

# GitHub

- Kreiranje naloga
- Korišćenje iz Visual Studio Code-a
- Korišćenje iz komandne linije

Git predstavlja sistem za kontrolu verzija, na osnovu praćenja promena datoteka i omogućava jednostavan rad više ljudi na projektima uz lako deljenje datoteka. Slobodan je softver i osnova je za razne distribuirane sisteme kontrole verzija, kako lokalnih, na različitim operativnim sistemima, tako i web baziranih.

Jedan od web servisa za hostovanje koda i statičkih web stranica, u vlasništvu Microsoft-a je i GitHub. Pošto se bazira na Git sistemu, potrebno je imati ga instaliranog na računaru, pored naloga na GitHub platformi. Na narednim stranicama, detaljnije će biti objašnjena procedura instalacije potrebnih alata i kreiranja naloga, kao i korišćenja GitHub hosting mogućnosti iz okruženja Visual Studio Code, kao i komandne linije.

## Git instalacija

Pošto si i Git i Visual Studio Code cross platform tehnologije, a jedini korak koji se razlikuje u korišćenju GitHub-a je instalacija Git softvera, u nastavku će biti reči o instalaciji na operativnim sistemima Windows, Linux i Mac.

### Windows

Git je alat koji se instalira automatski uz veliki broj okruženja ili softvera. Da li je instaliran moguće je proveriti komandom **git --version** u Command Prompt-u Windows-a. Ukoliko komanda ne bude pronađena, potrebno je instalirati ga. Instalaciju je moguće pronaći na sledećoj adresi: <https://git-scm.com/download/win>. Potrebno je samo ispratiti postupak instalacije, koji je jednostavan. Jedini korak na koji treba obratiti pažnju je *Adjusting your PATH Environment*, gde je najbolje obeležiti *Run Git from the Windows Command Prompt*. Ni ostale opcije ne izazivaju veće probleme u korišćenju alata koji dolaze sa ovom instalacijom, ali je putanju do instalacije potrebno naknadno podesiti u PATH promenljivoj, što može da bude zaobiđeno na ovaj način.

### Linux

Veliki broj korisnika Linux-a koristi neku od Debian based distribucija, od kojih je jedna Ubuntu Linux. Zato će u ovom odeljku biti prikazana procedura za

instalaciju upravo na ovim distribucijama. Procedura se ne razlikuje od ostalih distribucija koje imaju svoje interfejsa za instalaciju softvera, zato što su paketi istih naziva, samo je potrebno koristiti odgovarajući interfejs za konkretnu distribuciju. Takođe je moguće instalirati sve potrebne alate iz souce code-a, koji je dostupan na nekoliko adresa, od kojih je jedna na samom GitHub-u: <https://github.com/git/git/releases>. Ukoliko ipak imate instaliran jednu od Debian based distribucija, onda ste verovatno upoznati sa Advanced Package Tool alatom. Instalacija se korišćenjem ovog alata svodi na par koraka:

- |   |  |
|---|--|
| - <code>sudo apt-get update</code>      | - komanda koja update-uje liste paketa |
| - <code>sudo apt install git-all</code> | - instalacija svih potrebnih alata     |

## Mac

Instalacija na Mac operativnim sistemima je takođe jednostavna. Ukoliko je instaliran XCode IDE, Git je deo te instalacije, pa nema potrebe instalirati ga naknadno. Ukoliko nije, postoji nekoliko načina:

1. Homebrew: `brew install git`

2. Korišćenjem binarnog instalacionog fajla sa adrese: <https://sourceforge.net/projects/git-osx-installer/>

## Kreiranje GitHub naloga

Adresa na kojoj je neophodno kreirati nalog za korišćenje GitHub usluga je sledeća: <https://github.com/join>.

Na toj stranici neophodno je popuniti podatke o korisničkom imenu (Username), E-mail adresi koja će biti korišćena za ovaj nalog (bilo koja e-mail adresa) i šifre. Potrebno je i verifikovati nalog, na dnu stranice, a zatim i putem e-mail-a.

# Create your account

Username \*

Email address \*

Password \*

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.

[Learn more.](#)

## Email preferences

☐ Send me occasional product updates, announcements, and offers.

## Verify your account

Nakon par pitanja na narednoj stranici, na koje je moguće i ne odgovoriti i verifikacije e-mail adrese, nalog je kreiran. Podatke o nalogu je moguće pregledati i promeniti klikom na ikonicu u gornjem desnom uglu.



Nakon što je nalog kreiran, klikom na dugme + koje se nalazi levo od ikonice korisnika moguće je kreirati repozitorijum, na sledeći način:

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner \*



darkopuflovic ▾

Repository name \*

/ RomanNumbers ✓

Great repository names are short and memorable. Need inspiration? How about [shiny-octo-garbanzo?](#)

Description (optional)

Roman Number Conversions



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



Add a README file

This is where you can write a long description for your project. [Learn more.](#)



Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)



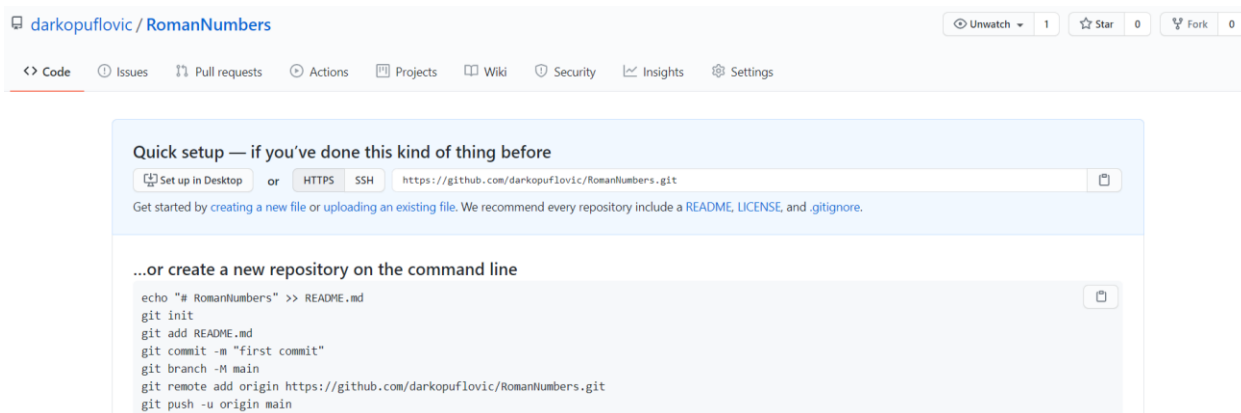
Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Potrebno je uneti ime repozirorijuma, opis, koji nije obavezan i odabrati da li će on biti privatan ili javan. Takođe je moguće odabrati i da li će biti kreirane datoteke README, .gitignore i pridodata licenca, ali nećemo detaljnije govoriti o nameni ovih opcija.

Popunjavanjem informacija o repozitorijumu i klikom na dugme *Create Repository*, repozitorijum je kreiran i prikazuje se sledeća stranica:



Na sredini se nalazi adresa do našeg repozitorijuma (<https://github.com/username/nazivrepozitorijuma.git>). Ova adresa će biti potrebna za dalju konfiguraciju, o čemu će biti reči u nastavku.

Brisanje repozitorijuma je moguće klikom na dugme Settings (poslednje dugme sa desne strane na vrhu stranice. Pored promene naziva repozitorijuma, na dnu stranice Options u uokvirenom delu sa nazivom Danger Zone se nalazi i opcija Delete This Repository sa istoimenim dugmetom, kojim je moguće izbrisati repozitorijum.

## Konfigurisanje repozitorijuma iz komandne linije

Komande koje mogu da se koriste iz komandne linije omogućavaju podešavanje trenutnog repozitorijuma, ali se koriste i za globalno konfigurisanje koje je potrebno uraditi samo jednom. Preporučuje se da se ovo podešavanje naloga uradi iz komandne linije i to na sledeći način:

- Konfiguracija korisničkog imena i e-mail adrese koja će se koristiti:
  - o **git config --global user.name "korisničko ime"**
  - o **git config --global user.email "email-adresa"**

Adresa i korisničko ime su one koje su korišćene za kreiranje GitHub naloga.

Za inicijalizaciju repozitorijuma u folderu projekta koristi se jedna od sledeće 2 komande:

- **git init** ili
- **git clone [url]** gde [url] predstavlja adresu sa koje se vrši kloniranje repozitorijuma

Za proveru modifikovanih datoteka, moguće je koristiti **git status** komandu. Ukoliko takvih datoteka ima i treba ih dodati u sledeću verziju na GitHub-u treba koristiti komandu **git add [file]** ili **git add .** koje dodaju određeni fajl ili sve fajlove iz direktorijuma (redom).

Izbacivanje fajla iz promena za sledeću verziju se vrši komandom **git reset [file]**.

Da bi promene bile poslate na server, potrebno je pozvati komandu **git commit -m "Opis promene"**. Opis promene je potreban da bi drugi korisnici ili sam korisnik imali predstavu šta je to izmenjeno u konkretnoj verziji fajla.

## Kreiranje grana

Korist Git alata leži i u tome što je omogućeno kreiranje grana. Grane su različite verzije projekta, koje se nezavisno razvijaju i koje se mogu spojiti sa glavnim projektom u bilo kom trenutku razvoja.

Komanda koja se koristi za grananje je **git branch "branch name"**. Ovom komandom se kreira nova grana sa imenom "branch name". Da bi se izlistale sve grane koje u projektu postoje moguće je koristiti **git branch** komandu bez parametara. Na taj način se prikazuju sve grane, dok je aktivna grana označena znakom \*.

Prebacivanje iz jedne u drugu granu (aktivne grane), vrši se komandom **git checkout**. Ova komanda pribavlja fajlove koji su potrebni za rad u grani koja je odabrana.

Komanda **git merge "branch"** se koristi za spajanje grane "branch" sa aktivnom granom.

## Povezivanje preko URL-a sa remote serverom

Za povezivanje sa GitHub repozitorijumom potrebno je koristiti sledeće komande:

- **git remote add "alias" "URL"** - Dodavanje Git URL-a (GitHub URL-a)
- **git fetch "alias"** - Pribavljanje svih grana sa remote Git-a
- **git merge "alias"/"branch"** - Spajanje udaljene grane u trenutnu lokalnu granu

- **git push "alias" "branch"** - Slanje lokalne grane na udaljenu repozitorijum
- **git pull** - Pribavljanje izmena sa udaljenog repozitorijuma na lokalni

Još par korisnih komandi koje mogu da se koriste su:

- **git rebase "branch"** - Kombinovanje više commit-ova u novu granu sa nazivom "branch"
- **git reset --hard "commit"** - Prepisivanje svih izmena iz odgovarajućeg commit-a "commit"

## Korišćenje Git-a iz Visual Studio Code-a

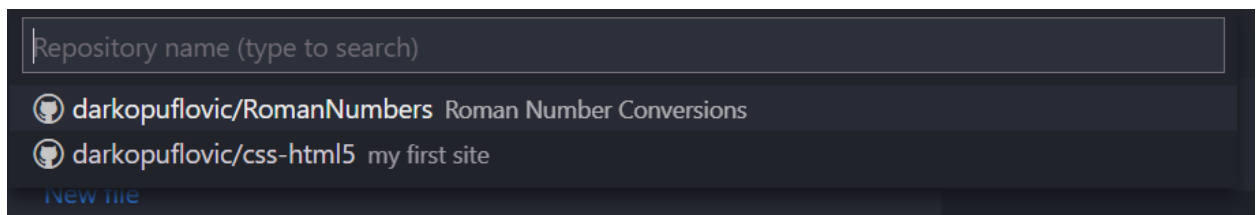
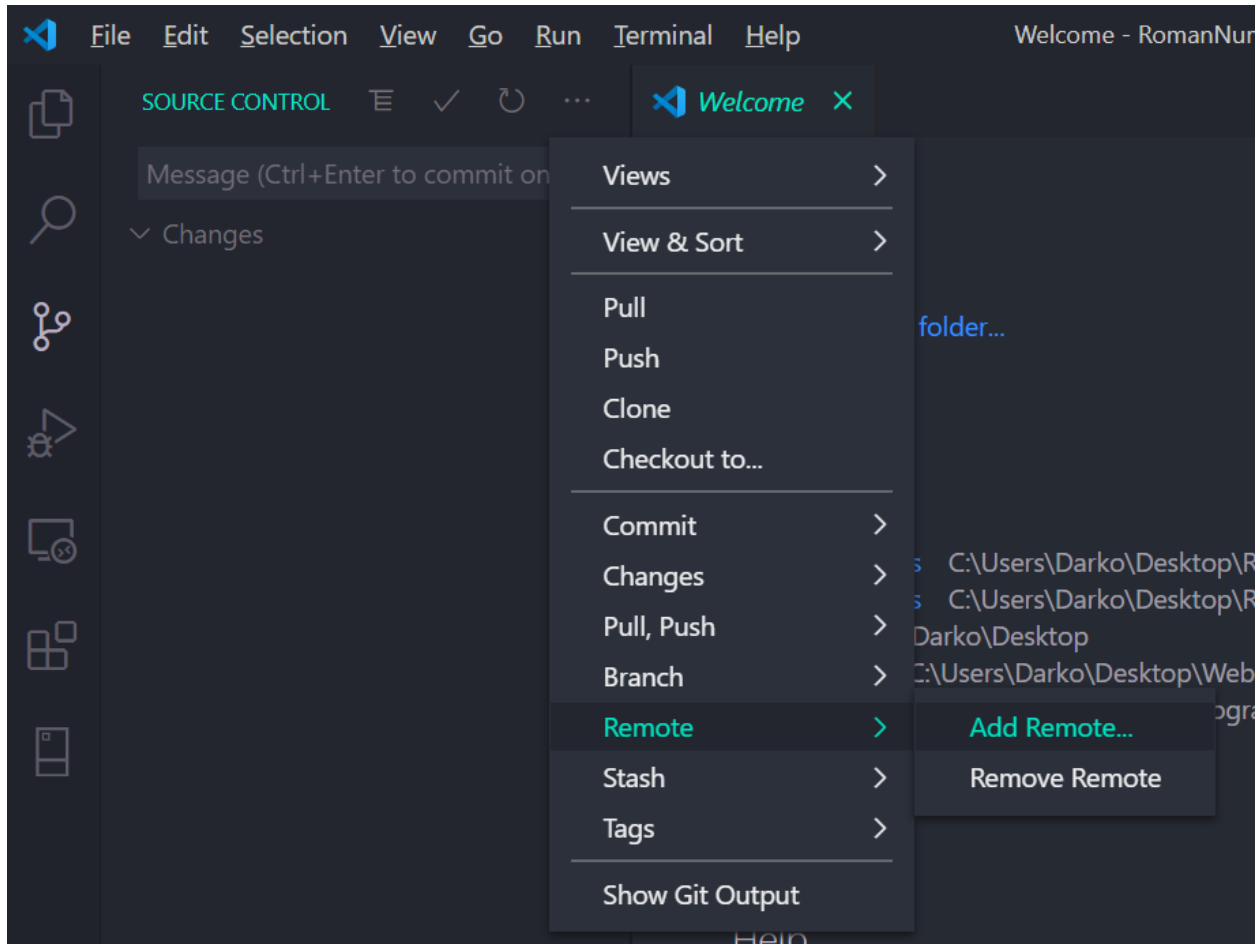
Prethodno prikazane komande se mogu koristiti i iz Visual Studio Code-a. Ipak, u nekim situacijama je potrebno znati i koju komandu iz komandne linije je potrebno znati da bi se pokrenula odgovarajuća akcija. Zato su prvo objašnjene komande komandne linije, a sada ćemo govoriti i o jednostavnijem načinu korišćenja najpotrebnijih za kreiranje projekata i povezivanje sa GitHub-om.

Prvi korak je kreiranje direktorijuma u kome će projekat da se nalazi i pokretanje Visual Studio Code-a na toj adresi. Nakon toga, direktorijum je potrebno povezati sa GitHub repozitorijumom (koji je kreiran ranije). To se može ostvariti na sledeći način:

- Klik na dugme *Source Control* (3. dugme u sidebar-u sa leve strane, nalik na graf)
- Klikom na ... u gornjem desnom uglu tog prozora otvara se meni koji nudi Git opcije, među kojima je i Remote... opcija koja ima podmeni Add Remote...
- Nakon toga otvara se mali prozor koji ima jednu opciju, Add remote from GitHub...
  - Klikom na ovu opciju, otvara se prozor koji autorizuje Visual Studio Code da koristi nalog na GitHub-u i pristupa njegovim



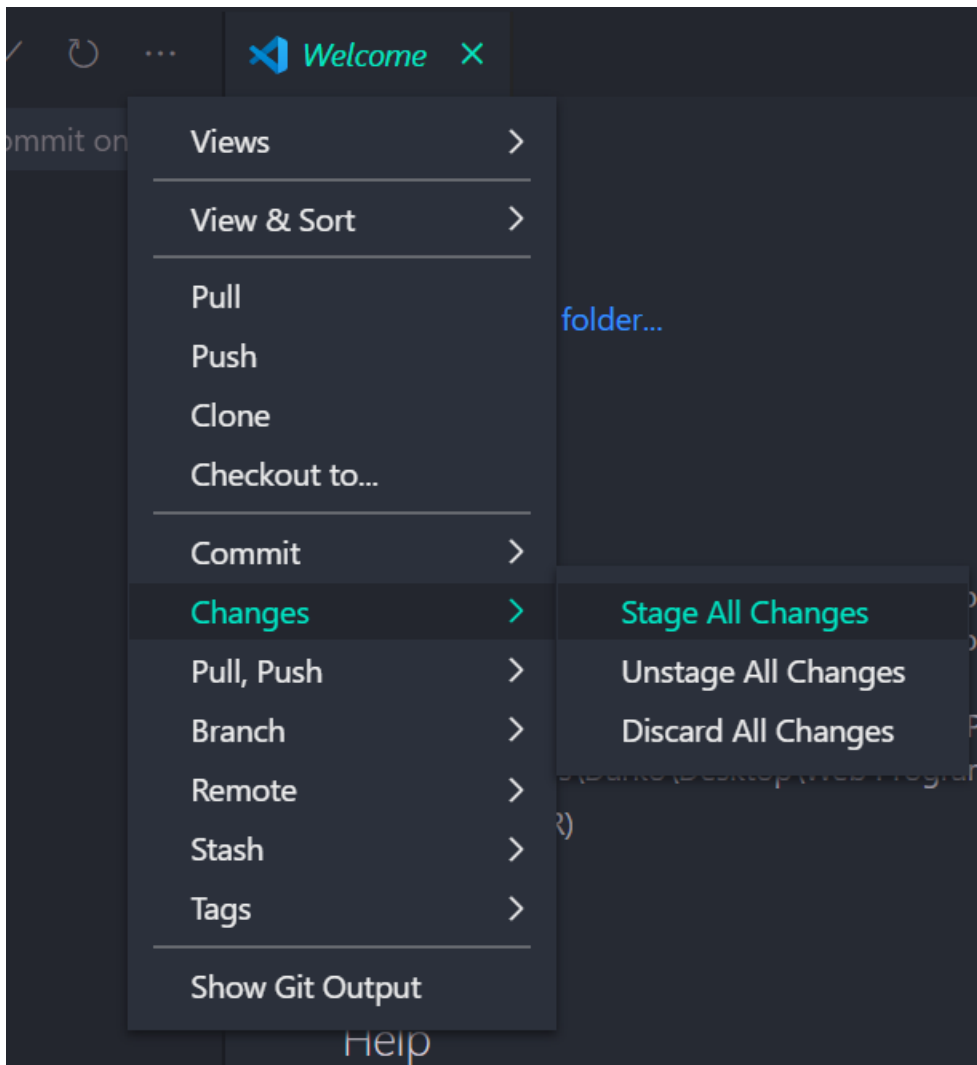
repozitorijumima. Nakon autorizacije (par klikova u browser-u u kome je potrebno biti ulogovan na nalog koji se koristi) i u Visual Studio Code-u koji potvrđuje autorizaciju, otvara se lista repozitorijuma, gde treba odabrati odgovarajući sa kojim treba raditi.



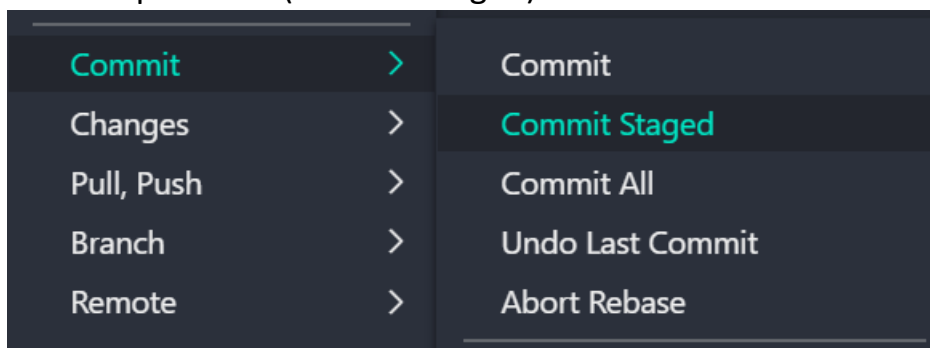
- Nakon ovih koraka, GitHub je povezan. Sada je moguće početi sa izradom projekta, gde se nakon svake bitne promene mogu koristiti komande iz prethodno prikazanog menija, koje odgovaraju komandama opisanim u korišćenju Git-a iz komandne linije.

Postupak za slanje izmena na GitHub je sledeći:

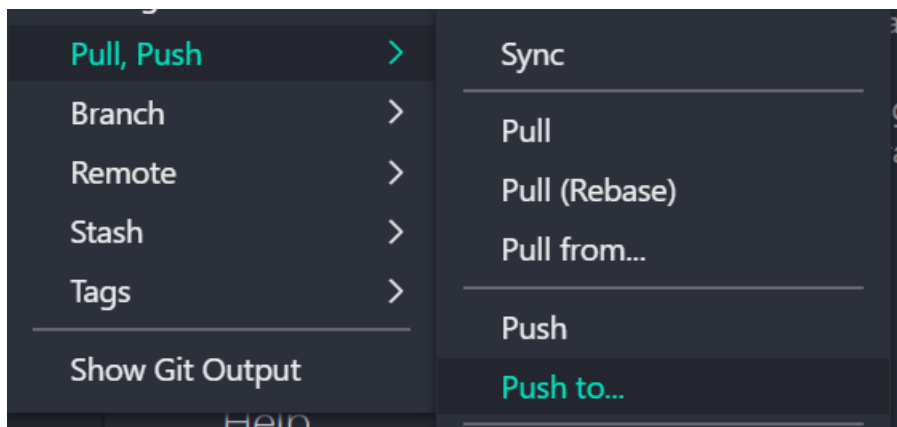
- Pripravljanje promena za slanje na server: *Stage All Changes*



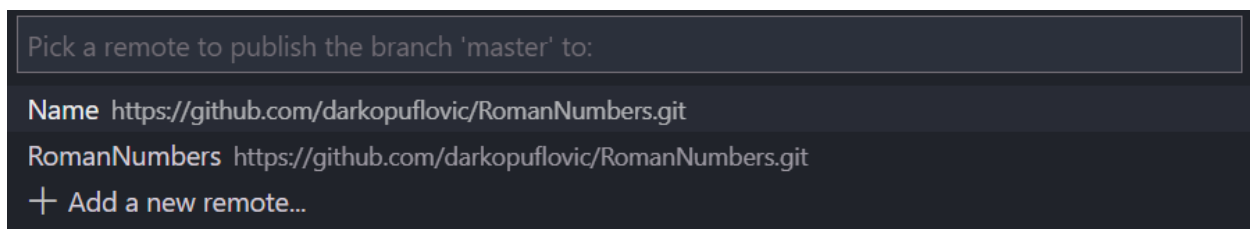
- Commit promena (Commit Staged)



- Konačno, slanje na GitHub



- Push to..., otvara se meni koji nudi repozitorijume dostupne na nalogu koji je povezan. Zatim se odabirom odgovarajućeg repozitorijuma vrši slanje izmena na server.



Moguće je da će prilikom odabira repozitorijuma da se pojavi još jedan prozor u donjem desnom delu prozora, koji nudi Yes i No opcije za periodično prikupanje (Pull) izmena sa GitHub-a. Obe opcije su adekvatne, ali treba imati u vidu da ukoliko se odabere No, potrebno je pre slanja prve promene u repozitorijum pozvati Pull ručno iz ovog menija. Ukoliko se odabere opcija Yes, ona to radi sama, ali ukoliko ima više od jednog kolaboratora na projektu, to može u nekim slučajevima da bude problem.

## Hostovanje statičke web stranice na GitHub Pages

Još jedna mogućnost koju GitHub nudi je hostovanje statičkih web stranica. Sve stranice koje se već nalaze na serveru, mogu da se jednostavno podese, tako da im je moguće pristupiti iz browser-a i pregledati. Procedura se sastoji od svih koraka koji su pobrojani ispred (da bi se kod poslao na GitHub). Nakon toga, sa Settings stranice je moguće podesiti da kod koji je upravo poslat na GitHub, može da se pregleda kao da je reč o live serveru. Sada ćemo u par koraka i slika da vidimo kako se to ostvaruje.

1. Klikom na Settings ikonicu, koju smo pominjali i prilikom brisanja repozitorijuma, otvara stranicu sa podešavanjima. U Options tab-u, gde se nalazi i Delete Repozitorijum opcija, nalazi se i sekcija GitHub Pages.

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

**Source**  
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Save

**Theme Chooser**  
Select a theme to publish your site with a Jekyll theme using the gh-pages branch. [Learn more.](#)

Choose a theme


2. Kao Source je potrebno odabrati granu koja sadrži kod koji hoćemo da hostujemo na GitHub Pages.


## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at <https://darkopuflovic.github.io/ColorsWebPage/>.

**Source**  
Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

 Branch: master ▾

 / (root) ▾

Save

**Theme Chooser**  
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

**Custom domain**  
Custom domains allow you to serve your site from a domain other than darkopuflovic.github.io. [Learn more.](#)

Save

☒ **Enforce HTTPS**  
— Required for your site because you are using the default domain (darkopuflovic.github.io)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

3. Klikom na dugme Save, kreira se GitHub strnica, koja je dostupna na adresi koja je prikazana na početku ove sekcije. Ovom linku je moguće pristupiti iz web browser-a.