

# Version (revision) Control

Verzioniranje Dokumentata

POWERED BY:



# GIT - setup

→ GitHub - kreiranje accounta

→ Osnovne postavke:

```
git config --global user.name "Ime Prezime"
```

```
git config --global user.email "user@example.com"
```

```
git config --list
```

# GIT - kreiranje repozitorija

## → Inicijalizacija repozitorija

```
$ mkdir dir_name; cd dir_name  
$ git init
```

## → Try it out:

Kreirajte direktorij naziva **sparkschool** i cd u njega  
Provjerite nalazite li se u novokreiranom direktoriju  
Inicijalizirajte repozitorij (git init)

## GIT - status

→ Koristi se za provjeru stanja lokalnog repozitorija

→ Try it out:

Provjerite jeste li u repozitoriju

```
$ git status
```

## GIT - add

- Koristi se za dodati file u git
- Try it out :

Provjerite jeste li u repozitoriju sparkschool

Kreirajte file naziva README.md

```
$ git add README.md
```

## GIT - commit

- Koristi se za commitati file iz radne kopije u lokalni repozitorij
- Try it out:

Provjerite jeste li u repozitoriju sparkschool

Odradite git status za provjeriti trenutno stanje

```
$ git commit -m "commit poruka"
```

Dodajte i commitajte još jedan file

# GIT - povezivanje s remote

## → Povezivanje s GitHub-om

Kliknite na New u izborniku s lijeve strane

Dajte naziv repozitoriju - najbolje isti kao i lokalno

Pratite uputstva u *'push an existing repository from the command line'* dijelu

## → Try it out:

Dodajte repo **sparkschool** na GitHub

## GIT - pull

- Koristi se za povlačenje promjena sa remote repozitorija
- Try it out:

```
Provjerite jeste li u repozitoriju  
$ git pull
```



## GIT - push

- Koristi se za pushanje lokalnih commitova na remote repozitorij
- Try it out:

Provjerite jeste li u repozitoriju **sparkschool**

Odradite git status za provjeriti trenutno stanje

\$ git push

Provjerite svoj GitHub account da vidite promjene

## GIT - get help

→ **Get help:**

```
$ git help
```

```
$ man git
```

→ **Primjer:**

```
$ git help config
```

# Playground

- U repozitoriju sparkschool kreirajte par fileova, dodajte, commitajte, pushajte
- Editirajte neke fileove, dodajte, commitajte, pushajte
- Obrišite par fileova, koristite git help da vidite kako obrisati fileove iz gita

# GIT - kreiranje repozitorija

## → Iz GitHuba

Kliknite na new u izborniku s desne strane

Dajte naziv repozitoriju

Odaberite opciju *Initialize this repository with a README*

## → Try it out:

Dodajte repo **vc1** na GitHub

# GIT - clone

## → Get a repo:

```
$ git clone https://github.com/Username/.....
```

## → Try it out:

Provjerite gdje se nalazite (kloniranje repozitorija u repozitorij nije dobra ideja)

U browseru otvorite <https://github.com/Lanietta/sparkschool-version-control>  
Klonirajte repozitorij **sparkschool-version-control**

## GIT - log & commit history

→ Za vidjeti kompletnu povijest nekog repozitorija koristimo **git log**

→ **Try it out:**

Potvrdite da se nalazite u repozitoriju **sparkschool-version-control**

`$ git log`

Pregledajte log

## GIT - branching

- **Branch** je pointer na neki commit
- Defaultni branch zove se **master**
- **Branch** se kreira komandom  
`git branch branch_name`

- **Try it out:**

`U repozitoriju sparkschool kreirajte branch testing`

## GIT - branching

- Za promijeniti branch koristimo komandu

```
git checkout branch_name
```

- Try it out:

```
Prebacite se na branch testing
```

```
Promijenite jedan file i commitajte
```

```
Prebacite se na branch master
```

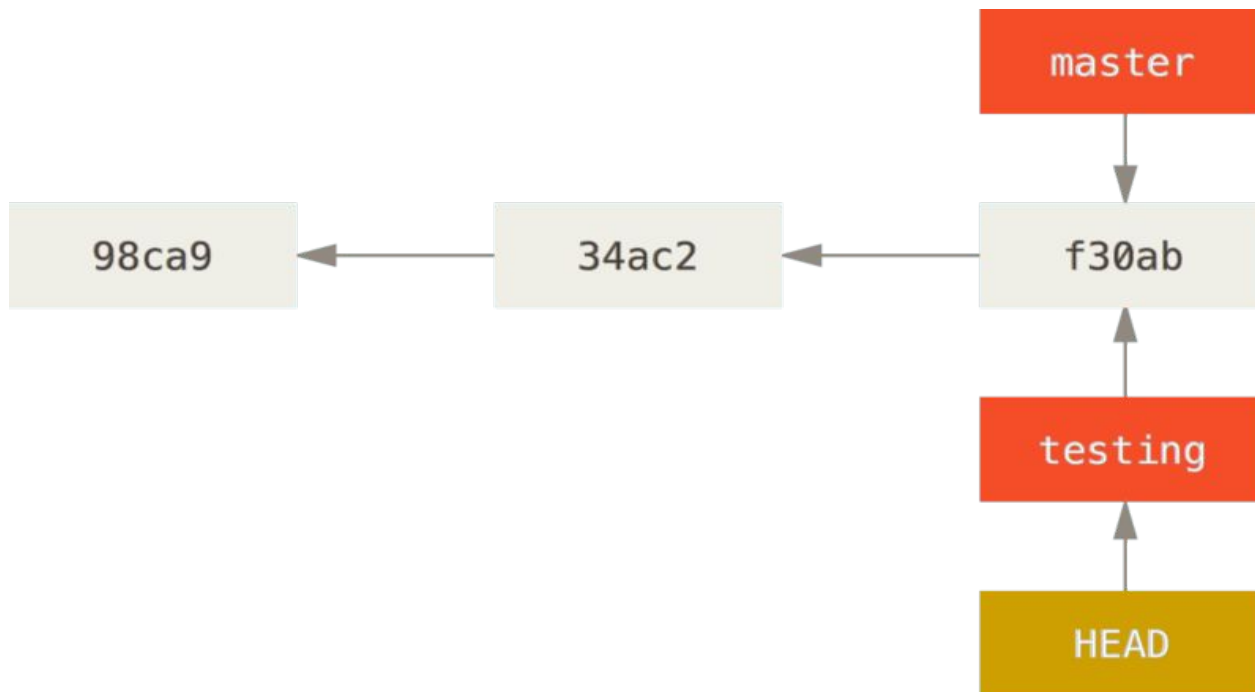
```
Promijenite jedan file i commitajte
```



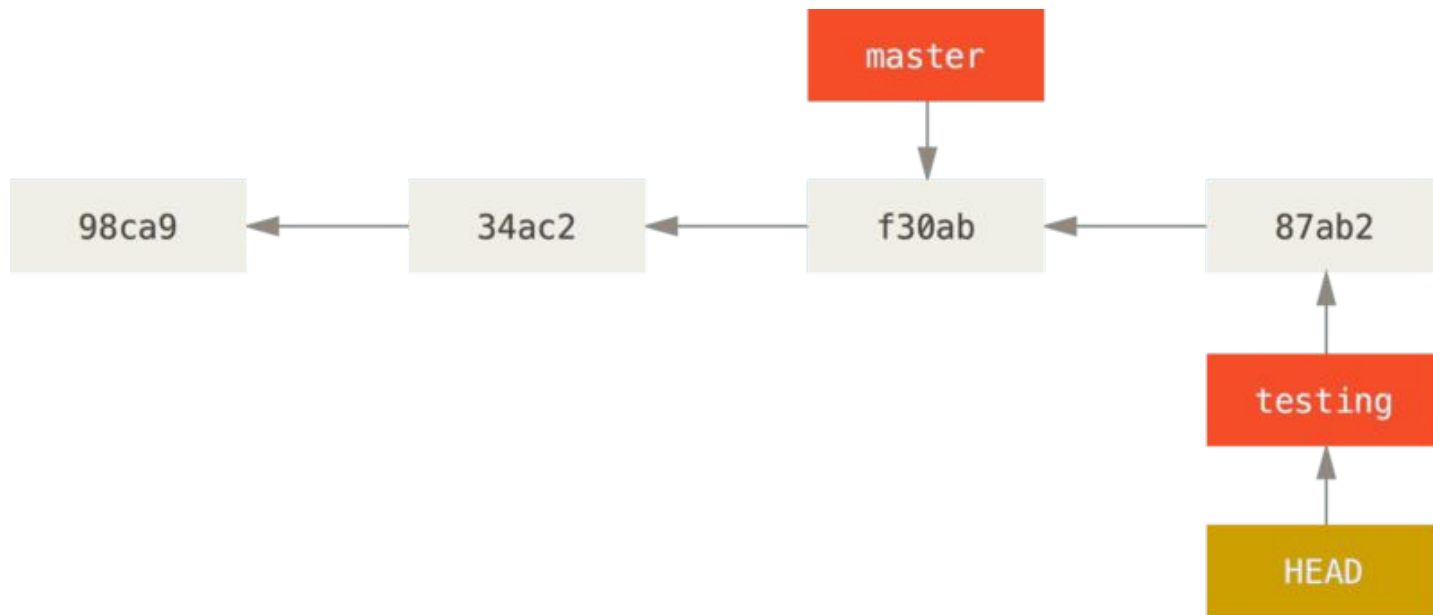
# GIT - branching - što smo uradili



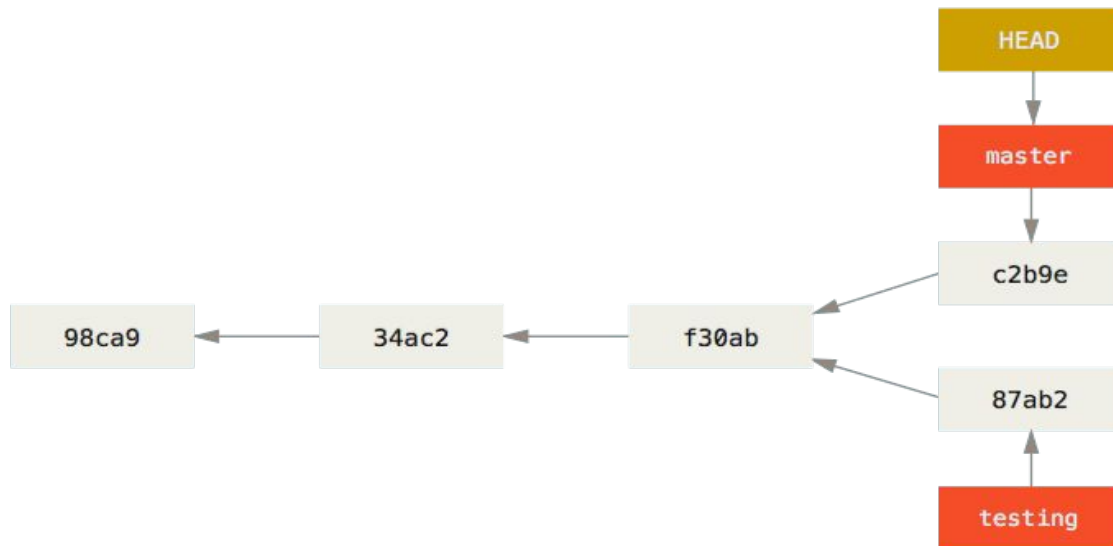
# GIT - branching - što smo uradili



# GIT - branching - što smo uradili



# GIT - branching - što smo uradili



## GIT - branching

→ Za kreirati branch na commitu koji nije latest koristimo komandu

```
git branch branch_name <commit_hash>
```

→ **Try it out:**

U repozitoriju sparkschool kreirajte branch **testing\_old** sa nekim starijim commitom

- Prebacite se na novi branch, kreirajte i commitajte file testing old
- `$ git push origin testing_old` će pushati promjene na remote repozitorij

## GIT - branching

- Za mergati dva brancha koristi se komanda  
`$ git merge branch_name`
- Merganje se radi sa brancha u koji želite mergati
- **Try it out:**

U repozitoriju **sparkschool** prebacite se na branch **master**

- Sa komandom **git branch** provjerite brancheve koje imate na repozitoriju
- Mergajte branch **testing** u branch **master**

## Playground

- U repozitoriju **sparkschool-version-control** koji ste klonirali ranije kreirajte branch naziva *prezime.ime*
- Na svom branchu kreirajte file naziva **didit.txt**
- U file upišite ime i prezime i što je git branch
- Commitajte file i pushajte ga na svoj branch
- Feel free to ask for help :)

## Something to remember

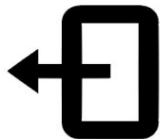
**In case of fire**



1. `git commit`



2. `git push`



3. `leave building`



POWERED BY:



THANK YOU  
for attention