GITHUB

Github è un servizio hosting (un servizio di rete che consiste nell’allocare su un server web le pagine di un sito web, rendendolo così accessibile dalla rete internet e ai suoi clienti) per progetti software. Il sito ha funzionalità simili ad un social network come feeds, follower e grafici per vedere come gli sviluppatori lavorano sulle varie versioni dei repository.

Un progetto **GIT** è strutturato in 3 parti. Il primo è la directory di lavoro che contiene i file attuali sui quali l’utente sta lavorando. Il secondo è l’index che fa da spazio di transito per i file (sono i file candidati per essere committati) e per finire l’head che punta all’ultimo commit fatto (contiene gli ultimi commit fatti).

****

**Inizializzare un repository git**

*Cosa è un repository?*

Il **repository** di solito viene usato per organizzare un singolo progetto. Il repository può contenere file, immagini, video, un insieme di dati.

***Inizializzare un repository git***

***git init*** permette di inizializzare un progetto esistente. Bisogna andare nella directory del progetto e poi digitare il comando.

*git status*  permette di visualizzare lo status dei file all’interno del repository locale ( restituisce informazioni riguardanti qualsiasi file non tracciato/modificato/in staging , lo stato del branch e molto altro.) È una buona idea usare il comando git status prima di git add.

*git add* permette di spostare uno o più file nell’index (o stage).Cioè si aggiungo delle modifiche.

***git add --all***

***git commit -m “messaggio per la commit”*** valido le modifiche

inserendo anche il messaggio per il quale si è commitato. Permette di spostare uno o più file nell’head.

Esempio

$ git init

Initialized empty Git repository in ~/djangogirls/.git/

$ git config --global user.name "Your Name"

$ git config --global user.email you@example.com

$ git status

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

blog/

manage.py

mysite/

nothing added to commit but untracked files present (use "git add" to track)

$ git add --all .

$ git commit -m "La mia app Django Girls, primo commit"

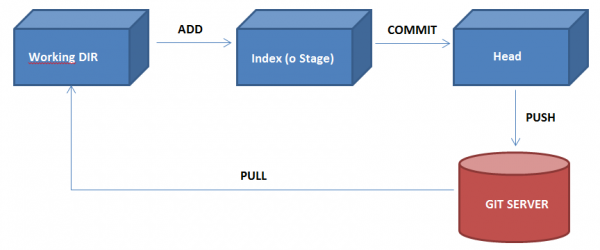
[...]

13 files changed, 200 insertions(+)

create mode 100644 .gitignore

[...]

create mode 100644 mysite/wsgi.py

******

[***Ignorare File***](https://git-scm.com/book/it/v1/Basi-di-Git-Salvare-le-modifiche-sul-repository#Ignorare-File)

Spesso hai dei file che non vuoi che Git aggiunga automaticamente e nemmeno che te li mostri come tracciati. Generalmente si tratta di file generati automaticamente, come i log o quelli prodotti dal tuoi sistema di build. In questi casi puoi creare un file chiamato .gitignore con la lista di pattern dei file che vuoi ignorare

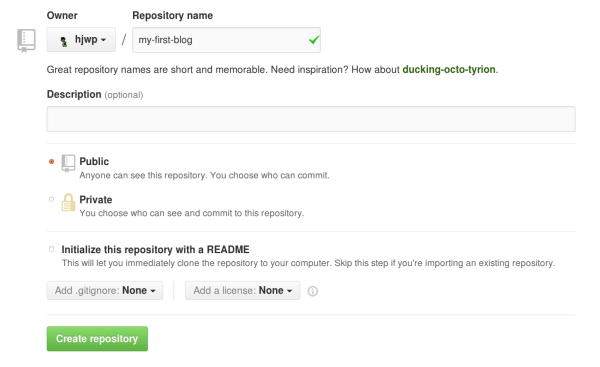
## Pubblichiamo il nostro codice su GitHub

Nell’angolo in alto a destra, accanto alla tua foto, clicca su + e seleziona new repository.

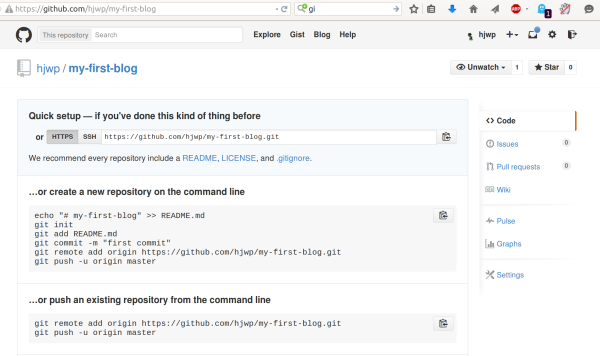
- Dai un nome al tuo repository.

- Scrivi una piccola descrizione.

- Seleziona Initialize this repository with a README.

******

Nella schermata successiva, ti verrà mostrato l'URL per clonare il tuo repository:

******

Ora abbiamo bisogno di collegare il repository Git sul tuo computer a quello su GitHub. I comandi che usiamo sono ***git remote add origin “url”(connette il tuo repository al server remoto) e git push –u origin master (***permette di inviare uno o più file al repository remoto).

Esempio

$ git remote add origin https://github.com/<your-github-username>/my-first-blog.git

$ git push -u origin master

Inserisci il tuo username e la tua password di GitHub. Dovresti vedere qualcosa di simile:

Username for 'https://github.com': hjwp

Password for 'https://hjwp@github.com':

Counting objects: 6, done.

Writing objects: 100% (6/6), 200 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/hjwp/my-first-blog.git

\* [new branch] master -> master

Branch master set up to track remote branch master from origin.

Adesso Il tuo codice è su GitHub.