



GUIDA

A

GIT

ANTONIO MERCURIO

Guida a GIT

I comandi base svelati

antonio mercurio

Questo libro è in vendita presso <http://leanpub.com/guidaagit>

Questa versione è stata pubblicata il 2016-07-24



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2016 antonio mercurio

Also By antonio mercurio

Php le basi: per principianti

Laravel 5.2 : Per principianti

Indice

Dedica	1
Supporto	2
Requisiti	3
Cosa è GIT	4
Installazione	5
Come creare un repository	6
Inizializzare un repository	7
Aggiungere files al repository in seguito	12

Dedica

Dedico questo libro a tutti gli sviluppatori opensource che ogni giorno ci forniscono software utile a farci risparmiare tempo e denaro. Ringrazio anche Linus Torvald per aver creato GIT.

Supporto

Per chiarimenti, suggerimenti o per comunicare qualsiasi errore potete scrivermi a tonymercurio75@gmail.com

Requisiti

Per capire questo libro non serve essere programmatori ma basta possedere un pc con Windows o Linux oppure un Mac. Basta sapersi spostare nelle cartelle da terminale, conoscere quindi i comandi base e sapere creare una cartella sulla propria macchina locale.

Cosa è GIT

Immaginiamo di dover lavorare ad un progetto con altri sviluppatori, ed ognuno lavora singolarmente sul proprio pc. Ogni sviluppatore avrà una copia di backup in locale del progetto. Ma che succede se ad un certo punto il responsabile del progetto che è distante dal team chiede di vedere l'avanzamento dello sviluppo?. Beh negli anni passati ci si spediva centinaia di email con le proprie modifiche. Tutto questo presenta l'ovvio svantaggio che è dovuto al fatto che le modifiche non sono sincronizzate e non risiedono in un luogo dove tutti le possono vedere contemporaneamente. Per questo si svilupparono negli anni dei sistemi centralizzati di Version Control. In pratica un team lavora ad un progetto e tutti i files risiedono su un server centrale, quindi tutti devono accedere a questo server per apportare modifiche o per vederle. Lo svantaggio è che se il server è down e non si ha una copia di backup si perde tutto in un attimo. Questo problema lo affrontò Linus Torvald inventore del kernel di Linux che utilizzava uno di questi server per le copie del kernel. Ad un certo punto inventò GIT. Esso si basa su un sistema decentralizzato di version control, in sostanza ogni sviluppatore ha una copia in locale detta repository e le modifiche che fa le committa e le invia ad un server remoto. Se il server remoto ad un certo punto è down si può facilmente recuperare l'ultima versione. Quando si vuole annullare una modifica locale si può facilmente ritornare sui propri passi e su una versione precedente.

Installazione

Git bisogna scaricarlo dal sito ufficiale [sito](https://git-scm.com/downloads)¹. Da questa pagina scarichiamo la versione per il nostro sistema operativo. Per Windows c'è il file .exe che basta lanciare per l'installazione. Durante il processo di installazione scegliete sempre le opzioni già selezionate che vanno bene. Per linux per esempio Ubuntu basta digitare nel terminale questo comando `sudo apt-get install git` mentre per Mac c'è l'installer che potete lanciare tranquillamente.

¹<https://git-scm.com/downloads>

Come creare un repository

Dobbiamo creare la configurazione di Git inserendo il nostro nome ed email che verranno inseriti nei repository e saranno utili agli altri per capire chi ha apportato delle modifiche. Digitate questi comandi in sequenza

```
git config --global user.name="tony mercurio".
```

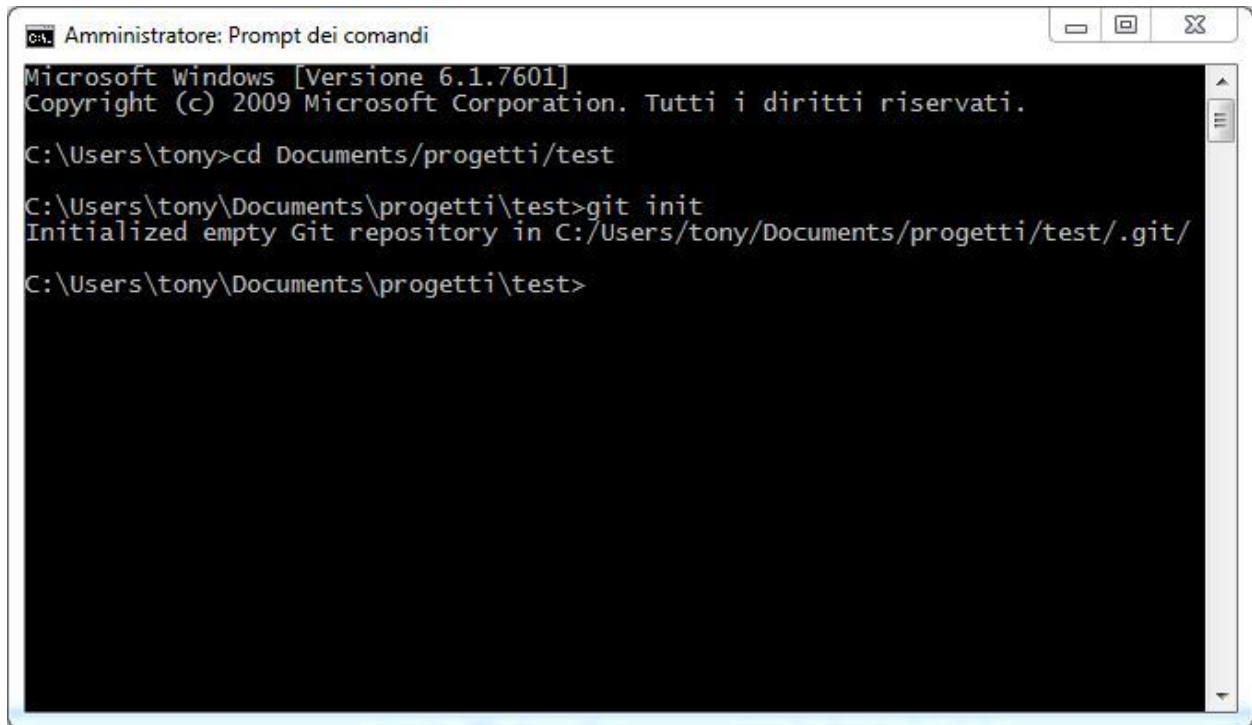
Questo comando serve ad inserire il vostro nome tra apici. Poi

```
git config --global user.email"tonymercurio75@gmail.com" .
```

Questo comando aggiunge la vostra email. Ora non dovrete più inserire questi comandi in seguito. La prima cosa da fare è creare sul proprio pc o mac una cartella con il nome del progetto in una posizione qualunque. Creiamo una cartella di nome progetti nella nostra cartella Documenti. Poi al suo interno creiamo una cartella per il nostro repository che chiamiamo test. Adesso da terminale spostiamoci in questa cartella `cd Documents/progetti/test`

Inizializzare un repository

Ora dobbiamo inizializzare il progetto con questo comando `git init` Ecco l'output nel terminale

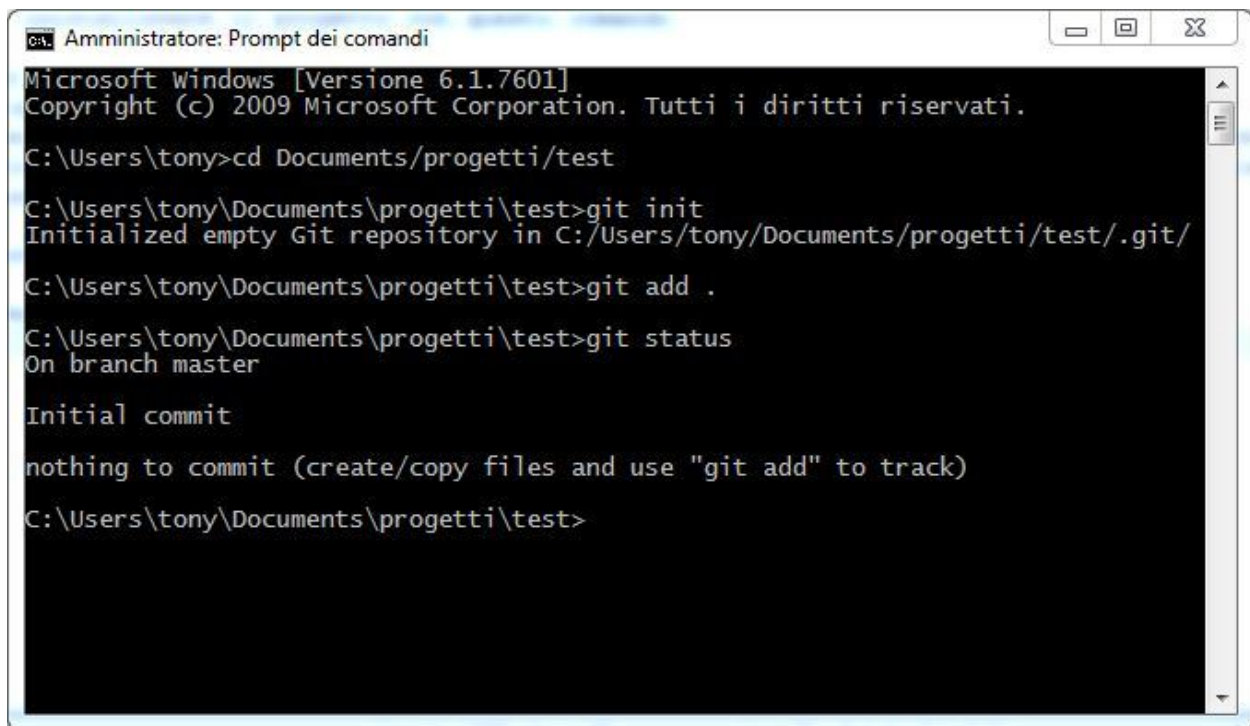
A screenshot of a Windows Command Prompt window titled "Amministratore: Prompt dei comandi". The window shows the following text: "Microsoft Windows [Versione 6.1.7601] Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati." followed by the command sequence: "C:\Users\tony>cd Documents/progetti/test", "C:\Users\tony\Documents\progetti\test>git init", and the output "Initialized empty Git repository in C:/Users/tony/Documents/progetti/test/.git/". The prompt then returns to "C:\Users\tony\Documents\progetti\test>".

```
Amministratore: Prompt dei comandi
Microsoft Windows [Versione 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\tony>cd Documents/progetti/test
C:\Users\tony\Documents\progetti\test>git init
Initialized empty Git repository in C:/Users/tony/Documents/progetti/test/.git/
C:\Users\tony\Documents\progetti\test>
```

`git init`

Ora create nella cartella del progetto i seguenti files in formato testo prova1 prova2 prova3 ed inserite qualche parola in essi e salvate. Dobbiamo aggiungere tutti i files del progetto e farli tracciare da Git con questo comando `git add .` Ora per vedere lo status del repository inseriamo questo comando `git status` e dovreste vedere questo risultato

A screenshot of a Windows Command Prompt window titled "Amministratore: Prompt dei comandi". The window shows the following commands and output:

```
Microsoft Windows [Versione 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\tony>cd Documents/progetti/test

C:\Users\tony\Documents\progetti\test>git init
Initialized empty Git repository in C:/Users/tony/Documents/progetti/test/.git/

C:\Users\tony\Documents\progetti\test>git add .

C:\Users\tony\Documents\progetti\test>git status
On branch master

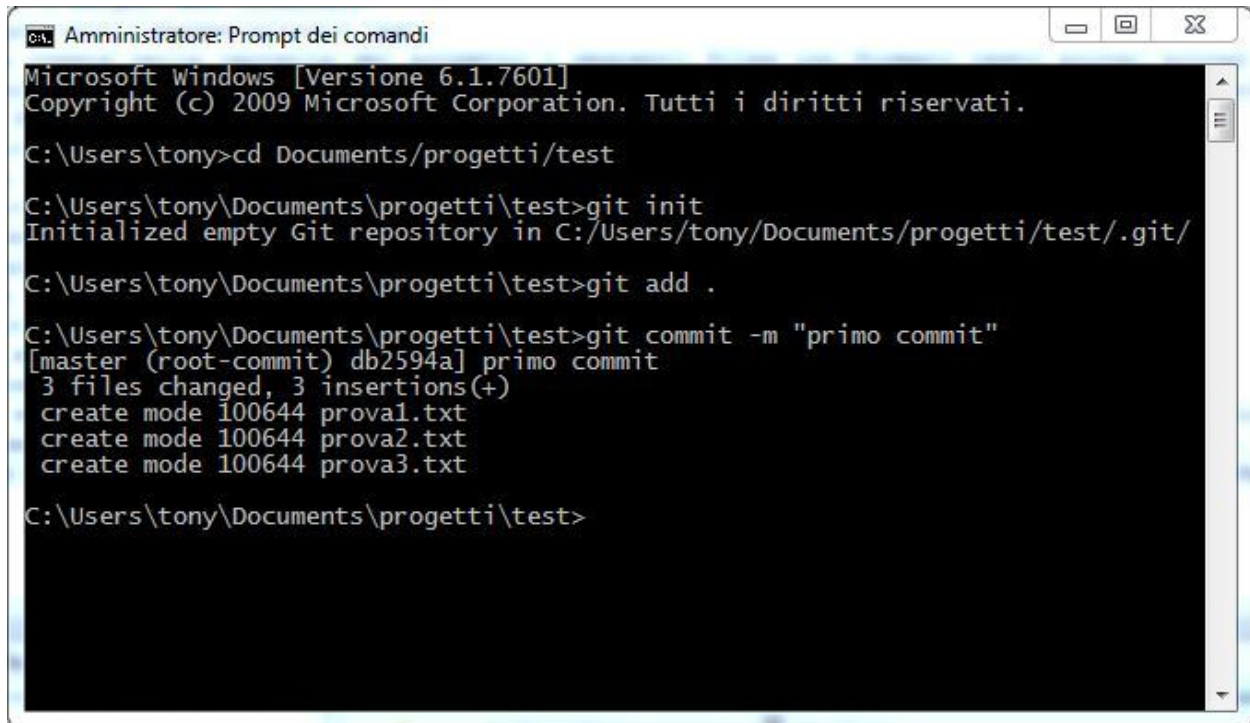
Initial commit

nothing to commit (create/copy files and use "git add" to track)

C:\Users\tony\Documents\progetti\test>
```

git status

Ora che i files sono tracciabili da Git dobbiamo eseguire un commit in questo modo `git commit -m "primo commit"` ed ecco il messaggio nel terminale



```
Amministratore: Prompt dei comandi
Microsoft Windows [Versione 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\tony>cd Documents/progetti/test

C:\Users\tony\Documents\progetti\test>git init
Initialized empty Git repository in C:/Users/tony/Documents/progetti/test/.git/

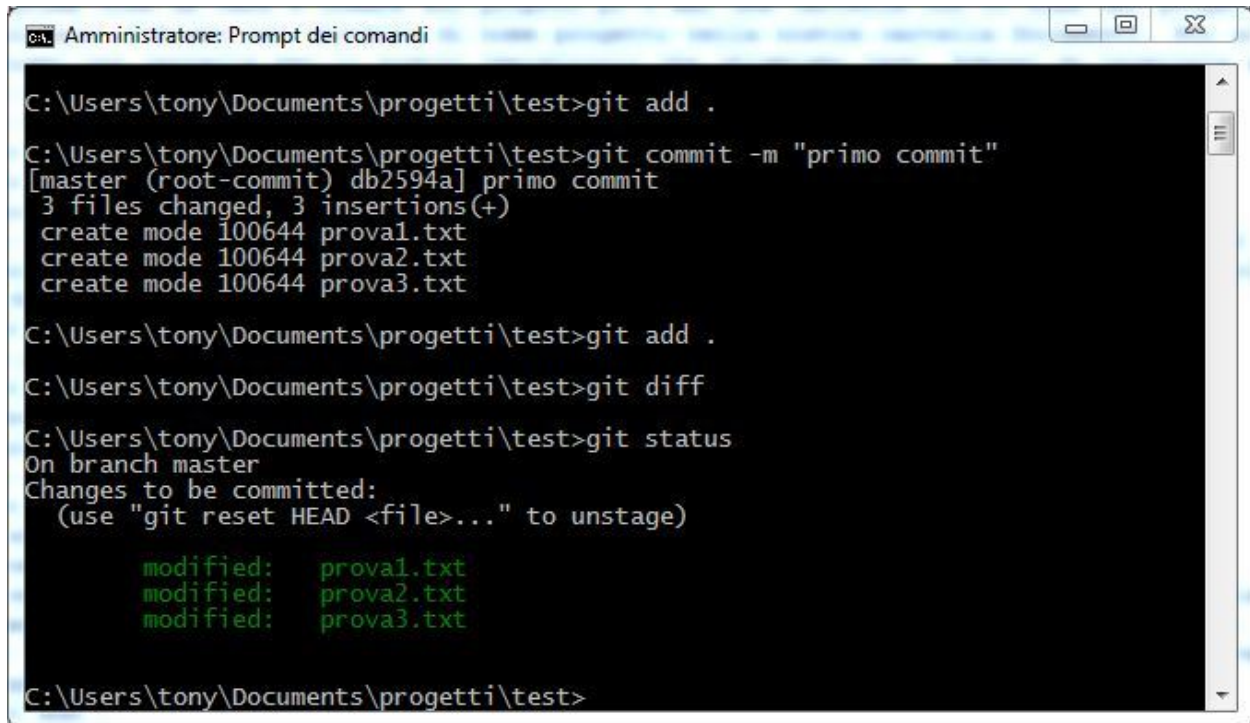
C:\Users\tony\Documents\progetti\test>git add .

C:\Users\tony\Documents\progetti\test>git commit -m "primo commit"
[master (root-commit) db2594a] primo commit
3 files changed, 3 insertions(+)
create mode 100644 prova1.txt
create mode 100644 prova2.txt
create mode 100644 prova3.txt

C:\Users\tony\Documents\progetti\test>
```

commit iniziale

In sostanza ci dice che ha aggiunto al commit i nostri tre files. Dovete usare il suffisso -m al comando per inserire un messaggio utile per far capire di che modifica stiamo parlando. Il messaggio è quello che inserite fra apici. Ora inserite una parola qualunque nei tre files di testo e salvate le modifiche. Eseguite adesso il seguente comando `git add .`. Dovete sempre digitarlo prima di ogni commit. Ora vediamo lo status del repository `git status`. Ed ecco il messaggio nel terminale che ci avvisa delle modifiche apportate ai files e che devono essere committate.



```
C:\Users\tony\Documents\progetti\test>git add .

C:\Users\tony\Documents\progetti\test>git commit -m "primo commit"
[master (root-commit) db2594a] primo commit
3 files changed, 3 insertions(+)
create mode 100644 prova1.txt
create mode 100644 prova2.txt
create mode 100644 prova3.txt

C:\Users\tony\Documents\progetti\test>git add .

C:\Users\tony\Documents\progetti\test>git diff

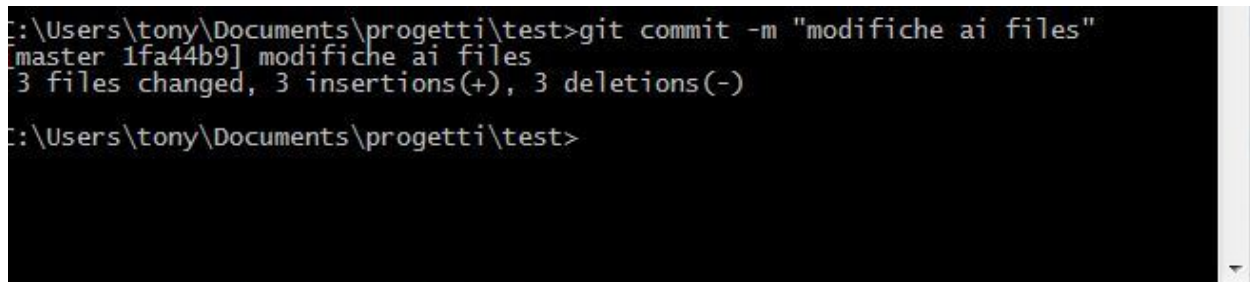
C:\Users\tony\Documents\progetti\test>git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

       modified:   prova1.txt
       modified:   prova2.txt
       modified:   prova3.txt

C:\Users\tony\Documents\progetti\test>
```

git status 2

Ora per salvare le modifiche eseguiamo il commit `git commit -m "modifica ai files"` . Il messaggio che inseriamo deve essere esplicito e chiarificante. Ora noterete questo messaggio

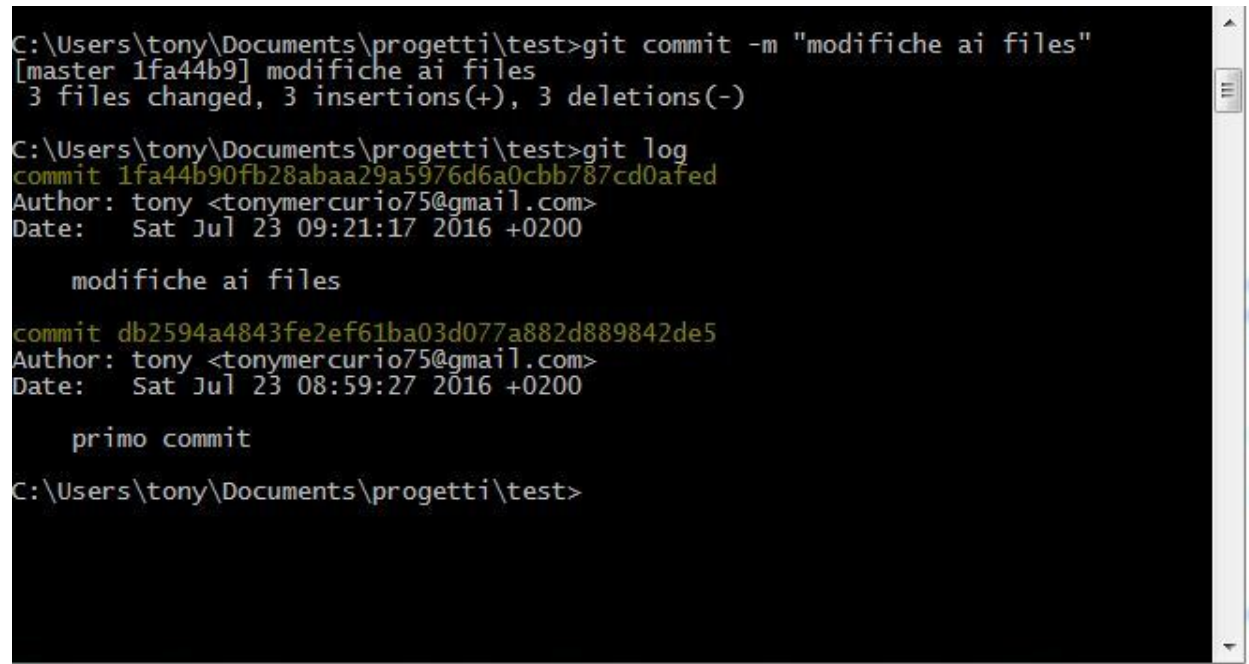


```
C:\Users\tony\Documents\progetti\test>git commit -m "modifiche ai files"
[master 1fa44b9] modifiche ai files
3 files changed, 3 insertions(+), 3 deletions(-)

C:\Users\tony\Documents\progetti\test>
```

commit 2

Per conoscere tutta la storia di un repository inserite questo comando `git log` . Ed ecco che ci fornisce informazioni sui commit che abbiamo eseguito fino ad ora e chi li ha eseguiti.

A screenshot of a Windows command prompt window. The background is black with white text. The prompt shows the user 'tony' in the directory 'C:\Users\tony\Documents\progetti\test'. The first command executed is 'git commit -m "modifiche ai files"', which results in a commit message '[master 1fa44b9] modifiche ai files' and a summary '3 files changed, 3 insertions(+), 3 deletions(-)'. The second command is 'git log', which displays two commit entries. The first entry is for commit '1fa44b90fb28abaa29a5976d6a0cbb787cd0afed' with the message 'modifiche ai files'. The second entry is for commit 'db2594a4843fe2ef61ba03d077a882d889842de5' with the message 'primo commit'. Both commits are attributed to 'tony <tonymercurio75@gmail.com>' and dated 'Sat Jul 23 09:21:17 2016 +0200' and 'Sat Jul 23 08:59:27 2016 +0200' respectively. The prompt ends with 'C:\Users\tony\Documents\progetti\test>'.

```
C:\Users\tony\Documents\progetti\test>git commit -m "modifiche ai files"
[master 1fa44b9] modifiche ai files
3 files changed, 3 insertions(+), 3 deletions(-)

C:\Users\tony\Documents\progetti\test>git log
commit 1fa44b90fb28abaa29a5976d6a0cbb787cd0afed
Author: tony <tonymercurio75@gmail.com>
Date: Sat Jul 23 09:21:17 2016 +0200

    modifiche ai files

commit db2594a4843fe2ef61ba03d077a882d889842de5
Author: tony <tonymercurio75@gmail.com>
Date: Sat Jul 23 08:59:27 2016 +0200

    primo commit

C:\Users\tony\Documents\progetti\test>
```

git log

Per allenarci un pò eseguiamo altre modifiche ai file e poi di nuovo git add .

e a seguire il commit che chiamiamo terzo commit git commit -m "terzo commit"

Aggiungere files al repository in seguito

Nella cartella del progetto aggiungete un nuovo file di testo prova4 con del testo dentro. Poi facciamolo tracciare da git `git add prova4.txt` ed eseguiamo il commit `git commit -m "aggiunto file prova4"`