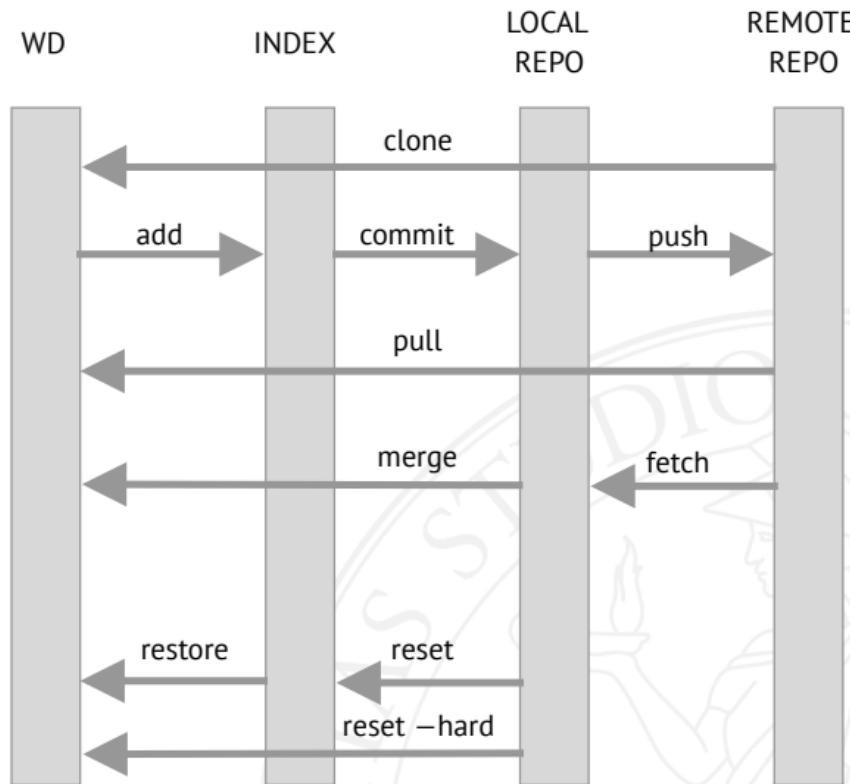
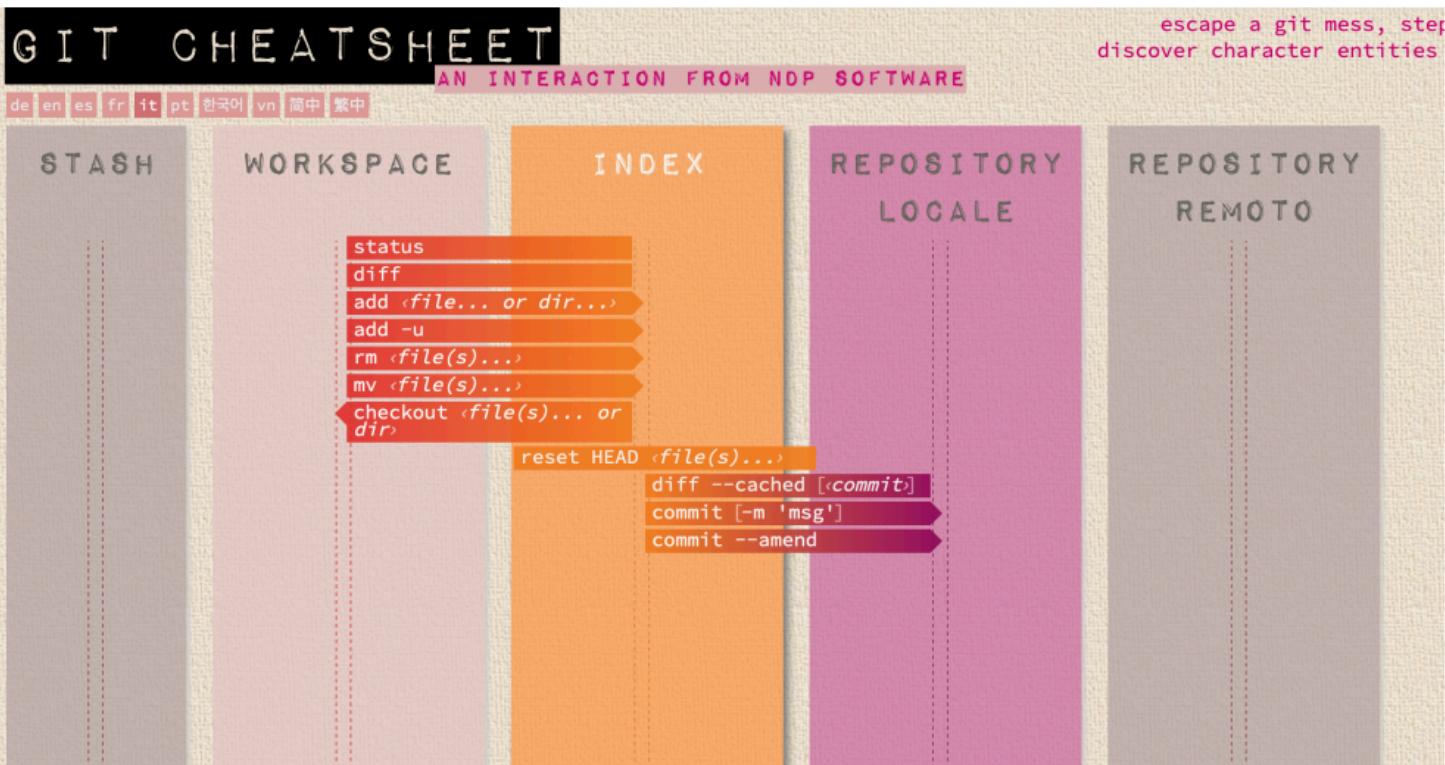


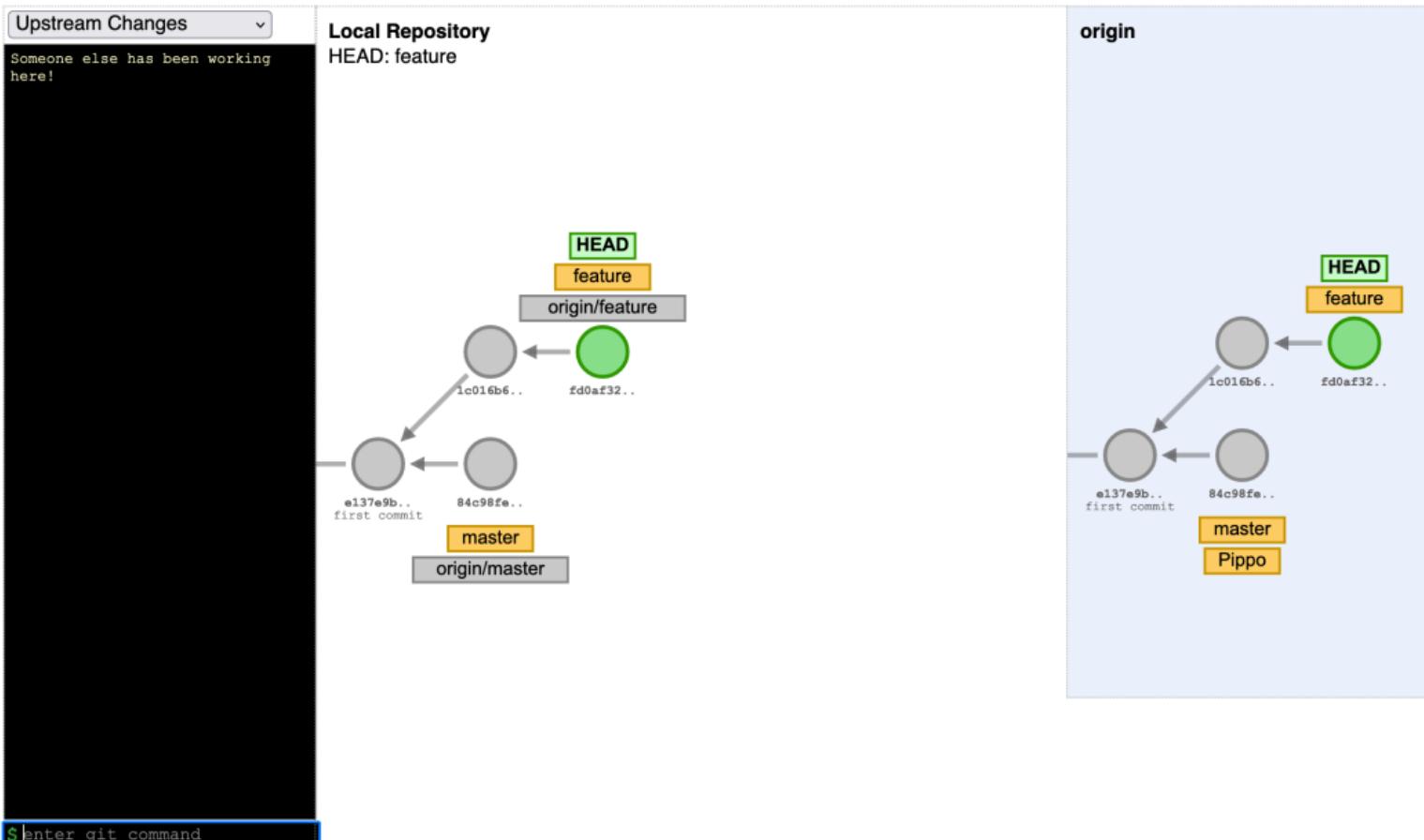
# Git common command/operation



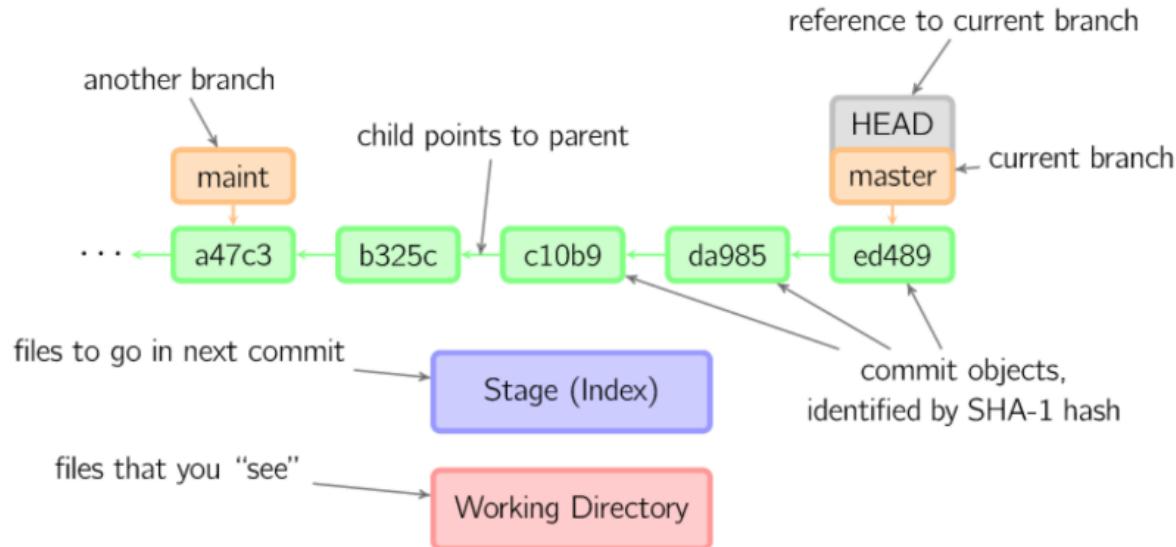


### repository locale

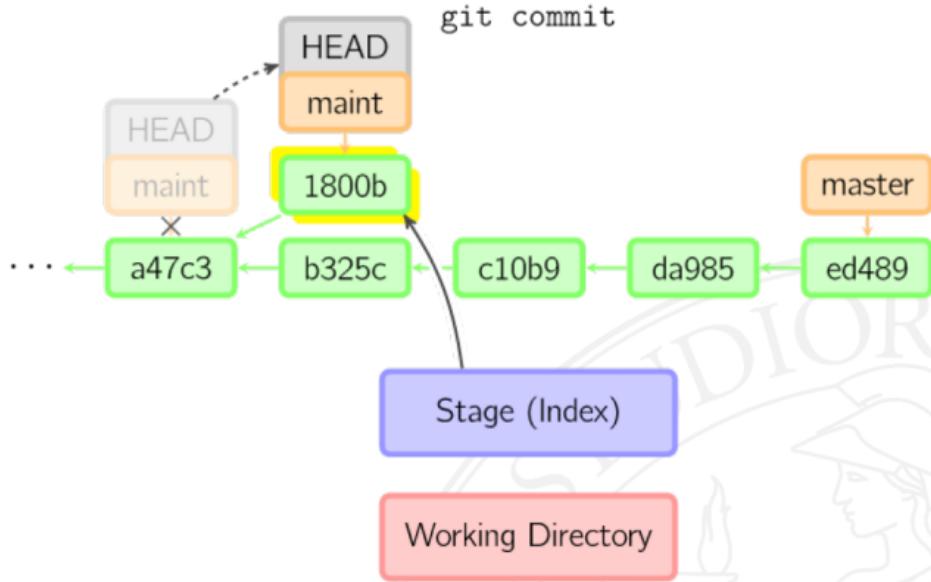
Uno spazio che contiene tutti i file committati e le informazioni sui branch. Si presenta come una sotto-cartella denominata .git. I nomi tipici per i branch usati qui sono: **main**, **feature-x**, **bugfix-y**.



# git

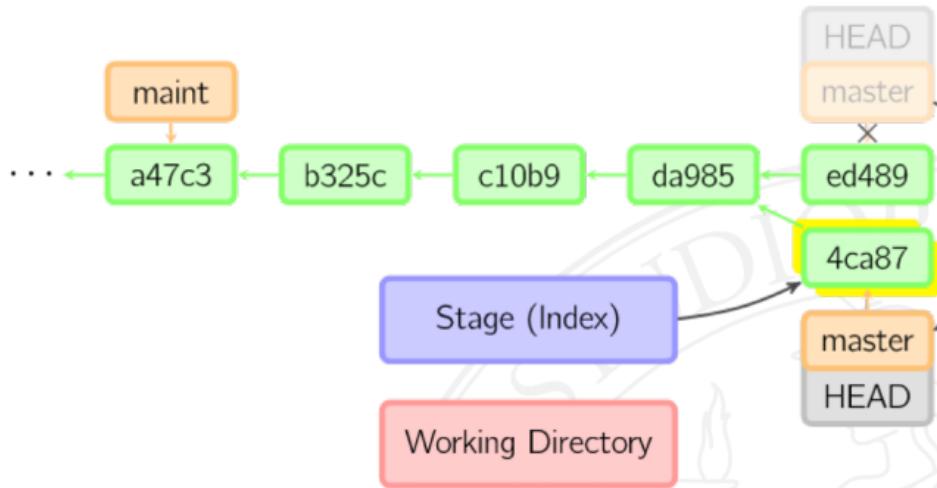


# git commit



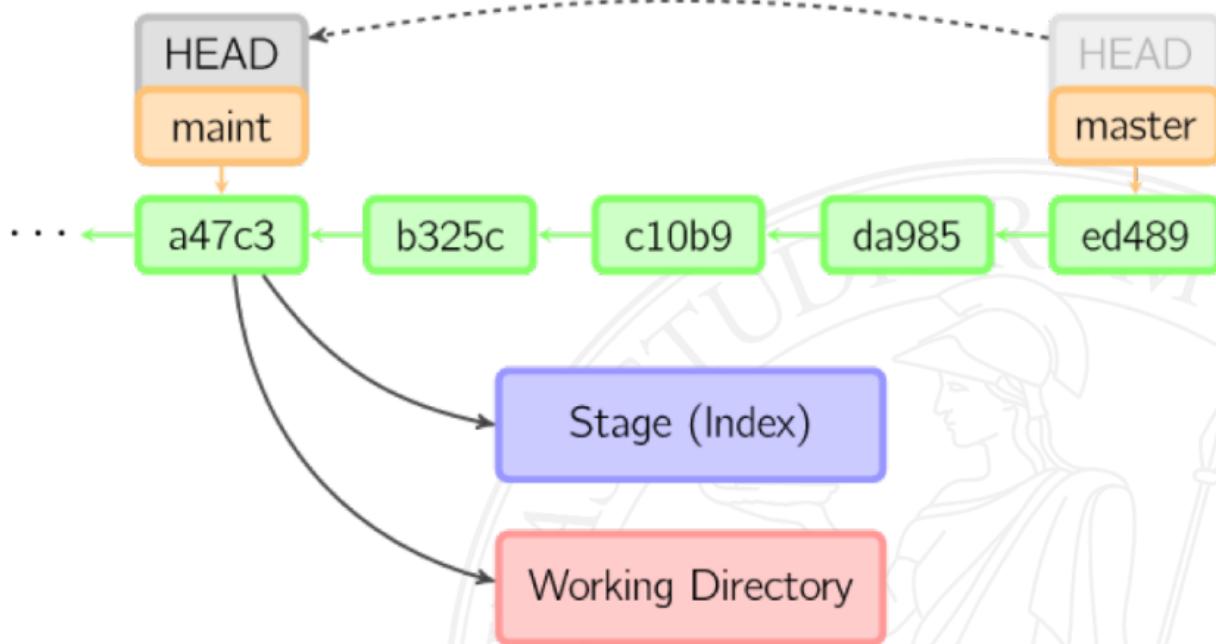
# git commit --amend

git commit --amend

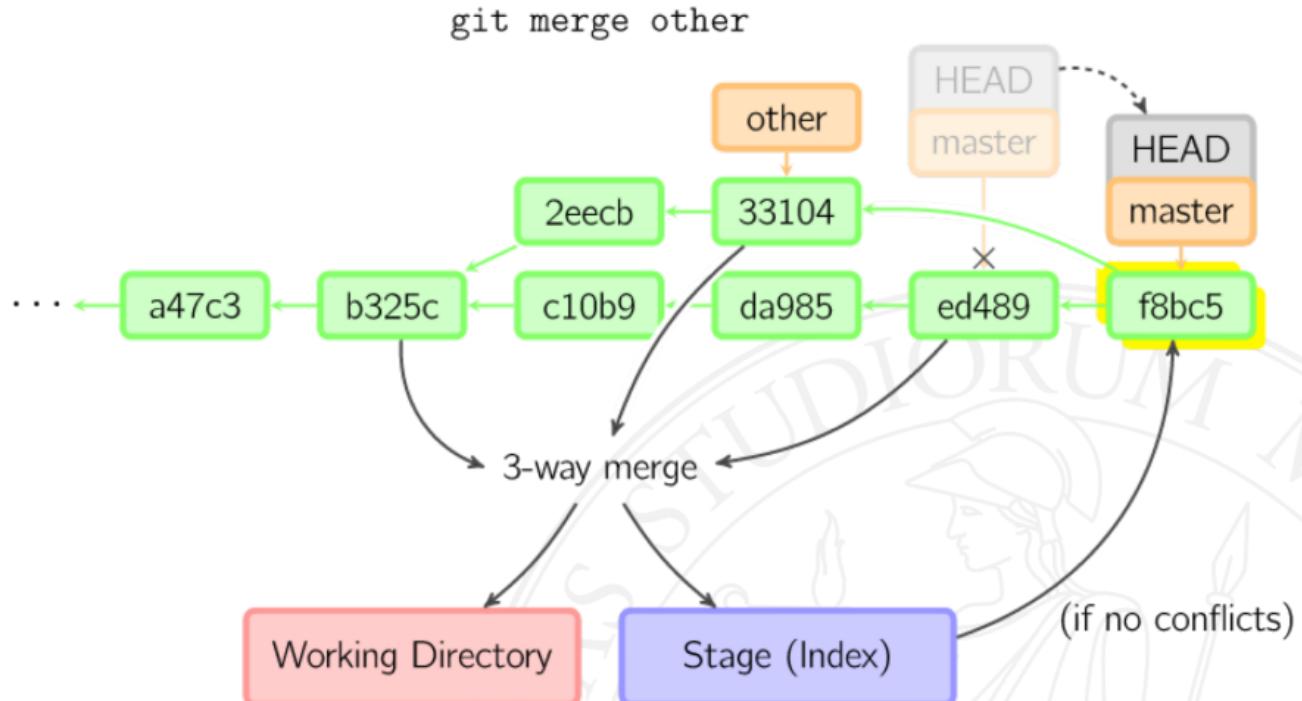


# git checkout

git checkout maint

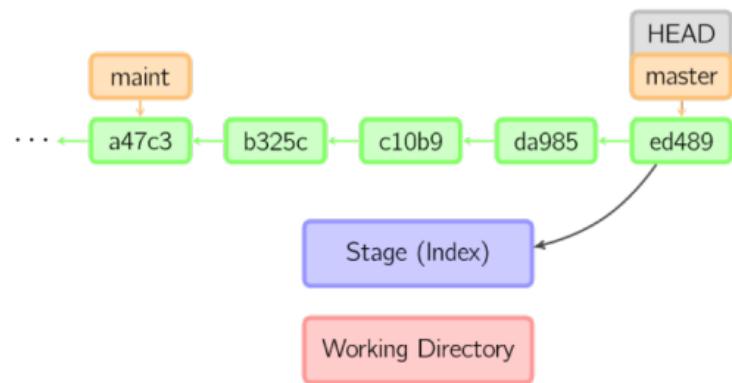


# git merge

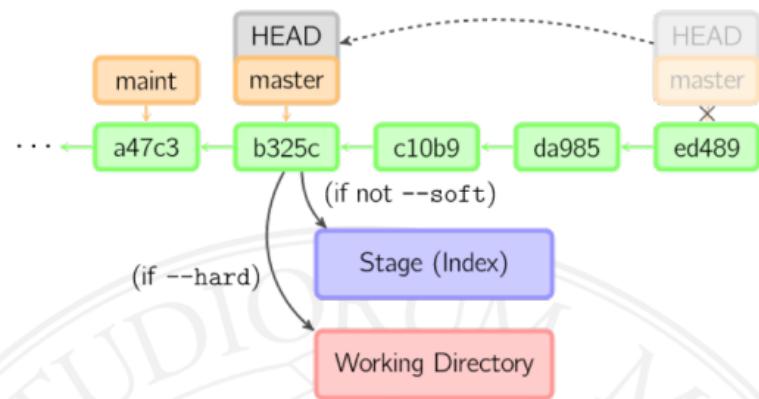


# git reset

`git reset -- files`

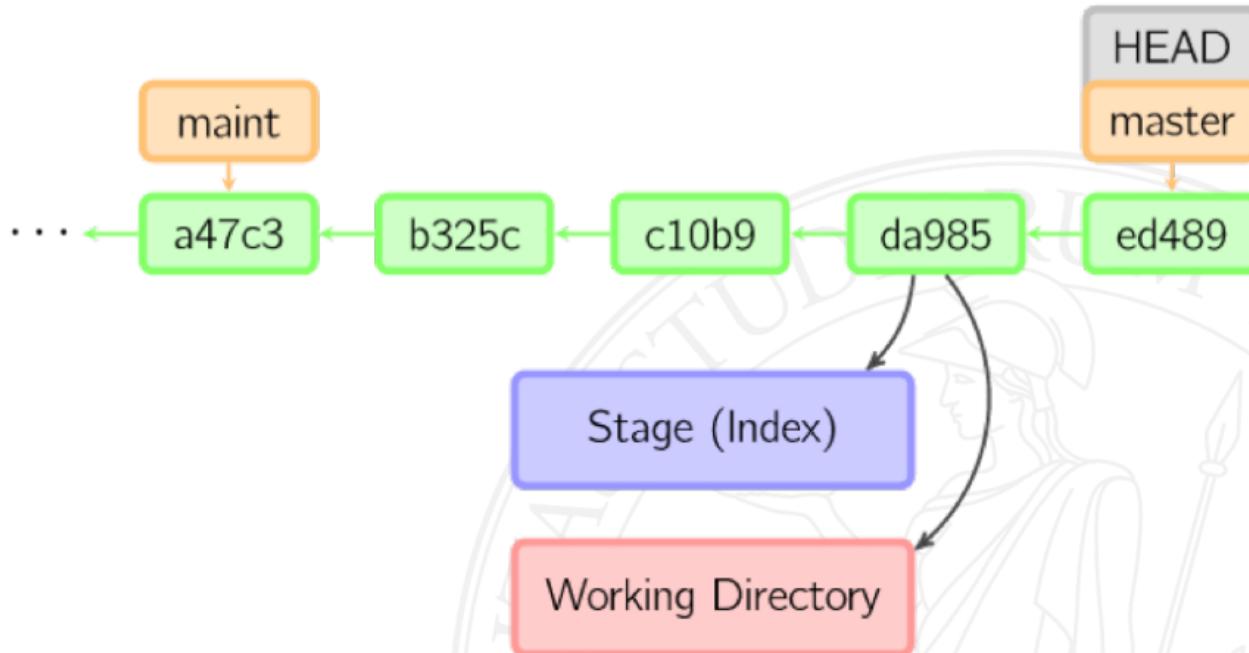


`git reset HEAD~3`



# git checkout

git checkout HEAD~ files



# Git Workflow

## Come si usano i branch?

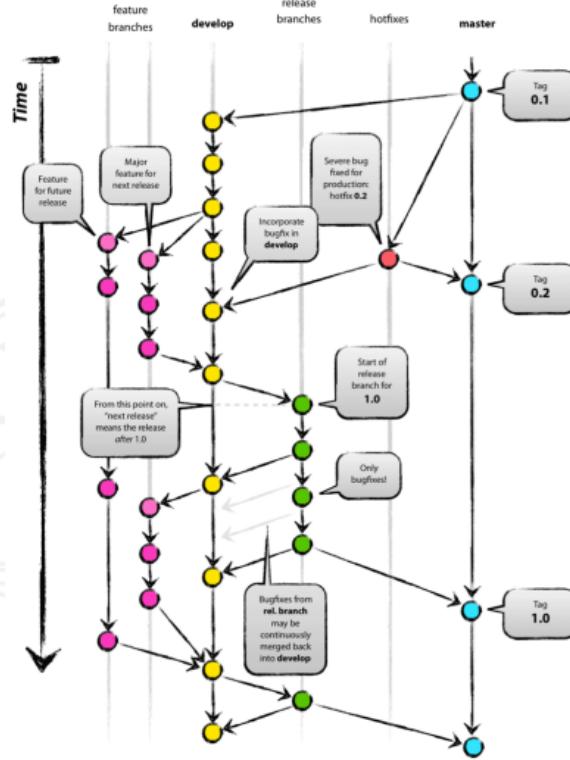
Praticamente impossibile non usarli!

- libertà completa
- seguendo delle linee guida
- all'interno di un workflow più grande



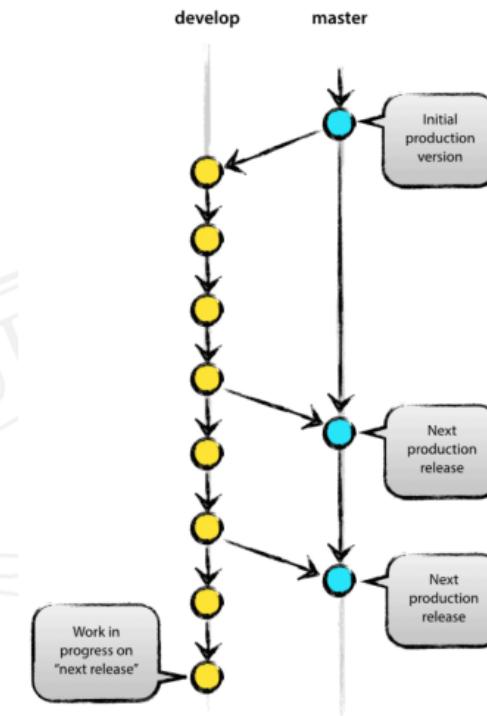
# GitFlow [AVH]

- Differenziazione tra tipi di branch
- nuove operazioni “guidate”
- non visibile nella figura, ma tratta anche i remote



# master e develop

- Sono due rami con “vita infinita”
- **master** (e in particolare origin/master) contiene le versioni “stabili e pronte alla consegna”
- **develop** è il ramo di integrazione



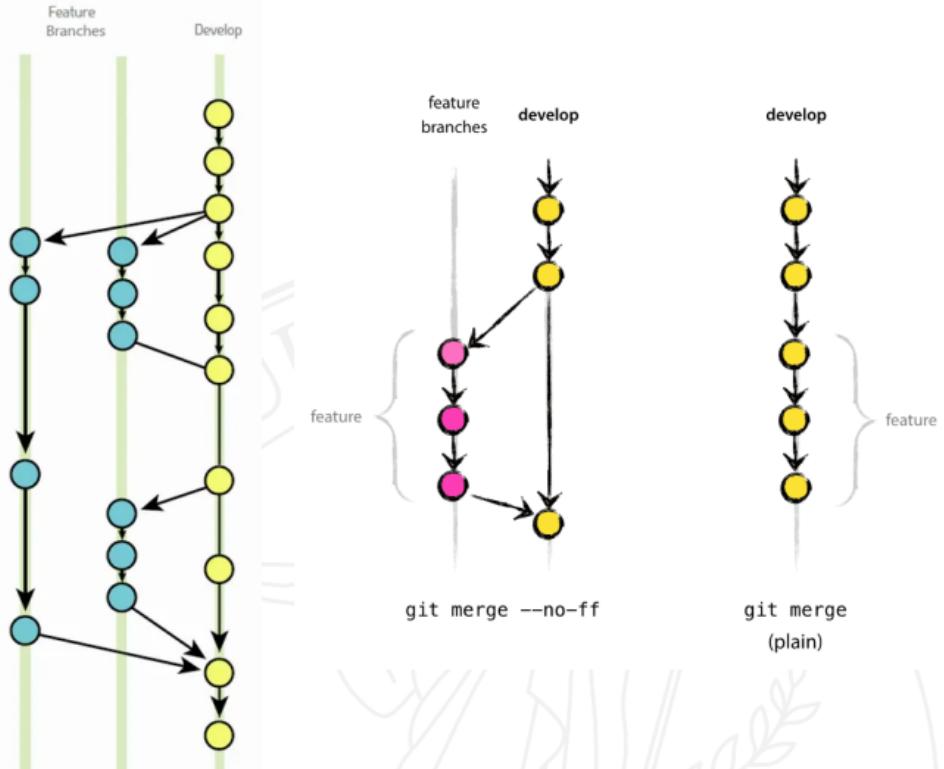
# feature

- git flow feature start *feat1*

```
git checkout develop  
git branch feat1  
git checkout feat1
```

- git flow feature finish *feat1*

```
git checkout develop  
git merge --no-ff feat1  
git branch -d feat1
```



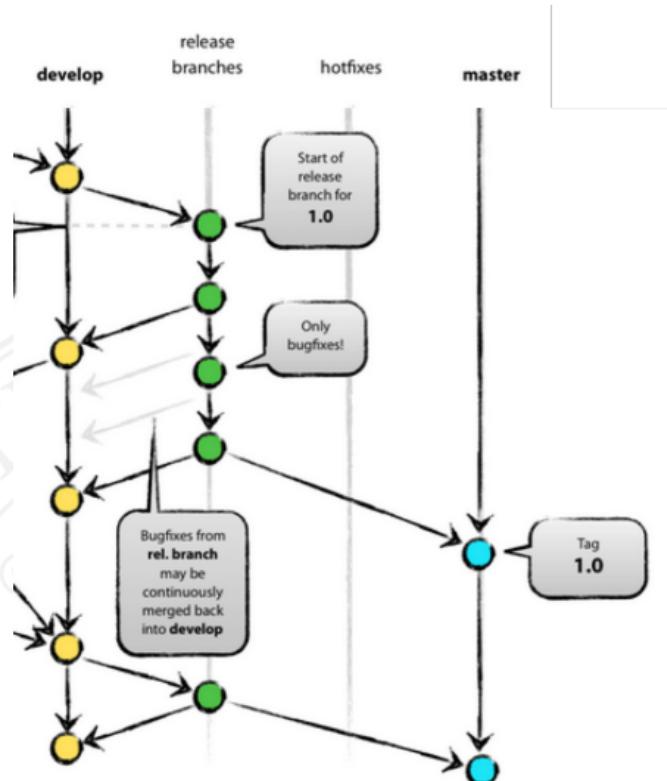
# release

- git flow release start ver

```
git checkout -b ver develop
```

- git flow release finish ver

```
git checkout master  
git merge --no-ff release-ver  
git tag -a ver  
git checkout develop  
git merge --no-ff release-ver  
git branch -d release-ver
```



# hotfix

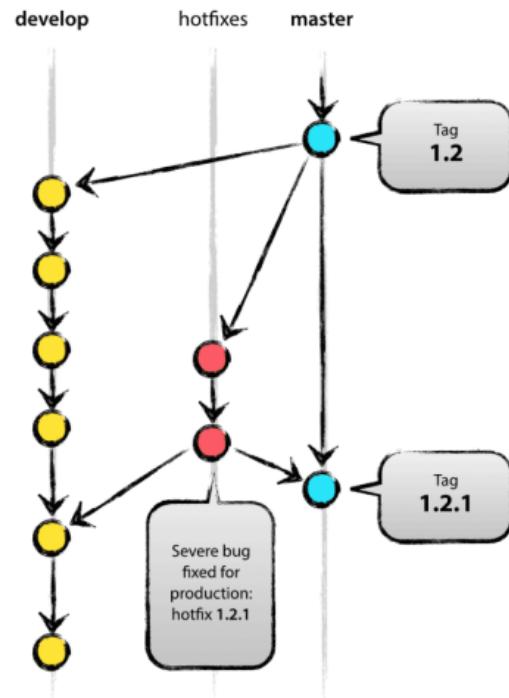
Riparazione veloce di difetti urgenti senza aspettare prossima release

- git flow hotfix start ver

```
git checkout -b ver master
```

- git flow hotfix finish ver

```
git checkout master  
git merge --no-ff ver  
git tag -a ver  
git checkout develop # uhm ???  
git merge --no-ff ver  
git branch -d ver
```



# Limiti di git

- singolo livello di autorizzazione
- nessun livello di review

