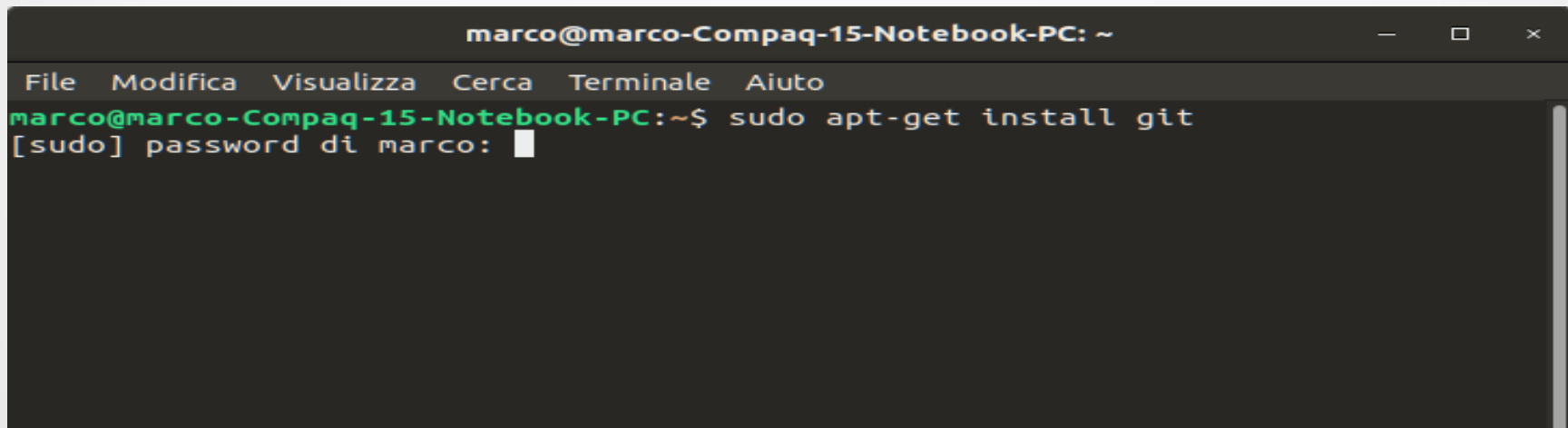


Git Tutorial da Terminale

INSTALLAZIONE:

- Apriamo il nostro terminale e procediamo utilizzando il comando:

A screenshot of a terminal window titled "marco@marco-Compaq-15-Notebook-PC: ~". The window has a menu bar with "File", "Modifica", "Visualizza", "Cerca", "Terminale", and "Aiuto". The terminal shows the command "sudo apt-get install git" being entered. Below it, the prompt "[sudo] password di marco:" is displayed with a cursor. The terminal background is dark, and the text is light green and white.

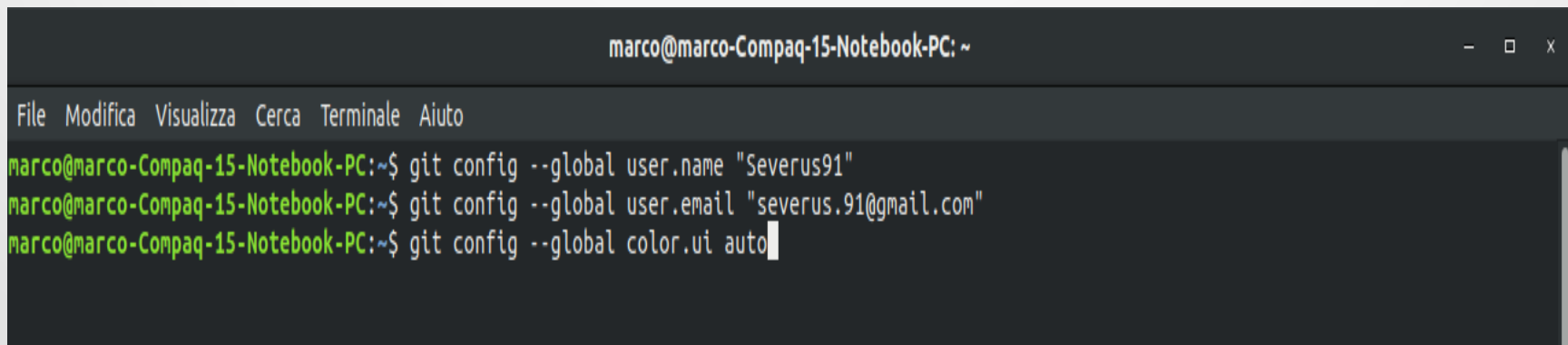
```
marco@marco-Compaq-15-Notebook-PC: ~  
File  Modifica  Visualizza  Cerca  Terminale  Aiuto  
marco@marco-Compaq-15-Notebook-PC:~$ sudo apt-get install git  
[sudo] password di marco: 
```

*Con l'utilizzo del comando "sudo", ubuntu per ottenere i permessi di ROOT ci richiederà la nostra password utente. Inseriamola, a questo punto l'installazione è partita, dopo alcuni secondi ci verrà chiesto: "Continuare? [S/n]" digitiamo una "s" e premiamo invio. Attendiamo il completamento dell'installazione

Git Tutorial da Terminale

- I seguenti comandi hanno la potenzialità di fare interagire git con il nostro profilo github:

- 1) `git config --global user.name "Severus91"`
configura il nome su cui congiungere le transazioni
- 2) `git config --global user.email "severus.91@gmail.com"`
configura l'e-mail su cui congiungere le transazioni
- 3) `git config --global color.ui auto`
abilita la colorazione d'aiuto della linea di comando in output



```
marco@marco-Compaq-15-Notebook-PC: ~  
File Modifica Visualizza Cerca Terminale Aiuto  
marco@marco-Compaq-15-Notebook-PC:~$ git config --global user.name "Severus91"  
marco@marco-Compaq-15-Notebook-PC:~$ git config --global user.email "severus.91@gmail.com"  
marco@marco-Compaq-15-Notebook-PC:~$ git config --global color.ui auto
```

Git Tutorial da Terminale

Esploriamo i comandi d'aiuto:

- Notebook-PC:~\$ **git**

mostra la lista di comandi base che interagiscono dal terminale con git

- Ed i principali comandi per utilizzarlo

- Notebook-PC:~\$ **git help -a**

mostra una lista di tutti i comandi utilizzabili

- Notebook-PC:~\$ **git help nome comando**

mostra una guida dedicata al comando specificato

- Notebook-PC:~\$ **git help -g**

- # mostra una lista delle guide concettuali di git

- Notebook-PC:~\$ **git help nome guida concettuale**

- # mostra la documentazione relativa alla guida concettuale indicata

•

- *In questa pagina troviamo una guida riassuntiva dei principali comandi:

- <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>

Git Tutorial da Terminale

Iniziamo ad utilizzare git:

- Partiamo con la creazione di una nuova directory on-line, accediamo al nostro account github, clicchiamo sull'icona contrassegnata col segno “+” in alto a destra e selezioniamo “new repository”; si aprirà una pagina in cui dovremo dare un nome alla nostra repository (e un eventuale descrizione), ricordandoci di selezionare le spunte “Public”* e “initialize a repository with a README”**.

*Per poter selezionare “Private” dobbiamo aver sottoscritto un abbonamento premium a pagamento (\$7 / month).

** Le repository devono contenere questo file iniziale per essere concrete. È possibile anche crearlo tramite terminale, attraverso un procedimento un po' più dispendioso in termini di tempo, per questo è consigliabile la spunta iniziale, attraverso i seguenti comandi:

```
echo "# nome_repository" >> README.md
```

```
git init
```

```
git add README.md
```

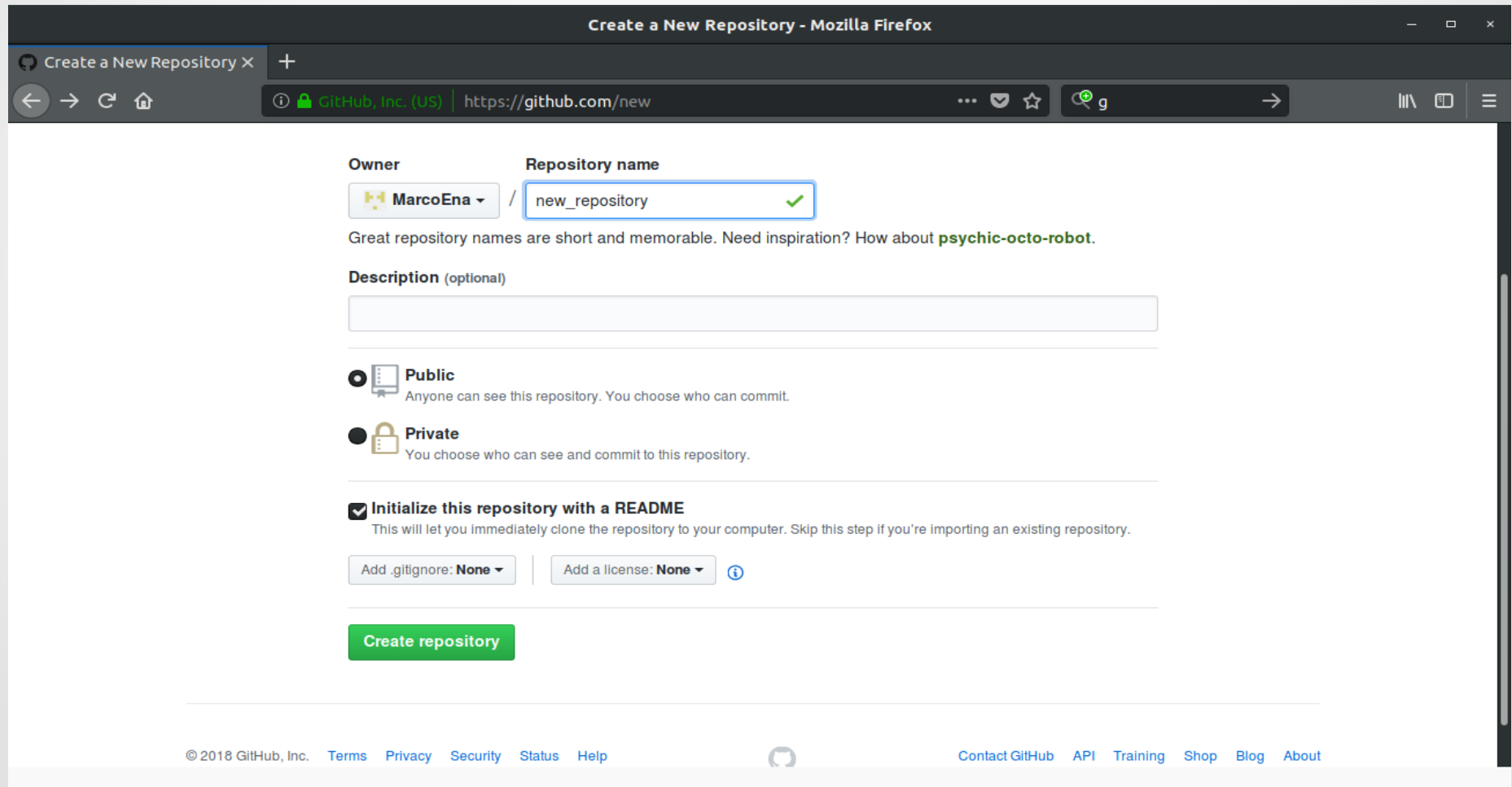
```
git commit -m "first commit"
```

```
git remote add origin https://github.com/MarcoEna/my_first_repository.git
```

```
git push -u origin master
```

Git Tutorial da Terminale

- Immagine di riferimento:



The screenshot shows the 'Create a New Repository' page on GitHub, viewed in Mozilla Firefox. The browser's address bar shows the URL 'https://github.com/new'. The page has a dark header with the title 'Create a New Repository - Mozilla Firefox'. Below the header, the 'Owner' is set to 'MarcoEna' and the 'Repository name' is 'new_repository', which is highlighted with a blue border and a green checkmark. A message below the name field suggests 'Great repository names are short and memorable. Need inspiration? How about **psychic-octo-robot**.' The 'Description (optional)' field is empty. Under the 'Visibility' section, 'Public' is selected with a radio button, and a description says 'Anyone can see this repository. You choose who can commit.' The 'Private' option is also visible with a description 'You choose who can see and commit to this repository.' The 'Initialize this repository with a README' checkbox is checked, with a note: 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' Below this, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. At the bottom of the form is a green 'Create repository' button. The footer of the page includes copyright information '© 2018 GitHub, Inc.', links for 'Terms', 'Privacy', 'Security', 'Status', and 'Help', the GitHub logo, and links for 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

Create a New Repository - Mozilla Firefox

Create a New Repository × +

GitHub, Inc. (US) | https://github.com/new

Owner: MarcoEna / Repository name: new_repository ✓

Great repository names are short and memorable. Need inspiration? How about **psychic-octo-robot**.

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☒ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

Create repository

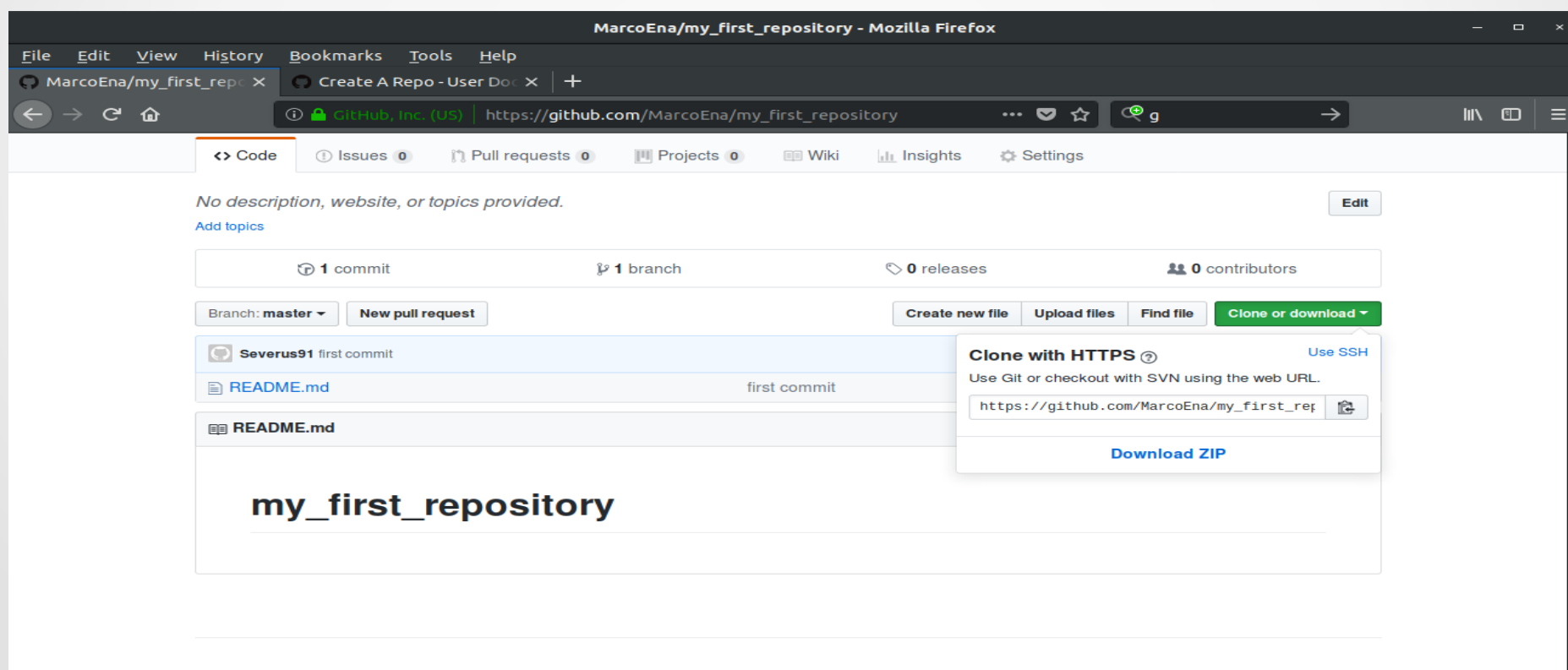
© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

Git Tutorial da Terminale

- Per sincronizzare git da remoto a locale e importare quindi la nostra repository utilizziamo il seguente comando:

Notebook-PC:~\$ **git clone https://github.com/MarcoEna/nome_cartella.git**



*Per trovare l'url osservare l'immagine sopra

Git Tutorial da Terminale

Per inizializzare una cartella da locale possiamo usare il comando:

```
Notebook-PC:~$ git init Nome_cartella
```

Se volessimo inizializzarla su una repository già esistente entriamo precedentemente su di essa con il comando:

```
Notebook-PC:~$ cd Nome_repository
```

Una volta eseguito procediamo con:

```
Notebook-PC:~/Nome_repository$ git init Nome_cartella
```

Eseguiamo, e ci verrà restituito il seguente output:

```
Initialized empty Git repository in  
/home/Utente/Nome_repository/nome_cartella/.git/
```

Git Tutorial da Terminale

Procediamo con la creazione di un nuovo branch:

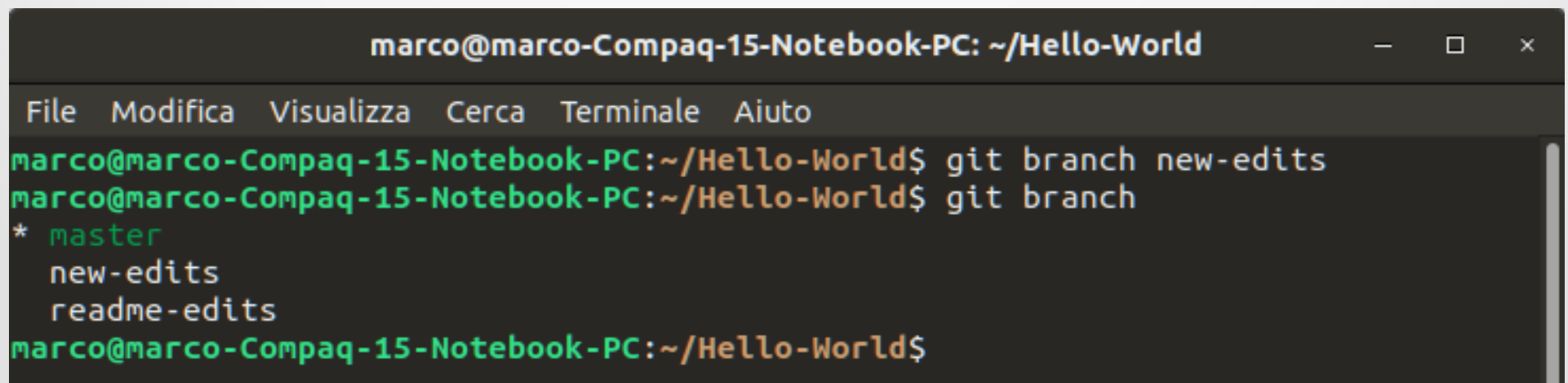
- Dopo essere entrati nella nostra repository inseriamo il comando:

Notebook-PC:~/Hello-World\$ **git branch nome-branch**

- Il secondo comando ci mostra i branch presenti sulla nostra repository

Notebook-PC:~/Hello-World\$ **git branch**

evidenziandoci in verde su quale branch ci troviamo, in questo caso siamo su quello di default il master

A screenshot of a terminal window titled "marco@marco-Compaq-15-Notebook-PC: ~/Hello-World". The window has a menu bar with "File", "Modifica", "Visualizza", "Cerca", "Terminale", and "Aiuto". The terminal shows two commands being executed: "git branch new-edits" and "git branch". The output of the second command lists three branches: "master" (highlighted in green), "new-edits", and "readme-edits". The prompt "marco@marco-Compaq-15-Notebook-PC:~/Hello-World\$" is visible at the bottom.

```
marco@marco-Compaq-15-Notebook-PC: ~/Hello-World
File  Modifica  Visualizza  Cerca  Terminale  Aiuto
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git branch new-edits
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git branch
* master
  new-edits
  readme-edits
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$
```

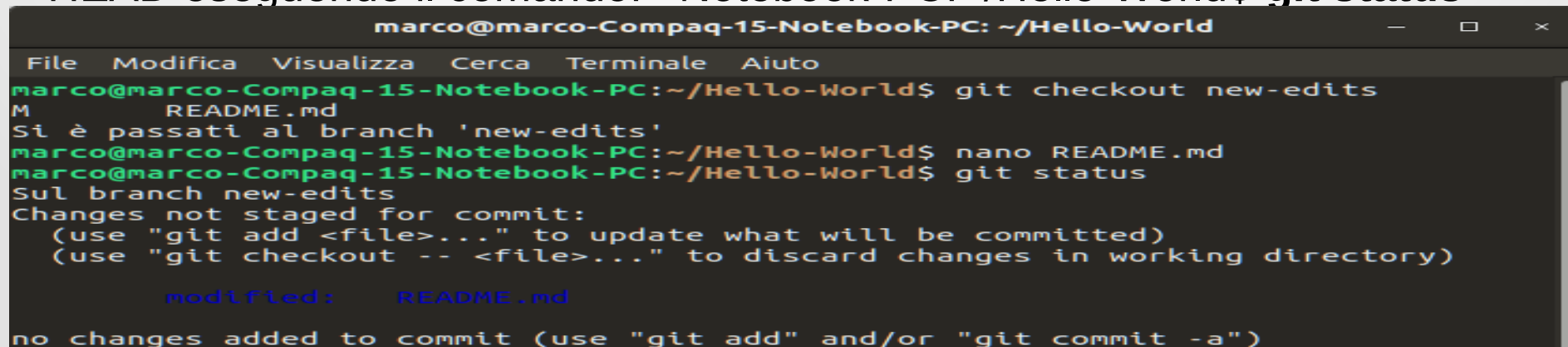

Git Tutorial da Terminale

Inoltriamoci nella comprensione delle potenzialità di git :

- Spostiamoci sul branch appena creato digitando:

Notebook-PC:~/Hello-World\$ **git checkout nome-branch**

- Apriamo un file con un editor di testo per esempio “**nano**” in questo caso “**README.md**” che avevamo creato in precedenza e modifichiamolo
- Essendo git basato su un sistema di “versioning”, la modifica del file non è mai definitiva finché non la salviamo (e attraverso l'utilizzo delle branch possiamo mantenere traccia di ogni cambiamento o passaggio anche dopo il salvataggio). Il file modificato si trova infatti sulla nostra “HEAD” una sorta di memoria temporanea. Mostriamo tutto quello che si trova sulla nostra HEAD eseguendo il comando: Notebook-PC:~/Hello-World\$ **git status**



```
marco@marco-Compaq-15-Notebook-PC: ~/Hello-World
File Modifica Visualizza Cerca Terminale Aiuto
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git checkout new-edits
M      README.md
Si è passati al branch 'new-edits'
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ nano README.md
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git status
Sul branch new-edits
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Git Tutorial da Terminale

Andiamo a salvare le modifiche del file con il comando:

- Notebook-PC:~/Hello-World\$ **git add README.md**

*In questo caso avendo un solo file sulla head potremmo utilizzare il comando “ **git add .** ” , ma in presenza di più file modificati andrebbe a salvarli tutti, essendo questa la sua funzionalità.

Le nostre modifiche sono ora salvate ma per renderle effettive bisogna dichiarare una “commit”** con il comando:

- Notebook-PC:~/Hello-World\$ **git commit -m “**spiegazione della commessa o commit, in cui spieghiamo le modifiche effettuate o descriviamo il file”**

Dopodiché possiamo sincronizzare su github con il comando “**git push**”

Essendo però la nostra prima commit su questo branch git ci chiederà di impostare il branch corrente da locale (sorgente) a remoto (di destinazione o a monte).

Utilizziamo il comando che ci suggerisce git:

Notebook-PC:~/Hello-World\$ **git push --set-upstream origin nome_branch**

*una volta eseguito il comando per sincronizzare le commit su questo branch basterà un “git push”

Git Tutorial da Terminale

```
marco@marco-Compaq-15-Notebook-PC: ~/Hello-World
File Modifica Visualizza Cerca Terminale Aiuto
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git add README.md
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git commit -m "I've made some example changes"
[new-edits 2a5abfa] I've made some example changes
1 file changed, 1 insertion(+), 1 deletion(-)
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git push
fatal: Il branch corrente new-edits non ha alcun branch upstream.
Per eseguire il push del branch corrente ed impostare remote come upstream, usa

git push --set-upstream origin new-edits

marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git push --set-upstream origin new-edits
Username for 'https://github.com': MarcoEna
Password for 'https://MarcoEna@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/MarcoEna/Hello-World.git
* [new branch]      new-edits -> new-edits
Branch new-edits set up to track remote branch new-edits from origin.
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$
```

Git Tutorial da Terminale

Andiamo a ora a congiungere le commit della nostra branch con il master:

- Ritorniamo sul master con “git checkout master”
- Utilizziamo la funzione di “**merge**” (fusione) con il comando:
Notebook-PC:~/Hello-World\$ **git merge nome_branch**
- Il seguente comando sarà un “git add .”
- Sincronizziamo la merge con un “git push”
- Dopo ogni “git push” ci vengono chieste le nostre credenziali di github, inseriamole e proseguiamo
- A questo punto abbiamo sincronizzato e fuso i due branch se volessimo poi riportare dei cambiamenti effettuati da remoto ci basterebbe usare il comando “**git pull**”

Git Tutorial da Terminale

```
marco@marco-Compaq-15-Notebook-PC: ~/Hello-World
File Modifica Visualizza Cerca Terminale Aiuto
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git checkout master
Si è passati al branch 'master'
Your branch is up-to-date with 'origin/master'.
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git merge new-edits
Aggiornamento di 1238450..2a5abfa
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git add .
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git push
Username for 'https://github.com': MarcoEna
Password for 'https://MarcoEna@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/MarcoEna/Hello-World.git
 1238450..2a5abfa master -> master
marco@marco-Compaq-15-Notebook-PC:~/Hello-World$ git pull
Already up-to-date.
```

Il nostro tutorial sui primi passi di git e github é concluso, questa presentazione vuole essere solo un introduzione, per approfondimenti consultare la Documentazione on-line