), quindi selezionare “**Run**” per eseguire il file selezionato (b). È anche possibile eseguire un file facendo tasto destro sul file da eseguire quindi “**Run 'nome\_file'**” (c).

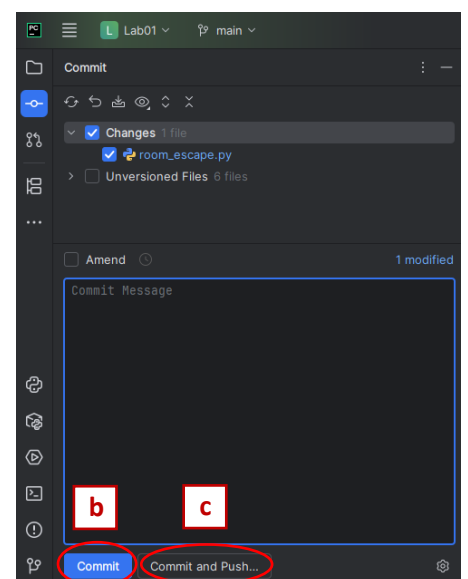
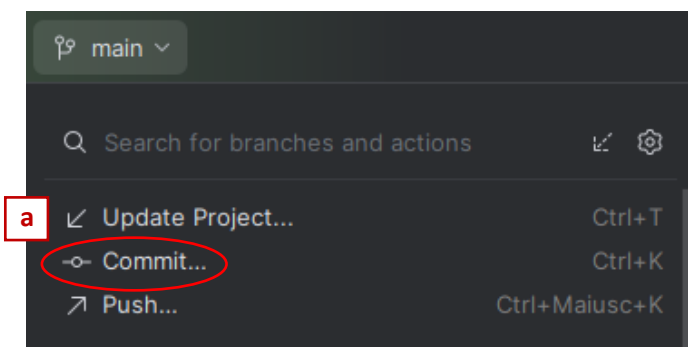


## COMMIT E PUSH:

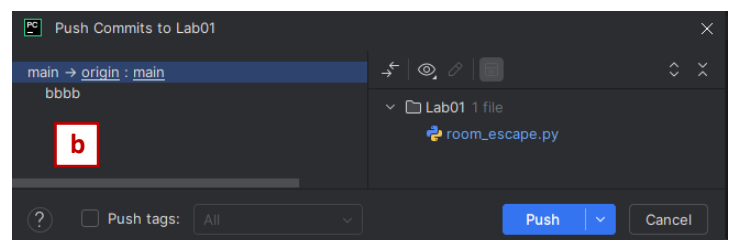
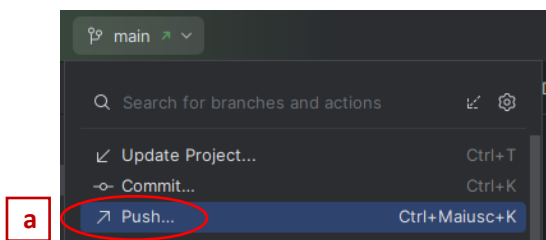
19. A fine lavoro è necessario effettuare il **commit** e il **push** usando l'apposito menu. Cliccando su “**main**” (a) si aprirà un menu che permette di effettuare varie azioni (update, commit, push, ecc) (b).



20. Per effettuare il commit selezionare “**Commit**” (a), questo aprirà un secondo pannello che mostra i file modificati e permette di aggiungere un commento al commit (se i file aggiunti non appaiono controllare dentro “**Unversioned Files**”, quindi selezionarli). Dopo aver controllato di star effettuando il commit in modo corretto, selezionare il pulsante “**Commit**” (b), oppure direttamente “**Commit and Push**” (c).



Solo dopo aver effettuato il commit, se non già fatto prima, bisogna effettuare anche il **push** (a) usando l'apposito menu (b).



**ATTENZIONE:** Solo se si effettua il **commit** e poi il **push** le modifiche locali saranno propagate sui server GitHub e saranno quindi accessibili da altri PC e dagli altri utenti che ne hanno visibilità. Quindi eventuali commit oltre la DEADLINE indicata per ogni lab non saranno presi in considerazione.

21. Se appare questa schermata, andare su “**Log In via GitHub**” per autorizzare JetBrains.

