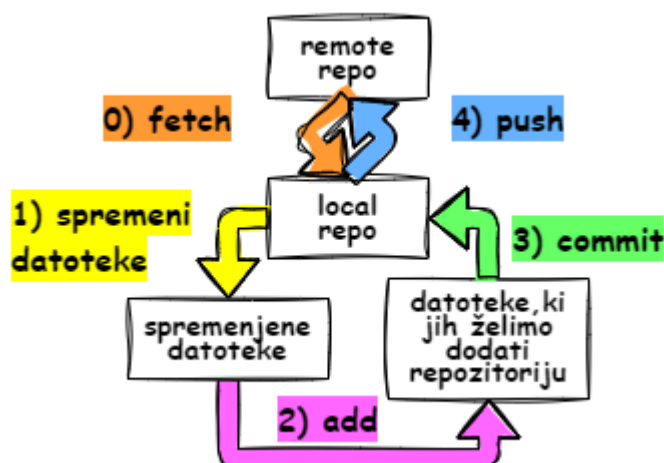
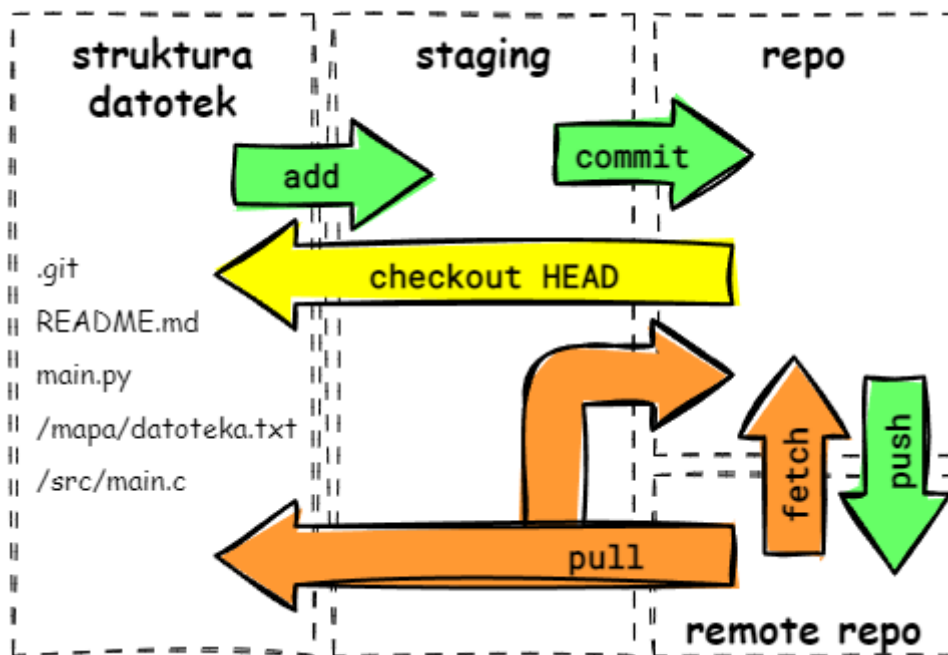


Primer na delavnici

Priprava

1. Install Git

2. Osnove



Ustvarjanje repozitorija

o. Ustvari repozitorij na GitHub.com

Nato preizkus ali je Git pravilno nameščen.

```
git --version
```

```
git --help
```

1. Ustvari projektno mapo

2. V ustvarjeni mapi z desnim klikom izberemo GIT Bash Here

3. Inicializiramo repozitorij

```
git init
```

Note: pojavi se mapa .git! ## 4. Dodaj nekaj datotek ### README.md

```
touch README.md
```

datoteka-1.txt

```
touch datoteka-1.txt
```

5 Dodajanje datotek v staging

Dodaj README.md

```
git add README.md
```

Preveri rezultat

```
git status
```

Note: datoteka-1.txt ni dodana v staging! Dodaj se vse ostale datoteke

```
git add *
```

Preveri de so sedaj dodane vse datoteke

```
git status
```

6 Ustvari prvi commit

```
git commit -m "Init repo"
```

6.1 Definiraj glavno vejo

```
git branch -M main
```

7 Dodaj oddaljeni repozitorij

```
git remote add origin git@github.com:Jure-D/olfs_git_delavnica.git
```

Note: po konvenciji glavni oddaljeni streznik poimenujemo origin, v teoriji lahko ima repozitorij vec oddaljenih streznikov npr: development servers in production z build sistemom ## 8 Poslji spremembe na oddaljeni streznik

```
git push -u origin main
```

Preveri rezultate z refresh!

Spremembe repozitorija

1 ustvari nekaj datotek

```
mkdir mapa_1 mapa_2
touch mapa_1/datoteka_1.txt
touch mapa_1/datoteka_2.txt
touch mapa_2/datoteka_1.txt
touch mapa_2/datoteka_2.txt
```

2 Ustvari novo vejo

```
git checkout -b veja-2
```

Note: nova veja postane aktivna! ## 3 Preizkusi dodati se eno vejo in preklopiti aktivno vejo

```
git branch veja-3      # 1e ustvari vejo
git checkout veja-3    # spremeni aktivno vejo
# NOTE: z '-b' enakovredno prejsnjemu ukazu
git branch -l
git branch -a
```

Na koncu preklopi na veja-2! ## 4 dodaj .gitignore datoteko

```
touch .gitignore
```

4.1 Razisci datoteko .gitignore

```
git add *
git status
```

```
git reset  
git add .gitignore
```

```
.gitignore
```

```
mapa_1/datoteka_1.txt
```

```
git add *  
git status
```

```
.gitignore
```

```
mapa_1
```

```
git reset  
git add .gitignore  
git add *  
git status
```

5 Dodaj nove datoteke

```
git commit -m "Added .gitignore"  
git push origin main  
git push origin veja-2
```

6 Dodaj nove spremembe v veji-2

Dodali bomo prej ustvarjeno mapo mapa-2, zato izbrisemo vsebino .gitignore in jo dodamo. ## 7 Spremeni aktivno vejo in spremljaj spremembe

```
git checkout main  
git checkout veja-2
```

8 Delo z vejami

Poglejmo trenutno stanje

```
git branch -a
```

Poglejmo pomoc za delo z vejami

```
git branch --help
```

Izbrisimo vejo veja-3

```
git branch -d veja-3
```

```
git branch -a
```

Ustvari novo vejo veja-2-copy, ki je osnovana na veji veja-2

```
git branch veja-2-copy
```

Poglej katere datoteke se nahajajo v novi veji

```
git checkout main
```

```
git checkout veja-2-copy
```

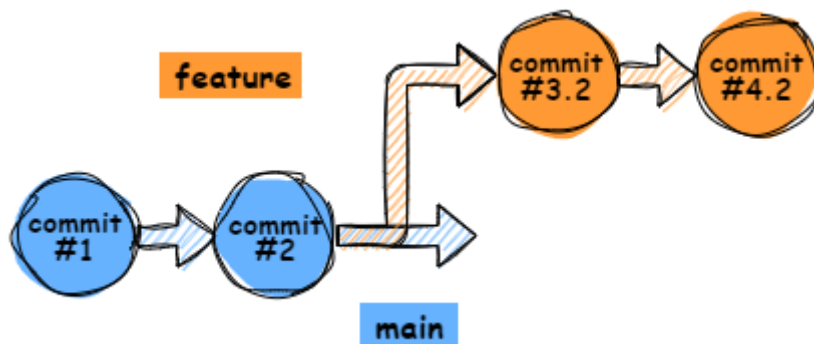
Novo vejo poslji se na oddaljeni strežnik

```
git push origin veja-2-copy
```

Preglej repozitorij na spletu, nato sinhroniziraj se spremembe v veji veja-2

```
git push origin veja-2
```

Združevanje vej



1 Združili bomo

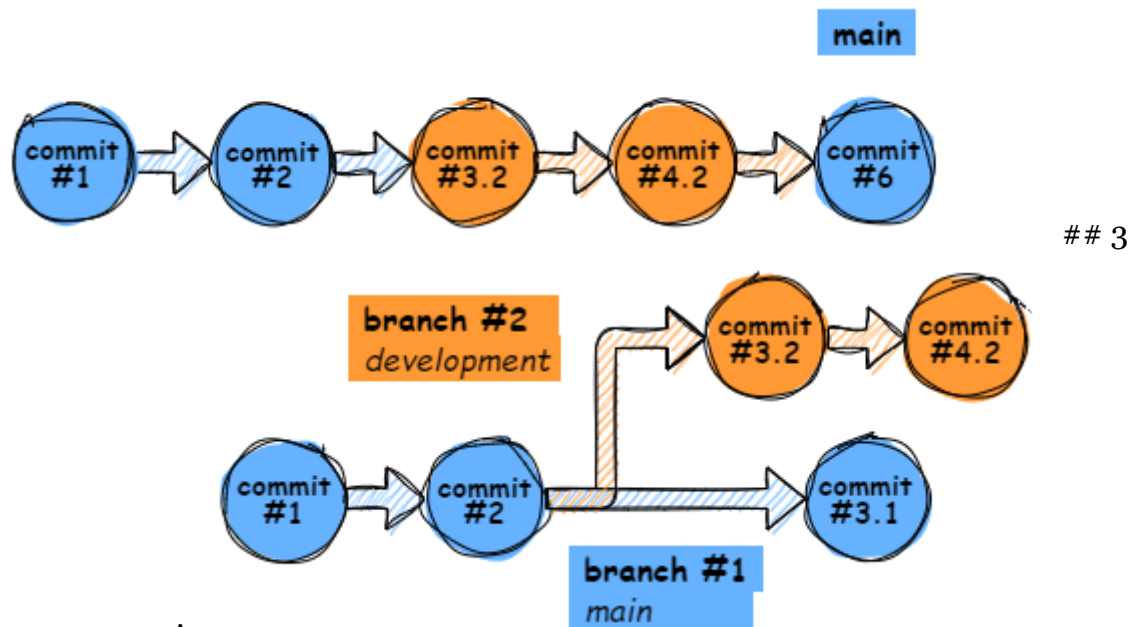
vejo veja-2 v vejo main

```
git checkout main
```

```
git merge veja-2
```

```
git push origin main
```

2 Fast forward merge



merge commit

3.1 Naredi spremembe v veji main

```
rm -r mapa_2
git add *
git status
git add mapa_2
```

```
mkdir app
touch app/dokument_1.txt
git add *
```

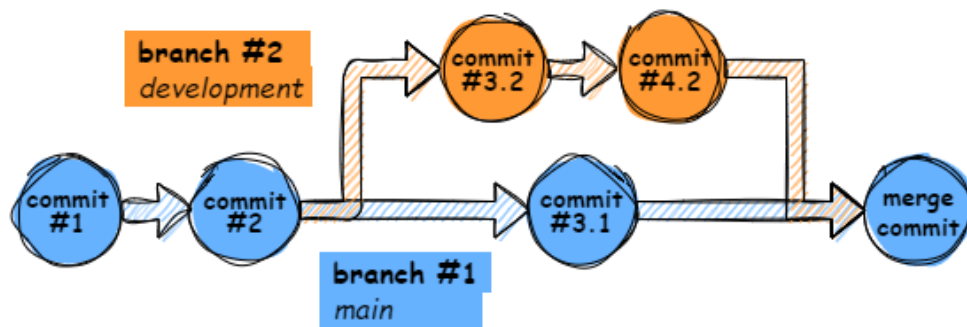
```
git commit -m "Spremembe v main"
git push origin main
```

3.2 Naredi spremembe v branch-2

```
git checkout veja-2
mkdir new_feature
touch new_feature/dokument_1.txt
git add *
git commit -m "Spremembe v feature"
git push origin veja-2
```

Izvedi merge commit

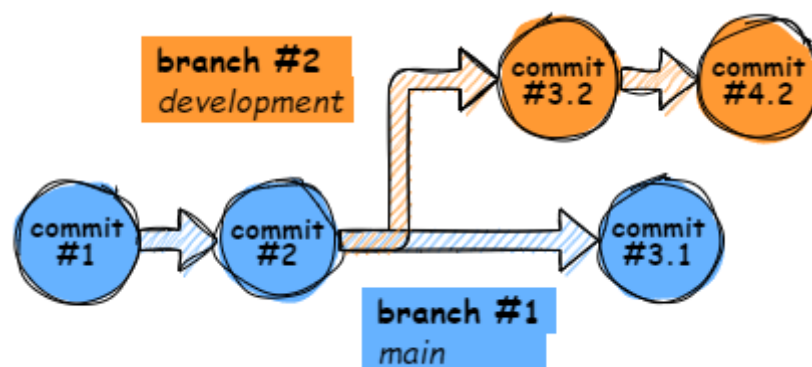
```
git checkout main
git merge veja-2
git push origin main
```



Slika merge commit rezultat

4 Rebasing

Stanje



Slika pred merge commit

4.1 Spremembe v main

```
touch app/dokument_2.txt
git add *
git commit -m "Nove spremembe v main"
git push origin main
```

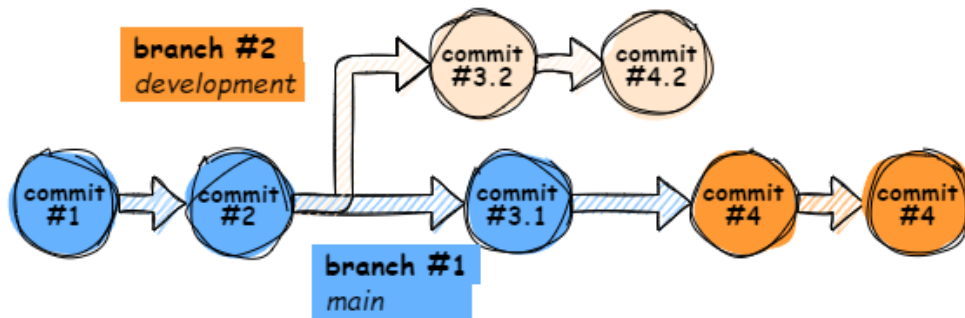
4.2 Ustvarimo spremembe v novi veji

```
git checkout -b new_feature_2
touch new_feature/super_new_feature.txt
git add *
git commit -m "Spremembe v super new feature"
git push origin new_feature_2
```

4.3 Izvedi rebase

```
git checkout new_feature_2
git rebase main
```

Rezultat



Slika rebase rezultat

Nadaljevanje: delo z drugimi uporabniki

Prenasanje sprememb na veji s streznika (kadar so spremembe v različnih datotekah ni tezav, sicer je potrebno ročno vnasati zeljene spremembe - obicajno se to pocne v GUI ali online).

```
git pull
```

Prenasanje nove veje s streznika

```
git fetch <ime_veje>
```

Prenesi novo verzijo repoitorija s streznika in jo nastavi kot lokalno verzijo

```
git pull --rebase
```