

Git e GitHub

Git e GitHub sono molto simili, la cosa più grande che cambia è che Git è una risorsa locale su rete locale, mentre GitHub è una risorsa globale servita da Microsoft dove può avere l'accesso ovunque nel mondo.

Utilizzo

Download Git sul pc: per verificare di averlo scaricato andare su terminale di windows ed eseguire

```
git --version
```

Creazione del account **GitHub**

Comandi per la creazione di un repository

Inizializzare git nella cartella selezionata (verrà creata una cartella nascosta chiamata git) eseguendo

```
git init
```

In questo modo è stata creato un repository in locale

Se si vuole inserire delle variabili globali in git si può procedere con

```
git config --global user.name "<nome utente>"
```

```
git config --global user.email "<nome email>"
```

Creazione del repository su GitHub per fare in modo che avvenga una connessione remota tra git e GitHub, quindi eseguire tutti i comandi che ci sono da fare su GitHub.

Copiare il link https del repository e collegarlo al nostro git, quindi eseguire il comando

```
git remote add origin <link>
```

Anche se sembra che non sia successo nulla ma in realtà è avvenuto il collegamento tra il nostro progetto e il repository appena creato

Operazione di PULL

Per vedere se il collegamento è riuscito basta effettuare l'operazione di **pull**, cioè consiste di fare una richiesta a GitHub per accedere ai file del repository e salvarli all'interno della nostra cartella di progetto (pullare i file nella cartella di progetto). Il comando è il seguente:

```
git pull origin <nome-branch>
```

Status

Per controllare lo stato dei file all'interno del progetto basta eseguire

```
git status
```

In questo modo si visualizza quali file sono stati modificati.

Branch

Se non si conosce il **nome-branch** basta eseguire il comando

```
git branch
```

Dopo di questo si può notare che nella cartella di progetto si ha tutto quello che ci sia nel repository.

Per creare un nuovo branch basta eseguire il comando

```
git branch <nome-del-nuovo-branch>
```

Per spostarsi successivamente sul nuovo branch basta eseguire il comando

```
git checkout <nome-del-nuovo-branch>
```

Per crearlo e spostarsi con un unico comando basta eseguire

```
git checkout -b nuovo-branch
```

Operazione di PUSH

Per aggiungere i file a git si può fare :

```
git add . -> per aggiungere tutti i file insieme a git
```

```
git add <nome-file> -> per aggiungere solo un file a git
```

Aggiunti i file bisogna commissionarli, quindi pronti per la spedizione su git eseguendo

```
git commit
```

Si aggiunge al comando il **-m** per inserire un messaggio all'invio.

L'operazione di **push** è l'inversa di quella di pull, quindi i file della cartella di progetto verranno copiati ed inseriti nel repository, con le varie commit, con i vari aggiornamenti

```
git push origin <nome-branch>
```

Operazione di CLONE

Il comando **clone** serve per la portabilità del codice, che permette di clonare i repository; quindi, serve a clonarti il repository in una nuova cartella facendo

```
git clone <link>
```

Riassunto comandi esame

La sequenza di comandi da fare all'esame è la seguente

```
git clone <link>
```

```
git branch -> per verificare su quale branch ci troviamo
```

```
git branch <numero di matricola>
```

```
git checkout <numero di matricola>
```

```
git branch -> per verificare su quale branch ci troviamo
```

Iniziare a fare il programma e tutto quello che si deve fare, aggiungendo e commissionandolo per ogni punto da fare

```
git add .....
```

```
git commit -m .....
```

Al termine della prova si apriranno le porte della spedizione, quindi eseguire

```
git push origin <numero di matricola>
```