Homework Day 1

1. Origin è un riferimento al nome del repository remoto in cui “pushare” le commit, per convenzione il repository remoto è chiamato origin, questa parola è un alias in locale per un particolare repository remoto.
2. Esistono due tipologie di creazione di un repository:

* Inizializzare il repository, cioè creare il progetto e tracciare il repository da zero.

I comandi sono:

* *$ git init [project-name]*
* *$ git remote add origin <remote\_repo\_url>*
* *$ git push -u origin master*
* Clonare un repository:
  + *$ git clone [url]*che scarica dalla url tutto il repository copiandolo in locale

1. *$ git push origin <branch\_name> - -force* oppure *$ git push origin <branch\_name> -f*.

Eseguire push force “aggira” la restrizione che non permette alla push di aggiornare un riferimento al repository remoto che non è un antenato del riferimento locale utilizzato per sovrascriverlo.

Bisogna essere sicuri che nessun altro stia effettuando delle commit prima del push con il flag force, altrimenti questa persona perderà il proprio lavoro.

1. L’Id univoco di 40 cifre esadecimali per le commit e viene generato applicando un algoritmo hash chiamato SHA-1. Git usa le seguenti informazioni per generare l’ID con lo SHA-1:

* L’albero sorgente delle commit
* Lo SHA-1 della commit parent
* Informazioni sull’autore
* Informazioni sul committer
* Il messaggio del commit

1. *$git reset <commit>* elimina l’ultima commit dall’elenco e pulisce la stage area lasciando intatta la working directory.

*$ git revert <commit>* crea una nuova commit annullando i cambiamenti effettuati dalla <commit> e la aggiunge al branch corrente.

*$ git reset --hard* elimina tutte le commit e pulisce la stage area lasciando inalterata la working directory.

1. *Merge* effettua l’integrazione di due branch, ad esempio il branch di feature con il develop. Non tiene traccia della history delle commit.

*Rebase* effettua l’integrazione del branch corrente su una “base”, che può essere l’Id della commit, il nome di un branch, un tag o un riferimento ad HEAD. Essa tiene traccia della history relativa a tutte le commit effettuate.

*$ git merge feature develop*

*$ git rebase develop*