**git reset**

* git reset este similar cu git checkout
* git checkout pur si simplu muta ref la Head de la un commit la altul din branch, si chiar de mutam ref la Head cu 5 commituri in urma, acete commituri oricum vor exista
* git reset in schimb, modifica nu doar referinta la Head, ci si la branch
* git checkout b





Acum repostioriul este in detached Head state

* Dar iata ce va face git reset b
* 

Vedem ca commits de pana la b au fost separate de branch

**Arguments**

* git reset are 3 argumente:

- --mixed este default. git reset = git reset --mixed HashCommit

- --hard

- --soft

* --mixed

Acesta este default.

Daca avem de ex urmatoarele commituri:

Commit3 <- Head

Commit2

Commit1

StartCommit

* Acum, daca vom folosi git reset Head StartCommit(Hash la el), branch in care suntem va muta Head pe StartCommit si commiturile de dupa el, adica Commit1,2 si 3 vor fi desprinse de brach**. Totusi, toate modificarile ce le salva Head vor ramane in working tree, adica pe masa de lucru, doar ca totul vor fi unstagd. Deci, nu le pierdem, putem sa dam git add si crea un commit iar cu ele, si va fi unul doar in loc de atat de multe. E bundaca am facut niste commituri cam rele si vrem sa refacem totul in unul doar.**

**Deci, dupa ce Head a trecut la StartCommit**

* --soft este exact ca git reset --mixed, doar ca ceea ce pastra Head va fi pus in staging area, nu pur si simplu in working tree.
* --hard

Acesta este cel mai periculos, dar si folosit. Daca vom folosi --hard, se vor pierde commiturile si ceea ce a salvat commitul Head va fi pierdut!Nu va fi pus in working tree nimic! Nu mai putem nicidecum restabili ce am pierdut.