

IT: Manuale di utilizzo rapido di GIT

GIT è un sistema di controllo delle versioni distribuito (Distributed Version Control System).

E' stato sviluppato nel 2005 da Linus Torvalds, creatore di Linux.

Esistono diverse modalità di *Versioning*, cioè di gestione delle versioni di un software che consistono nella creazione di copie della cartella di lavoro ogni qualvolta si verifichi una modifica sostanziale del progetto; differiscono tra loro nel tipo di salvataggio che può venire in locale, semplicemente per tenere conto delle modifiche del progetto, o attraverso un sistema centralizzato su un server, per permettere a diversi collaboratori di lavorare al progetto essendo sempre aggiornati sulle modifiche effettuate dagli altri componenti.

Un'ultima soluzione è quella del sistema distribuito, in cui ogni componente del team è in grado "separarsi" dalla cartella di lavoro principale (*master*) con la creazione di un branch. In ogni caso, il contenuto presente sul server verrà scaricato creando un backup locale sul proprio terminale. Per lavorare sulla base delle ultime modifiche salvate sul server, bisogna scaricare l'ultima versione disponibile (*push*) e, al termine dei lavori, effettuare il processo inverso caricando sul server (*pull*).

L'insieme dei file e delle cartelle di lavoro è chiamato *repository*.

Nonostante negli ultimi anni siano stati sviluppati diversi software che permettono l'utilizzo di GIT tramite interfaccia grafica, è importante sottolineare che ogni operazione del sistema è eseguibile su linea di comando tramite l'uso di specifici comandi; di seguito vedremo i principali.

Per creare un nuovo repository locale dobbiamo portarci nella cartella in cui vogliamo salvare il contenuto e utilizzare il comando

```
git init
```

In alternativa è possibile clonare una repository già esistente da un percorso relativo al proprio terminale o da un collegamento internet:

```
git clone /path/locale
```

```
git clone nomeutente@host:/u/r/l
```

Per apportare delle modifiche alla nostra repository abbiamo bisogno di eseguire diversi passaggi, necessari per non rischiare la perdita di dati. Inizialmente aggiungiamo i file al nostro indice con il comando

```
git add <nomedelfile>
```

dopodichè, dobbiamo eseguire il commit, ovvero validare la modifica effettuata aggiungendo i file alla repository locale:

```
git commit -m "Descrizione Commit"
```

Non ci resta che caricare sul repository remoto utilizzando

```
git push origin master
```

Per aggiornare il nostro repository locale con ciò che si trova in remoto utilizziamo

```
git pull
```

Possiamo creare un branch utilizzando il comando

```
git branch <nome Branch>
```

Se vogliamo caricare ciò che è stato aggiornato nel branch dobbiamo ricordarci di esplicitare il nome durante il push:

```
git push origin <nome Branch>
```

Per tornare a lavorare sul master utilizziamo

```
git checkout master
```

Possiamo decidere di incorporare un branch con un altro branch o con il master:

```
git merge <nome branch>
```

Prima di effettuare questa operazione può essere utile visualizzare analogie e differenze dei due branch per quanto riguarda la composizione dei file:

```
git diff <nome branch1> <nome branch2>
```

EN: GIT quick use manual

GIT is a Distributed Version Control System.

It was developed in 2005 by Linus Torvalds, creator of Linux.

There are different versioning modes, that is, the management of the versions of the software which consist in the creation of copies of the working folder every time a substantial modification of the project occurs; they differ from each other in the type of saving that can come, for example locally, simply taking account of the changes in the project, or through a centralized system on a server, to allow different teammates that work on the project to being always updated on the changes made by the other components.

A final solution is the distributed system, in which each team member is able to "split" from the main (master) working folder by creating a branch. In any case, the content present on the server will be downloaded making a local backup on your terminal. To work on the latest changes saved on the server, you must download them (push) and, at the end of the work, carry out the reverse process by uploading to the server (pull).

The set of files and folders is called a repository.

In recent years several software has been developed to allow the use of GIT through a graphical interface. It is important to underline that every GIT operation can be performed on the command line through the use of specific commands; below we will see the most used commands.

To create a new local repository we have to go to the folder where we want to save the content and use the command

```
git init
```

Alternatively, you can clone an existing repository from a local path or from the web:

```
git clone /path/locale
```

```
git clone nomeutente@host:/u/r/l
```

To make changes to our repository we need to perform several steps, which are necessary in order not to risk data loss. We initially add the files to our index with the command

```
git add <file name>
```

after which, we must commit, to validate the changes made by adding the files to the local repository:

```
git commit -m "commit description"
```

Now, we just have to upload to the remote repository using

```
git push origin master
```

To update our local repository with what is located remotely we use

```
git pull
```

We can create a branch using the command

```
git branch <branch name>
```

If we want to load what has been modified in the branch we must remember to make explicit the name during the push:

```
git push origin <branch name>
```

To return to work on the master we use

```
git checkout master
```

We can merge a branch with another branch or with the master:

```
git merge <branch_name>
```

Before doing this it can be useful have a look at similarities and differences between the two branches as regards the composition of the files:

```
git diff <branch1> <branch2>
```