

# Prontuario Comandi GIT

## SU PROGETTO NUOVO IN LOCALE:

```
git init
```

CREARE I FILE .gitignore e README.md *(importante farlo subito!)*

```
git add *
```

```
git commit -a -m "First commit"
```

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

```
git push -u origin master (dalla seconda volta: git push)
```

## SE VOLETE CONOSCE L'URL DEL REPOSITORY REMOTO

```
git remote show origin
```

## SE AVETE COLLEGATO UN REPOSITORY SBAGLIATO

```
git remote remove origin (rimuove un collegamento errato)
```

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

## PER SCARICARE IN LOCALE DA REMOTO

```
git clone https://github.com/.../....git (una volta sola!)
```

## PER SINCRONIZZARE REPOSITORY LOCALE E REMOTO

git add *	<i>(solo se ci sono nuovi file)</i>
git commit -a -m "commit message"	
git push	<i>(da locale verso remoto)</i>
git pull	<i>(da remoto verso locale)</i>

## **QUANDO MODIFICATE IL .gitignore**

git rm -r --cached .	<i>(attenzione al puntino finale!)</i>
git add .	<i>(attenzione al puntino finale!)</i>
git commit -m "The new .gitignore is now working"	
git push	

## **CREAZIONE E UTILIZZO DEI BRANCH**

git checkout -b develop	<i>(crea e attiva il branch develop)</i>
-------------------------	--

### **IN ALTERNATIVA:**

git branch develop	<i>(crea il branch develop)</i>
git checkout develop	<i>(attiva il branch develop)</i>

(... si sviluppa una nuova funzionalità lavorando su develop ...)

git add *	
git commit -a -m "commit message new feature"	
git push --set-upstream origin develop	<b>(solo la prima volta)</b>

(... si sviluppano ulteriori funzionalità lavorando su develop ...)

git add *	
git commit -a -m "commit message other new features"	
git push	<b>(dalla seconda volta in poi)</b>

(... si decide di allineare master a develop - **merge** ...)

git checkout master	<b>(torna su branch master)</b>
git merge develop	<b>(porta su master gli sviluppi di develop)</b>