[*Sadržaj*]

Predgovor **xi**

Poglavlje 1 **Git i komandna linija** **1**

Šta je Git? 1

Grafički korisnički interfejs i komandna linija 2

Otvaranje prozora komandne linije 3

Izvršavanje komandi na komandnoj liniji 5

Instalacija Gita 7

Opcije i argumenti komandi 7

Čišćenje komandne linije 8

Otvaranje prozora sistema datoteka 9

Rad sa direktorijumima 9

Zatvaranje komandne linije 16

Postavljanje Git konfiguracija 17

Priprema tekst editora 19

Integrisani terminali 19

Rezime 20

Poglavlje 2 **Lokalna skladišta** **21**

Tekuća postavka 21

Upoznavanje sa skladištima 21

Inicijalizacija lokalnog skladišta 22

Oblasti Gita 26

Dodavanje datoteke u Git projekat 30

Rezime 32

Poglavlje 3 **Pravljenje komita** **33**
Tekuća postavka 33
Zašto pravimo komite? 34
Dva koraka za pravljenje komita 34
Prikazivanje liste komita 40
Rezime 42

Poglavlje 4 **Grane** **43**
Stanje lokalnog skladišta 43
Zašto koristimo grane? 44
Neizmenjene i izmenjene datoteke 48
Pravljenje komita na grani 51
Pravljenje grane 53
Šta je HEAD? 55
Prebacivanje sa grane na granu 57
Rad na odvojenoj grani 61
Rezime 62

Poglavlje 5 **Spajanje** **63**
Stanje lokalnog skladišta 63
Uvod u spajanje 64
Tipovi spajanja 64
Spajanje nadovezivanjem 69
Provera komita 79
Kreiranje grane i prebacivanje na nju u jednom koraku 85
Rezime 86

Poglavlje 6 **Hosting servisi i autentifikacija** **87**
Hosting servisi i udaljena skladišta 88
Postavljanje naloga na hosting servisu 88
Postavljanje podataka za autentifikaciju 89
Rezime 91

Poglavlje 7 **Kreiranje i slanje podataka na udaljeno skladište** **93**
Stanje lokalnog skladišta 93
Dva načina za početak rada na Git projektu 94
Interakcija između lokalnih i udaljenih skladišta 96
Zašto koristimo udaljena skladišta? 96
Kreiranje udaljenog skladišta sa podacima 97
Rad na udaljenom skladištu direktno na hosting servisu 111
Rezime 112

Poglavlje 8 **Kloniranje i preuzimanje** **113**
Stanje lokalnih i udaljenih skladišta 113
Kloniranje udaljenog skladišta 114
Brisanje grana 122
Git saradnja i grane 125
Inkorporisanje promena iz udaljenog skladišta 131
Brisanje grana (nastavak) 136
Rezime 138

Poglavlje 9 **Trosmerno spajanje** **139**
Stanje lokalnog i udaljenog skladišta 139
Zašto su trosmerna spajanja važna? 141
Postavljanje scenarija za trosmerno spajanje 144
Definisanje uzvodnih grana 145
Izmena iste datoteke više puta između komita 149
Rad u isto vreme sa drugima na različitim fajlovima 156
Trosmerno spajanje u praksi 159
Povlačenje promena iz udaljenog skladišta 165
Stanje lokalnih i udaljenih skladišta 168
Rezime 169

Poglavlje 10 **Konflikti spajanja** **171**
Stanje lokalnih i udaljenih skladišta 171
Uvod u konflikte spajanja 173
Kako rešiti konflikte spajanja 174

Postavljanje scenarija konflikta spajanja 175
Proces rešavanja sukoba prilikom spajanja 181
Rešavanje sukoba prilikom spajanja u praksi 183
Ostanite u toku sa udaljenim skladištem 186
Sinhronizacija skladišta 187
Stanje lokalnih i udaljenih skladišta 189
Rezime 190

Poglavlje 11 **Promena baze 191**
Stanje lokalnog i udaljenog skladišta 191
Integracija izmena u Gitu 193
Zašto je rebaziranje korisno? 194
Postavljanje primera za promenu baze 197
Uklanjanje i dodavanje fajlova u prostor pripreme 199
Priprema za promenu baze 209
Pet faza procesa promene baze 211
Konfikti promene baze i spajanja 215
Promena baze grane u praksi 216
Zlatno pravilo promene baze 221
Sinhronizacija skladišta 224
Stanje lokalnog i udaljenog skladišta 227
Rezime 227

Poglavlje 12 **Zahtevi za povlačenje (zahtevi za spajanje) 229**
Stanje lokalnih i udaljenih skladišta 229
Upoznavanje sa zahtevima za povlačenje 231
Specifičnosti hosting servisa 232
Zašto koristiti zahteve za povlačenje? 233
Kako se spajaju zahtevi za povlačenje 235
Priprema zahteva za povlačenje 238
Lakši način za definisanje uzvodnih grana 240
Kreiranje zahteva za povlačenje na hosting servisu 243
Pregledanje i odobravanje zahteva za povlačenje 244

Spajanje zahteva za povlačenje 246
Brisanje udaljenih grana. 247
Sinhronizacija lokalnih skladišta i čišćenje. 249
Stanje lokalnih i udaljenih skladišta. 253
Rezime 254

Epilog **255**

Dodatak A: Potrebna podešavanja po poglavljima. **257**

Dodatak B: Spisak naredbi. **281**

Dodatak C: Reference vizuelnog jezika **285**

Indeks **289**

[*Predgovor*]

Nikada nisam mislila da ću pisati knjigu o Gitu. Ali kroz srećan niz događaja, došla sam do kreativne ideje kako bih mogla jednostavno predstaviti tehnologiju.

Moje putovanje je počelo kada sam pohađala centar za obuku za programiranje da bih učila web razvoj. Profesori u centru za obuku su ukratko predstavili Git studentima, ali pošto su svi naši projekti bili individualni, nismo ga morali koristiti intenzivno.

Nakon centra za obuku za programiranje, dobila sam posao kao junior frontend stručanjak za razvoj na veb sajtu u velikoj kompaniji. Moj pravi put učenja sa Gitom počeo je prvog dana mog novog posla. U tim prvim mesecima, radeći kao deo tima u velikoj kompaniji, shvatila sam da me je to plašilo. Svaki put kad sam morala da uradim nešto što je izgledalo iole komplikovano koristeći Git, mislila sam da ću uništiti skladište ili ozbiljno nešto pokvariti.

Da bih mogla pravilno da radim sa kolegama, odlučila sam da sam naučim Git. Ali dok sam čitala različite onlajn resurse, brzo mi je postalo jasno da većina materijala tamo nije dizajnirana za ljude koji tek počinju. Kada sam shvatila osnove, počela je da se formira ideja u mojoj glavi kako bih mogla jednostavnije da naučim ovu tehnologiju koristeći vizuale i boje.

Napokon sam kreirala onlajn kurs koji sam postavila na web. Dok sam radila na kursu, u pozadini sam razmišljala da bih jednog dana mogla napisati i knjigu o tome.

Dobila sam mnogo pozitivnih povratnih informacija o kursu i konačno, leta 2021, odlučila sam da je došlo vreme da počnem sa tim projektom. Knjiga koju sada čitate je proizvod te odluke i nadam se da će vam pomoći na vašem putu učenja o Gitu!

Kome je ova knjiga namenjena

Ova knjiga je za svakoga ko želi da uči osnove kako Git funkcioniše. Posebno je dizajnirana za pojedince koji tek počinju da uče tehničke veštine, ili za one koji rade na netehničkim pozicijama ali moraju koristiti Git da sarađuju sa svojim tehničkim kolegama. Neki primeri

pojedinaca koji bi mogli imati koristi od ove knjige uključuju (ali nisu ograničeni na) studente centra za obuku za programiranje, studente računarske nauke, tehničke pisce, menadžere proizvoda, dizajnere, junior razvojne radnike, naučnike podataka i samouke programere.

Knjiga je napisana za ljude bez iskustva u korišćenju Gita, kao i za one sa malo iskustva u korišćenju Gita. Ako nemate iskustva sa Gitom, to nije problem jer ova knjiga počinje od nule. Počećemo sa instaliranjem Gita i kako se koristi komandna linija i odande graditi.

Ako već imate neko iskustvo sa korišćenjem Gita ili komandne linije, prvo poglavlje može biti kratak pregled materije. Međutim, podstičem vas da ga ne preskačete jer postavlja projekat Rainbow koji će se koristiti kroz ostatak knjige.

Korišćenje ove knjige

Ova knjiga je praktično iskustvo učenja, gde ćete izvoditi vežbe na vašem računaru dok učite osnovne koncepte Gita. Kroz knjigu ćete naići na dva projekta: Projekat Rainbow i Projekat knjige.

Projekat Rainbow je praktičan projekat na kom ćete raditi prolazeći kroz vežbe u knjizi. To je pojednostavljen projekat namenjen samo u svrhu učenja. Projekat knjige je zamišljen projekat koji ću koristiti da demonstriram kako se određene karakteristike Gita mogu koristiti za realan projekat. Hajde da pogledamo bliže svaki od ovih projekata, kao i kako je knjiga strukturisana.

[NAPOMENA]

Ne brinite ako ovaj uvod sadrži terminologiju sa kojom još niste upoznati, poput *skladišta* i *komita*. Objasniću sve ove koncepte u narednim poglavljima.

PROJEKAT RAINBOW

Da biste naučili osnove Gita, kroz ovu knjigu ćete raditi na Projektu Rainbow. Da biste pratili, trebalo bi da čitate knjigu od Poglavlja 1 do Poglavlja 12 linearno i trebalo bi da završite svaku vežbu na vašem računaru. Na primer, vežbe u Poglavlju 4 će pretpostaviti da ste već završili vežbe u Poglavljima 1, 2 i 3.

Skladišta

Na osnovnom nivou, *skladište* je kopija Git projekta. Prvo ćete kreirati jedno lokalno skladište pod nazivom `rainbow` za rad na projektu Rainbow. Naknadno (u Poglavlju 7), kreiraćete udaljeno skladište pod nazivom `rainbow-remote`. I konačno, u Poglavlju 8, simuliraćete da sarađujete sa koautorom na projektu Rainbow i kreiraćete drugo lokalno skladište pod nazivom `friend-rainbow`. Od Poglavlja 8 nadalje, kad god se napravi referenca na vašeg „saradnika" koji nešto radi, moraćete izvesti akciju u skladištu `friend-rainbow`.

> **[NAPOMENA]**
>
> Kada se pozivam na projekat Rainbow sa velikim R, mislim na ceo projekat koji počinje sa jednim skladištem i na kraju sadrži tri skladišta. Kada se pozivam na direktorijum projekta `rainbow` ili skladište `rainbow` (sa malim r), mislim na specifično lokalno skladište koje je deo projekta Rainbow.

Komiti

U projektu Rainbow, kreiraćete i urediti datoteke kako biste naveli boje duge i neke boje koje nisu deo duge. Ovo nije namenjeno da bude realističan primer projekta koji je kontrolisan verzijama od strane Gita. To je pojednostavljen projekat koji vam omogućava da se fokusirate na učenje umesto na kreiranje nečeg komplikovanog.

Kroz knjigu, koristiću dijagrame da ilustrujem šta se dešava u projektu Rainbow. Svaki put kada dodate boju projektu Rainbow, izvršićete *komit* u skladištu. Komit u osnovi predstavlja verziju vašeg projekta. Na dijagramima, komit će biti predstavljen krugom u boji koju ste dodali i koristićemo ime boje kao ime komita takođe. Na primer, prva boja koju ćete dodati projektu Rainbow je crvena, pa će krug koji će predstavljati taj komit biti obojen crveno i pozivaćemo se na njega kao na crveni komit.

Prevod ove knjige nije štampan u koloru pa će od pomoći biti ispisano ime komita (ili skraćenica imena) u dijagramima. Vidite Sliku P-1 za primer crvenog komita.



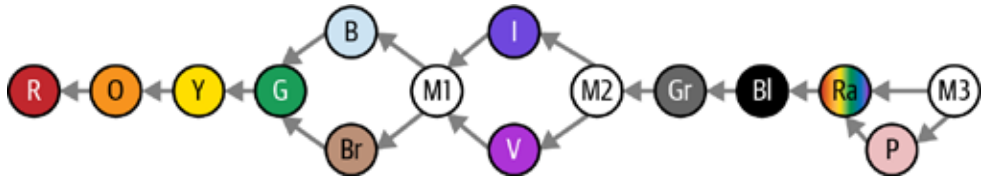
SLIKA P-1
Primer komita sa njegovim punim imenom

Tabela P-1 sadrži listu svih komita koje ćete napraviti u projektu Rainbow u ovoj knjizi, prikazujući njihova puna imena i njihove skraćenice.

Tabela P-1. Puna lista komita napravljenih u projektu Rainbow

KOMIT PUNO IME	KOMIT SKRAĆENICA
red (crvena)	R
orange (narandžasta)	O
yellow (žuta)	Y
green (zelena)	G
blue (plava)	B
brown (braon)	Br
merge commit 1 (spajanje komita 1)	M1
indigo (indigo)	I
violet (ljubičasta)	V
merge commit 2 (spajanje komita 2)	M2
gray (siva)	Gr
rainbow (duga)	Bl
rainbow	Ra
pink (ružičasta)	P
merge commit 3 (spajanje komita 3)	M3

Slika P-2 je dijagram svih komita.



SLIKA P-2
Petnaest komita koje ćete napraviti u projektu Rainbow do kraja ove knjige

DODACI

Iako je knjiga dizajnirana da se čita od Poglavlja 1 do Poglavlja 12 linearno, može biti situacija u kojima želite ili morate početi od specifičnog poglavlja. Na primer:

- Prošli ste kroz vežbe u celoj knjizi jednom i želite ponovo pregledati od određenog poglavlja nadalje.
- Nešto je pošlo po zlu u projektu Rainbow u prethodnom poglavlju i niste mogli naći rešenje i želite da nastavite sa novog poglavlja na svež start.

U tom slučaju, možete koristiti uputstva u Dodatku A za kreiranje kako projekt Rainbow treba izgledati na početku poglavlja odakle želite početi.

Dodatak B sadrži brz vodič kroz komande predstavljene u svakom poglavlju.

Dodatak C je vodič kroz vizuelni jezik korišćen u dijagramima u knjizi.

PROJEKAT KNJIGE

Projekat knjige je zamišljen projekat koji ću koristiti za demonstraciju kako se Git može koristiti za realistične projekte. Za ovaj projekat, pretvaraćemo se da pišem knjigu i želim da koristim Git za kontrolu verzija datoteka. Knjiga će se sastojati od 10 poglavlja predstavljenih sa 10 tekstualnih datoteka, jednom za svako poglavlje: `chapter_one.txt`, `chapter_two.txt`, i tako dalje. Ponekad, simuliraću takođe kako bi bilo raditi na projektu Knjiga sa koautorom i/ili urednikom. Ove diskusije će se odvijati u sekcijama *Projekt knjige*.

Nećete aktivno raditi na Projektu knjige. Projekat će biti korišćen samo da se pruže primeri i opisi kako se određene funkcionalnosti Gita koriste.

Osim sekcija Projekta knjige, u knjizi ćete naći i na neke druge sekcije. Pogledaćemo ih u nastavku.

SEKCIJE U KNJIZI

Evo kratkog vodiča za različite tipove sekcija sa kojima ćete se susresti u ovoj knjizi:

Projekat knjige

Sekcije Projekat knjige pružaju dodatni kontekst i primere o korišćenju Git funkcionalnosti i komandi, na osnovu projekta Knjige.

Koraci

Sekcije Koraci predstavljaju numerisane liste koraka koje treba izvesti na računaru. Ako korak sadrži podebljanu komandu, onda morate uneti i izvršiti tu komandu u komandnoj liniji. Primer izlaza je dat za sve komande koje proizvode izlaz. Ovaj izlaz se zasniva na Rainbow projektu na kojem sam radila dok sam pisala ovu knjigu; generisan je od operativnog sistema macOS, ali izlaz Git komandi bi trebalo da bude isti na Microsoft Windowsu. Kada postoje značajne razlike između izlaza komande na Microsoft Windowsu i macOSu, to je navedeno u tekstu.

Zapamtite komandu

Sekcije Zapamtite komandu uvode korisne komande, neke od kojih ćete koristiti u sekcijama Koraci.

Puna lista svih važnih komandi grupisanih po poglavljima je dostupna kao referenca u Dodatku B.

Prikaz

Sekcije Prikaz prikazuju dijagrame šta se dešava u Rainbow projektu. Dva važna dijagrama koja se koriste u ovim sekcijama (kao i slike) su *Git Dijagram* i *Dijagram skladišta*. Upoznaću vas sa Git Dijagramom u Poglavlju 2 i počećemo da koristimo Dijagram Skladišta u Poglavlju 4. Svaki dijagram u knjizi je izgrađen korak po korak, sa objašnjenjima u tekstu.

Rezime vizuelnog jezika korišćenog kroz celu knjigu je dostupno kao referenca u Dodatku C.

Napomena

Sekcije Napomena pružaju korisne informacije vezane za materijal u tekstu.

UČENJE GIT SKLADIŠTA

Iako je moj cilj da pružim većinu informacija koje su vam potrebne kako biste prošli *Naučite Git* iskustvo unutar same knjige, postoje neke tehnologije i procesi koji se previše često menjaju da bi se dokumentovali u knjizi. Kreirala sam javno Learning Git skladište (*https://github.com/gitlearningjourney/learning-git*) na GitHub-u da pružim ažurirane informacije o ovim tehnologijama i procesima. Skladište, između ostalog, sadrži:

- Informacije o preuzimanju Gita
- Linkove do resursa vezanih za rad sa hosting uslugama
- Informacije o podešavanju HTTPS ili SSH pristupa udaljenim skladištima

Kroz knjigu, pominjaću kada bi trebalo da se konsultujete sa Naučite Git skladištem za više uputstava.

ŠTA OVA KNJIGA NIJE

Ovo nije referentna knjiga. Neću pokrivati svaku pojedinačnu Git komandu (verujte mi, ima ih puno!). Takođe, ovo nije napredni vodič za Git. Postoje mnoge funkcionalnosti Gita koje neće biti izložene u ovoj knjizi zato što nisu neophodne da izvedete osnovne akcije kojima želim da vas naučim. Bila sam veoma selektivna u onome što sam uključila u knjigu. Moj cilj je da vam dam jasan mentalni model osnova Gita tako da možete da nastavite da učite o svim dodatnim funkcionalnostima koje su vam potrebne sa solidnom osnovom na mestu.

Ova knjiga neće reći kako treba da vaš Git radni tok izgleda ili kako treba koristiti funkcionalnosti Gita. Koliko god je moguće, trudim se da se držim dalje od davanja mišljenja o ovim temama i umesto toga se fokusiram na učenje kako alat radi. U zavisnosti od vašeg individualnog konteksta i preferencija, alat može biti korišćen na različite načine.

Gde je moguće, ova knjiga se trudi da ne bude preskriptivna. Na primer, dozvoljeno vam je da koristite bilo koji tekst editor ili hosting servis koji preferirate za izvođenje vežbi u knjizi.

REZIME POGLAVLJA

Knjiga je podeljena na dva logična dela. U prvom delu knjige (Poglavlja 1 do 5), naučićete o radu sa lokalnim skladištima na vašem računaru. U Poglavljima 6 do 12, dodatno ćete naučiti o radu sa hosting servisima i udaljenim skladištima. Evo kratkog rezimea šta ćemo pokriti u svakom poglavlju:

- U Poglavlju 1, pripremićete se za rad na projektu koristeći Git instaliranjem, učenjem osnova komandne linije, pripremanjem Git podešavanja, pravljenjem jednog od direktorijuma projekta koje ćete koristiti kroz ostatak knjige i pripremom tekst editora.
- U Poglavlju 2, pretvoričete vaš direktorijum projekta u Git skladište. Upoznaću vas sa Git Dijagramom koji predstavlja različite oblasti Gita, uključujući radni direktorijum, prostor pripreme (staging area) istoriju komita i lokalno skladište. Na kraju poglavlja ćete kreirati prvu datoteku u vašem direktorijumu projekta.
- U Poglavlju 3, naučićete i izvesti dva glavna koraka da napravite prvi komit u vašem lokalnom skladištu.

- U Poglavlju 4, naučićete o granama: šta su one, kako ih napraviti, kako se prebaciti na grane i kako identifikovati na kojoj se grani nalazite.
- U Poglavlju 5, naučićete o dva tipa spajanja i izvesti ćete spajanje nadovezivanjem (fast-forward merge).
- U Poglavlju 6, pripremićete se za rad sa udaljenim skladištima biranjem hosting servisa i postavljanjem autentifikacionih detalja za spajanje na ta skladišta preko ili HTTPS ili SSH.
- U Poglavlju 7, diskutovaćemo o različitim načinima rada sa lokalnim i udaljenim skladištima. Naučićete kako kreirati udaljeno skladište i poslati podatke na njega.
- U Poglavlju 8, počećemo simulirati šta je to kao raditi sa drugima na Git projektu. Kako biste to uradili, kreiraćete drugo lokalno skladište; pretvaraćete se da pripada saradniku koji će vam pomoći na Rainbow projektu i to je na njihovom računaru. U procesu, naučićete o kloniranju udaljenih skladišta i preuzimanju podataka.
- U Poglavlju 9, izvešćete trostruko spajanje (three-way merge) i naučićete o razlici između preuzimanja podataka i povlačenja podataka.
- U Poglavlju 10, pregledaćemo primer rešavanja sukoba spajanja tokom trosmernog spajanja (three-way merge).
- U Poglavlju 11, saznaćete o rebaziranju (rebasing). To je alternativan način uključivanja promena iz jedne grane u drugu, suprotno od spajanja.
- U Poglavlju 12, naučićete o zahtevima za povlačenje (poznatim kao zahtevi za spajanje) i kako oni olakšavaju saradnju u Git projektima.
- Knjiga uključuje tri dodatka, čiji se sadržaji nalaze u „Dodaci“ na strani xv.

Konvencije korišćene u ovoj knjizi

Sledeće tipografske konvencije su korišćene u ovoj knjizi:

Kurziv
: Označava nove termine, URL-ove, imejl adrese i naglašavanje.

`Konstantna širina`
: Koristi se za komande, izlaz komandi, nazive datoteka, nazive direktorijuma, nazive grana, nazive skladišta i drugi tekst na ekranu.

`Konstantna širina podebljano`
: Prikazuje komande ili drugi tekst koji bi trebalo da se kuca ili unosi od strane korisnika.

`<Konstantna širina uglaste zagrade>`

: Prikazuje tekst koji bi trebalo da se zameni sa vrednostima koje korisnik obezbedi ili sa vrednostima koje se određuju kontekstom.

Kako da nas kontaktirate

Komentare i pitanja u vezi ove knjige uputite izdavaču:

MIKRO KNJIGA, DOO
Kneza Višeslava 34
11030 Beograd, Srbija
+381 11 7702-883
redakcija@mikroknjiga.rs

Ispravke, primeri i dodatne informacije originala knjige nalaze se na stranici *https://oreil.ly/learning-git*. Posetite *http://oreilly.com* za vesti i informacije o našim knjigama i kursevima.

Zahvalnosti

Želela bih da se zahvalim svim ljudima koji su me podržali kroz proces pisanja ove knjige. To uključuje prijatelje i porodicu, neke od kojih su bili prisiljeni da testiraju ranije verzije knjige. To takođe uključuje ceo O'Reilly tim koji me je podržavao kroz ovaj kreativni projekat i konačno sve tehničke recenzente i korisnike koju koji su pročitali različite iteracije knjige i pružili mi neprocenjive povratne informacije. Oh, i mog partnera, koji je bio strpljiv dok sam više puta odlagala datum objavljivanja!

[1]

Git i komandna linija

U ovom poglavlju ću vam predstaviti šta je Git i zašto ga koristimo, i proverićete da li ga imate instaliranog na svom računaru. Takođe ćete postaviti neke Git konfiguracione promenljive (tzv. *postavke*). Predstaviću grafički korisnički interfejs i komandnu liniju, koji su dva alata koja ćete koristiti za interakciju sa Gitom i praktičnim projektom Rainbow. Kako biste se osećali ugodno radeći na komandnoj liniji, proći ćemo kroz osnovne radnje kao što su pregled tekuće lokacije direktorijuma, navigacija unutar i izvan direktorijuma, i kreiranje direktorijuma. Na kraju poglavlja, pripremićete tekst editor koji ćete koristiti za rad na projektu Rainbow u Poglavlju 1.

Ako već imate neko iskustvo radeći na komandnoj liniji, možda već znate neke informacije iz ovog poglavlja. Međutim, ne preporučujem preskakanje ovog poglavlja jer priprema postavke koje ćete koristiti tokom ostatka knjige.

[NAPOMENA]

Da biste razumeli kako koristiti ovu knjigu, morate pročitati „Korišćenje ove knjige“ na strani BROJ_STRANE. Ako niste pročitali tu sekciju, preporučujem da se vratite i uradite to sada.

Šta je Git?

Git je tehnologija koja se koristi za praćenje promena u projektu i kao pomoć za više ljudi koji srađuju na istom projektu. Na osnovnom nivou, verzija projekta kontrolisana Gitom sastoji se od foldera sa datotekama, a Git prati promene koje su napravljene u datotekama projekta. To vam omogućava da sačuvate različite verzije posla koji obavljate, zbog čega nazivamo Git *sistemom za kontrolu verzija*.

Git je kreirao Linus Torvalds za kontrolu verzija rada na velikom projektu razvoja softvera poznatom kao Linux kernel. Međutim, budući da Git može pratiti promene u različitim vrstama datoteka, može se koristiti za različite vrste projekata.

Git je moćna tehnologija, a obilje funkcija koje pruža – kao i činjenica da je prvobitno dizajniran za korišćenje na komandnoj liniji – znači da je njegovo korišćenje malo složenije od jednostavnog izbora File→Save na vašem računaru.

Da sumiram, Git je sistem za kontrolu verzija koji možete preuzeti na svoj računar i omogućava vam praćenje istorije projekta i saradnju sa drugim ljudima. Pogledajmo Projekat knjige 1-1 da vidimo kako bih mogao koristiti Git za svoj projekat knjige.

Projekat knjige 1-1

Pretpostavimo da pišem knjigu i da želim koristiti Git za kontrolu verzija svih datoteka u svom projektu knjige. Svaki put kad napravim promene u knjizi, mogu sačuvati verziju koristeći Git. Na primer, pretpostavimo da napravim promene u knjizi ponedeljkom, sredom i petkom, i da sačuvam jednu verziju svaki od tih dana. To znači da imam barem tri verzije svog projekta. Verzija projekta u Gitu naziva se *komit* (commit), upis promena. U Poglavlju 1, saznaćete više o komitima. Za sada, sve što treba da znate je da u mom primeru imam barem tri komita.

Ta tri komita omogućavaju mi da pogledam različite verzije knjige koje sam imao na kraju svog radnog dana ponedeljkom, sredom i petkom. Git takođe omogućava upoređivanje bilo kojih od tih komita (ili sačuvanih verzija mog projekta) međusobno kako biste proverili šta se promenilo između različitih verzija. Ovo ilustruje kako mi Git pomaže da pratim istoriju svog projekta.

Sada, pretpostavimo da odlučim da radim na projektu knjige zajedno sa koautorom. Git mi omogućava da moj koautor i ja radimo na istom projektu istovremeno i kombinujemo naš rad kad budemo spremni. Na primer, ja mogu raditi na Poglavlju 1, a moj koautor može raditi na Poglavlju 2, i kada budemo spremni, možemo kombinovati rad koji smo uradili.

Ako imamo urednika da pregleda knjigu, on takođe može napraviti izmene na svim poglavljima knjige koje smo napisali, i možemo integrisati te promene u glavnu verziju knjige. Ovo ilustruje kako je Git korisno sredstvo za saradnju.

Sada ćemo saznati nešto više o drugim alatima koje ćete koristiti u ovom iskustvu učenja i kako ćete biti u interaiciji sa Gitom.

Grafički korisnički interfejs i komandna linija

Dva glavna načina za interakciju sa računarom su korišćenje grafičkog korisničkog interfejsa ili komandne linije.

Grafički korisnički interfejs (graphical user interface, GUI) predstavlja skup grafičkih objekata (ikonice, dugmadi itd.) koji vam omogućava da budete u interakciji sa računarom. Možete ga zamisliti kao interfajs sa pokazivačem i klikovima. Na primer, folder predstavljen ikonicama fascikli na radnoj površini računara deo su GUI-ja vašeg računara.

Komandna linija, takođe poznata kao interfajs komandne linije (CLI, command line interface), terminal ili shell, je mesto gde možete unositi tekstualne komande kako biste komunicirali sa računarom.

Podrazumevani način rada sa Gitom je preko komandne linije. Međutim, postoje i načini na koje možete raditi s Gitom putem GUI-ja: na primer, korišćenjem klijenta za Git sa grafičkim korisničkim interfejsom ili tekst editora koji ima integraciju s Gitom. To znači da možete izvršavati Git akcije pritiskom na dugme i izborom opcija umesto unošenja komandi na komandnoj liniji.

U ovoj knjizi saznaćete kako da koristite Git na komandnoj liniji, jer vam to omogućava da izgradite čvrst mentalni model kako radi i daje vam pristup svim svojim funkcionalnostima. GUI računara ćete koristiti samo za druge akcije, na primer da pregledate datoteke u fajl sistemu ili da radite u svom tekst editoru za upravljanje datotekama. Detaljnije ćemo istražiti komandnu liniju u sledećem odeljku.

[NAPOMENA]

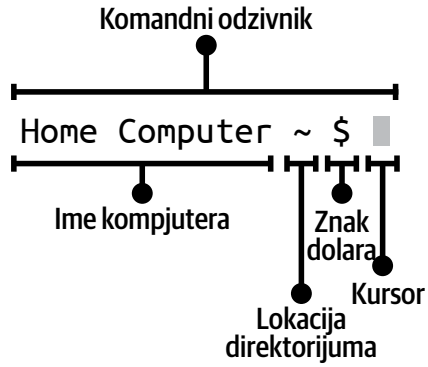
U ovoj knjizi pružiću neke specifične instrukcije za korisnike macOS-a i Microsoft Windowsa. Ako ste korisnik Linuxa, pretpostaviću da već poznajete osnove komandne linije.

Otvaranje prozora komandne linije

Da biste koristili komandnu liniju, morate otvoriti prozor komandne linije pomoću aplikacije za komandnu liniju. U bilo kom trenutku, u prozoru komandne linije, nalazite se u određenom direktorijumu, koji nazivamo *tekući direktorijum*. Direktorijum, za naše potrebe, isto je što i fascikla.

Kada otvorite prozor komandne linije, u gornjem levom uglu nalazi se komandni *odzivnik* (prompt), kratak tekst čiji će tačan sadržaj zavisiti od vašeg operativnog sistema i postavki računara. Međutim, po difoltu, komandni odzvinik označava lokaciju direktorijuma na komandnoj liniji (vaš tekući direktorijum). Kada otvorite novi prozor komandne linije, lokacija direktorijuma počinje od tekućeg korisničkog direktorijuma (matični folder), što je predstavljeno znakom tilde (~). Ova lokacija direktorijuma biće jedini važan deo komandnog odzivnika koji ćete morati prepoznati za vežbe u ovoj knjizi. Nakon odzivnika, nalazi se kursor koji označava gde unosite komandu.

Pogledajte Sliku 1-1 za primer komandnog odzivnika. U primerima u ovoj knjizi koristimo znak dolara (`$`) na kraju komandnog odzivnika, ali to je samo jedan primer kako se komandni odzivnik može završiti. Vaš komandni odzivnik može završiti drugim znakom ili simbolom.



SLIKA 1-1:
`Primer komandnog odzivnika`

Aplikacija za komandnu liniju koju ćete koristiti za obavljanje vežbi u ovoj knjizi zavisiće od operativnog sistema koji koristite:

- macOS – aplikacija za komandnu liniju se zove Terminal.
- Microsoft Windows – aplikacija za komandnu liniju se zove Git Bash. Dostupna je samo ako imate instaliran Git na svom računaru.

[NAPOMENA]

Ako koristite Microsoft Windows i nemate instaliran Git, trebaćete da posetite Git Learning repozitorijum (*https://github.com/gitlearningjourney/learning-git*) i pratite korake za instalaciju Gita za Microsoft Windows kako biste imali pristup Git Bashu pre nego što nastavite sa ostatkom ovog poglavlja.

Ako koristite macOS i nemate instaliran Git, možete nastaviti sa ovim odeljkom. Git ćete instalirati po uputstvu pod naslovom „Instalacija Gita“.

Da biste otvorili prozor komandne linije, možete koristiti funkciju pretrage na svom računaru kako biste pronašli aplikaciju za komandnu liniju, izabrali je i otvorili. Pogledajte sekciju Koraci 1-1 da biste otvorili prozor komandne linije i videli komandni odzivnik.

[KORACI 1-1]

1. Koristite svoju aplikaciju komandne linije da otvorite prozor komandne linije.
2. Pogledajte komandni odzivnik u prozoru komandne linije.

Obratite pažnju:

- Komandni odzivnik pokazuje lokaciju direktorijuma.

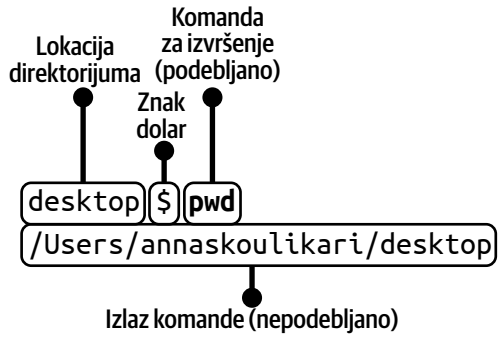
Sada kada ste otvorili prozor komandne linije, hajde da vidimo kako ćete izvršiti svoju prvu komandu na komandnoj liniji.

Izvršavanje komandi na komandnoj liniji

Na kraju komandnog odzivnika u vašem komandnom prozoru je *kursor* koji pokazuje gde ćete unositi komande. U Terminalu (macOS), po difoltu, kursor je na istom redu kao komandni odzivnik, dok je u Git Bashu (Microsoft Windows) na redu ispod. Nakon što unesete komandu, da biste je izvršili, morate pritisnuti taster Enter (Return).

Ako korak u sekciji Koraci u ovoj knjizi sadrži komandu podebljanu nakon znaka dolara (**$**), onda je morate izvršiti na komandnoj liniji. Ako komanda proizvodi izlaz, biće prikazana ispod komande (nepodebljano). Ako se komanda treba izvršiti u direktorijumu koji nije tekući korisnički direktorijum, lokacija direktorijuma će biti naznačena pre simbola za dolar.

Slika 1-2 prikazuje primer kako će izgledati kada izvršite komandu u sekciji Koraci. U ovom slučaju, izvršavam komandu `pwd` dok sam u direktorijumu `desktop`. Kasnije ćete saznati šta komanda `pwd` radi u ovom poglavlju; za sada, pogledajte Sliku 1-2 da biste identifikovali gde se prikazuje lokacija direktorijuma, komanda za izvršavanje i izlaz.



SLIKA 1-2:
Kako izvršiti komandu koja je uključena u sekciji Koraci

U štampanoj verziji ove knjige, neke duže komande se prelamaju zbog ograničenja širine stranice. Ako vidite komandnu liniju kao što je sledeća:

```
rainbow $ git remote add origin https://github.com/gitlearningjour
ney/rainbow-remote.git
```

treba da unesete celu komandu u jednom redu. Podsetnici će biti postavljeni na odgovarajućim mestima. Ako čitate verziju ove knjige koja nema ovu ograničenost dužine linije, možete zanemariti ove podsetnike.

IZLAZ KOMANDE

Neke komande proizvode izlaz, a neke ne. Za komande koje proizvode izlaz, pružiću uzorak izlaza koji se bazira na projektu Rainbow na kojem sam radio prilikom pisanja ove knjige. Ovaj izlaz je generisan od strane macOS operativnog sistema, ali izlaz za Git komande se ne razlikuje između operativnih sistema. U nekoliko slučajeva gde izlaz komande koja nije povezana sa Gitom značajno varira između operativnih sistema, to će biti naznačeno u tekstu.

Ako, dok obavljate vežbe iz sekcije Koraci, vidite izlaz koji se značajno razlikuje od izlaza u ovoj knjizi ili dobijete neočekivanu grešku, tada ste verovatno nešto drugačije uradili od onoga što uputstva ukazuju i možda ćete morati da ponovite korake.

[NAPOMENA]

Moguće je da će Git proći kroz ažuriranja u budućnosti koja mogu malo uticati na izlaz. Ako postanem svestan značajnih promena, trudiću se da dokumentujem te situacije na stranici sa ispravkama u Learning Git repository (*https://github.com/gitlearningjourney/learning-git*).

Od sada, svaki put kada naiđete na komandu u vežbama u sekciji Koraci, treba da unesete podebljano komandu na komandnoj liniji i izvršite je.

IZVRŠAVANJE PRVE KOMANDE NA KOMANDNOJ LINIJI

Prva komanda kojom ćete vežbati izvršavanje na komandnoj liniji je `git version` komanda. Ako imate Git instaliran na svom računaru, tada će vam dati verziju koju ste instalirali. Ako nemate Git instaliran na svom računaru, izlaz će vam dati poruku koja ukazuje da nije instaliran.

Da biste imali pristup svim komandama korišćenim u vežbama u ovoj knjizi, preporučujem da imate verziju Gita veću od 2.28. Idite na Korake 1-2 da proverite da li imate Git instaliran i koja verzija je u pitanju.

[KORACI 1-2]

```
1 $ git version
  git verzija 2.35.1
```

Obratite pažnju:

- Ako imate Git instaliran, videćete verziju koja je instalirana na vašem računaru.
- Ako nemate Git instaliran, umesto toga ćete videti poruku o grešci.

Ako izlaz komande `git version` ukazuje da imate instaliranu verziju Gita koja je veća od 2.28, možete preskočiti sekciju „Instalacija Gita“ i preći na sekciju „Opcije i argumenti komandi“.

Ako izlaz komande `git version` ukazuje da nemate Git instaliran na svom računaru ili da je verzija Gita starija od 2.28, treba da nastavite sa sledećom sekcijom kako biste instalirali ažuriranu verziju Gita na svom računaru.

Instalacija Gita

Ako nemate Git (verzija 2.28 ili novija) instaliran, idite na Korake 1-3. U suprotnom, pređite na sledeću sekciju.

[KORACI 1-3]

1. Posetite Learning Git repository (*https://github.com/gitlearningjourney/learning-git*) i pratite korake za preuzimanje Gita za svoj operativni sistem.

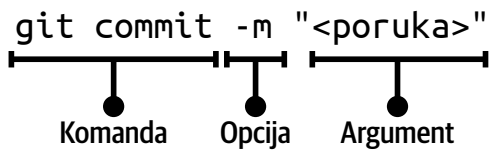
Sada kada ste instalirali ažuriranu verziju Gita na svom računaru, nastavite sa učenjem o komandama koje ćete koristiti u sekcijama Koraci u ovoj knjizi.

Opcije i argumenti komandi

Ponekad ćete koristiti komande sa opcijama i/ili argumentima. *Opcije* su postavke koje menjaju ponašanje komande. Opcija se obeležava jednom crticom (–) ili dvostrukom crticom (––).

Argumenti su vrednosti koje daju informacije komandi. Označavaće se uglastim zagradama (<>), što ukazuje da ove stavke treba zameniti vrednostima koje korisnik unosi. U vežbama, moraćete uneti vrednost za argument, bez uključivanja uglastih zagrada.

Primer komande sa opcijom i argumentom koju ćete koristiti je `git komit -m "<poruka>"`. U ovom primeru, `-m` je opcija, a `<poruka>` je argument, kako je prikazano na slici 1-3. Objasnićemo šta ova komanda radi u Poglavlju 3.



SLIKA 1-3
Primer komande sa opcijom i argumentom

Osim unošenja komandi na komandnoj liniji, takođe je važno naučiti kako da ih očistite.

Čišćenje komandne linije

Svaki put kada unesete komandu u prozoru komandne linije, ona će biti navedena neposredno ispod prethodne komande koju ste uneli (ili njenog izlaza). Kada unesete mnogo komandi, prozor komandne linije postaje prilično zagušen. Da biste očistili sadržaj prozora komandne linije, možete koristiti komandu `clear`.

[ZAPAMTITE KOMANDU]

clear
Briše prozor komandne linije

Idite na Korake 1-4 da vežbate upotrebu komande `clear`.

[KORACI 1-4]

```
1 $ clear
```

Obratite pažnju:

- Prozor komandne linije je prazan.

Objasnili smo unošenje komandi i brisanje komandi iz prozora komandne linije. Sada je vreme da pripremite sledeći alat koji će vam pomoći u vašem učenju, prozor sistema datoteka.

Otvaranje prozora sistema datoteka

Koristićete aplikaciju sistema datoteka da otvorite prozor sistema datoteka, koji je deo grafičkog korisničkog interfejsa. Tokom vašeg učenja, komuniciraćete sa prozorom sistema datoteka i prozorom komandne linije. Stoga je korisno da oba prozora budu otvorena jedan pored drugog na ekranu vašeg računara.

Aplikacija sistema datoteka koju ćete koristiti zavisiće od vašeg operativnog sistema:

- macOS – aplikacija sistema datoteka je Finder.
- Microsoft Windows – aplikacija sistema datoteka je File Explorer (ili Windows Explorer).

Idite na Korake 1-5 da otvorite prozor sistema datoteka.

[KORACI 1-5]

1. Pronađite svoju aplikaciju sistema datoteka i otvorite prozor sistema datoteka pored prozora komandne linije.

Sada kada imate oba prozora otvorena, vratimo se osnovama komandne linije.

Rad sa direktorijumima

U bilo kom trenutku u prozoru komandne linije nalazite se u određenom direktorijumu (tekući direktorijum). Pretpostavimo da niste promenili difolt postavke, kada prvi put otvorite aplikaciju komandne linije, početni direktorijum će biti tekući korisnički direktorijum (vaš matični folder), što je naznačeno znakom tilde (~) u komandnom odzivniku.

Kada idete do drugih direktorijuma na komandnoj liniji, odzvinik će se promeniti kako bi pokazao direktorijum u kojem se nalazite. Možete koristiti komandu `pwd` (print working directory), što znači „ispiši radni direktorijum," da biste videli putanju do tekućeg direktorijuma.

[ZAPAMTITE KOMANDU]

pwd

Prikazuje putanju do tekućeg direktorijuma

Idite na Korake 1-6 da vežbate upotrebu komande `pwd`.

[KORACI 1-6]

```
1 $ pwd
  /Korisnici/annaskoulikari
```

Obratite pažnju:

- Nalazite se u svom tekućem korisničkom direktorijumu.

[NAPOMENA]

U Koracima 1-6, izlaz komande pwd za korisnike Microsoft Windowsa će biti sličan kao `/c/Users/annaskoulikari`, što je malo drugačije od izlaza za korisnike macOS-a. Imajte to na umu dok prolazite kroz preostale vežbe u ovom poglavlju.

Izlaz komande `pwd` ispisuje putanju do tekućeg direktorijuma. U nastavku 1-6, `/Users/annaskoulikari` je primer putanje. Moje ime je Anna Skoulikari i `annaskoulikari` je moje korisničko ime na mom računaru. `Users` i `annaskoulikari` su dva direktorijuma. Direktorijumi u putanji su odvojeni kosom crtom (`/`). Direktorijum `annaskoulikari` se nalazi unutar direktorijuma `Users`.

Poznavanje lokacije direktorijuma u prozoru komandne linije je korisno jer mnoge komande prikazuju informacije ili utiču na tekući direktorijum kada ih izvršite. Takođe vam pomaže pri navigaciji kroz vaš fajl sistemski, što ćemo kasnije obraditi u ovom poglavlju.

Sada kada smo objasnili kako identifikovati tekući direktorijum, hajde da istražimo kako videti stvarne sadržaje direktorijuma.

PREGLED SADRŽAJA DIREKTORIJUMA

Možete pregledati sadržaj direktorijuma u grafičkom interfejsu i prozoru komandne linije. Ali pre nego što to uradimo, želim da napomenem da postoje dva tipa datoteka i direktorijuma koji postoje u fajl sistemu: vidljive datoteke i direktorijumi i skrivene datoteke i direktorijumi. *Vidljive* datoteke i direktorijumi su uvek vidljivi u fajl sistemu. *Skrivene* datoteke i direktorijumi su vidljivi samo u fajl sistemu ako promenite svoje postavke da biste ih videli. Često su to datoteke ili direktorijumi koji sadrže informacije kojim mi kao korisnici ne moramo da pristupimo, kao što su postavke aplikacija i razne sistemske postavke.

Ne preporučujem da menjate ili brišete skrivene datoteke ili direktorijume, osim ako zaista znate šta radite. Kada promenite postavke da biste videli skrivene datoteke i direktorijume, one postaju zasivljene (posvetljene). Njihova imena često počinju tačkom (`.`).

Postoje neki važni skriveni fajlovi i direktorijumi o kojima ćete želeti da budete obavešteni u svom putovanju u učenju Gita, pa morate znati kako da ih vidite, kako u grafičkom interfejsu, tako i komandnoj liniji.

U grafičkom interfejsu, da biste videli skrivene datoteke i direktorijume u prozoru fajl sistema, morate ih eksplicitno učiniti vidljivim:

- macOS – za prebacivanje između prikazivanja i skrivanja skrivenih datoteka i direktorijuma, pritisnite Cmd-Shift-tačka.
- Microsoft Windows – morate promeniti postavke fajl sistema da biste videli skrivene datoteke i direktorijume. Po potrebi se konsultujte sa online resursima za korak-po-korak instrukcije za vaš računar.

Idite na Korake 1-7 da biste videli skrivene datoteke i direktorijume na vašem fajl sistemu.

[KORACI 1-7]

1 Učinite da skrivene datoteke i direktorijumi budu vidljivi na vašem fajl sistemu.

U komandnoj liniji, da biste videli listu vidljivih datoteka i direktorijuma u tekućem direktorijumu, koristite komandu `ls` (što znači „list“).

Da biste videli kako i skrivene datoteke i direktorijume u tekućem direktorijumu, koristite komandu `ls` sa opcijom `-a`: `ls -a`.

[ZAPAMTITE KOMANDU]

ls
Listanje vidljivih datoteka i direktorijuma

ls -a
Listanje skrivenih i vidljivih datoteka i direktorijuma

Idite na Korake 1-8 da biste vežbali korišćenje ovih komandi za listanje različitih vrsta datoteka.

[KORACI 1-8]

```
1 $ ls
  Applications          Downloads             Music
  Desktop               Library               Pictures
  Documents             Movies                Public

2 $ ls -a
  .                     .config               Desktop
  ..                    .local                Documents
  ..CFUserTextEncoding  .npm                  Downloads
  .DS_Store             .ssh                  Library
  .Trash                .viminfo              Movies
  .bash_history         .vscode               Music
  .bash_sessions        .yarnrc               Pictures
  .bashrc               Applications          Public
```

Obratite pažnju:

- Nazivi mnogih skrivenih datoteka i direktorijuma počinju tačkom (`.`).
- Vidljive i skrivene datoteke i direktorijumi prikazani u izlazu u ovoj knjizi razlikovaće se od onih na vašem računaru jer su sadržaji računara različiti.

Sada kada smo objasnili kako da identifikujemo tekući direktorijum i vidimo njegov sadržaj na komandnoj liniji, istražićemo kako da se krećemo između direktorijuma.

NAVIGACIJA UNUTAR I VAN DIREKTORIJUMA

U grafičkom korisničkom interfejsu (GUI), da biste ušli u direktorijum možete ga dvaput kliknuti. Na komandnoj liniji, da biste ušli u direktorijum koristite komandu `cd`, koja označava „promeni direktorijum“ (change directory), i prosledite ili naziv direktorijuma ili putanju do direktorijuma.

[ZAPAMTITE KOMANDU]

cd <putanja_do_direktorijuma>
Menja direktorijum

Idite na Korake 1-9 da biste vežbali navigaciju u direktorijumu `desktop`.

[KORACI 1-9]

```
1 $ cd desktop
2 desktop $ pwd
  /Users/annaskoulikari/desktop
```

Obratite pažnju:

- U koraku 1, komanda `cd` ne proizvodi nikakav izlaz.
- U koraku 2, komandni odzivnik i izlaz komande `pwd` pokazuju da se nalazite u direktorijumu `desktop`.

Ranije u ovom poglavlju sam spomenula da komandni odzivnik prikazuje lokaciju direktorijuma. U Koracima 1-9, uočite da se komandni odzivnik ažurira i pokazuje da je tekući direktorijum `desktop`. Po podrazumevanim postavkama, način prikaza lokacije direktorijuma razlikuje se zavisno o operativnom sistemu koji koristite:

- macOS – u Terminalu se prikazuje ime tekućeg direktorijuma.
- Microsoft Windows – u Git Bashu se prikazuje putanja do tekućeg direktorijuma.

[NAPOMENA]

Navigacija unutar i van direktorijuma na komandnoj liniji ne utiče na ono što vidite u fajl sistemu. Na primer, navigacija u direktorijum `desktop` u naredbenom redu neće automatski izazvati prikazivanje sadržaja direktorijuma `desktop` u vašem fajl sistemu.

U grafičkom interfejsu, da biste se vratili u nadređeni direktorijum, možete odabrati dugme za povratak. U komandnoj liniji, da biste se vratili u nadređeni direktorijum, možete koristiti dve tačke (`..`) uz `cd` komandu. Dve tačke predstavljaju nadređeni direktorijum tekućeg direktorijuma. Idite na Korake 1-10 da isprobate ovo.

[KORACI 1-10]

```
1 desktop $ cd ..
2 $ pwd
  /Users/annaskoulikari
```

Obratite pažnju:

- U koraku 2, odzivnik i izlaz komande `pwd` pokazuju da ste se vratili u direktorijum tekućeg korisnika.

Sada kada smo objasnili kako da se krećete unutar i izvan direktorijuma, hajde da pređemo na to kako da kreirate novi direktorijum.

KREIRANJE DIREKTORIJUMA

U grafičkom interfejsu možete kreirati direktorijum desnim klikom ili odabirom odgovarajuće opcije u meniju ili ikoni. U komandnoj liniji, za kreiranje direktorijuma ćete koristiti komandu nazvanu `mkdir`, što znači „make directory". Direktorijum će biti kreiran unutar tekućeg direktorijuma kada izvršite komandu.

[ZAPAMTITE KOMANDU]

mkdir <ime_direktorijuma>
Kreira direktorijum

Da biste olakšali stvari, izbegavajte uključivanje razmaka u imenima direktorijuma. Ako ime direktorijuma sadrži razmake, tada morate napraviti modifikacije u načinu kako koristite određene komande u komandnoj liniji, što čini vaše zadatke složenijim.

[NAPOMENA]

Uopšteno, prilikom rada u komandnoj liniji treba izbegavati korišćenje razmaka u imenima datoteka, direktorijuma ili drugih stvari koje stvarate, jer to može izazvati komplikacije prilikom korišćenja naredbi.

Kao što je navedeno u predgovoru, kroz ovu knjigu radićete na jednom projektu u kojem ćete kreirati i uređivati datoteke koje će nabrajati boje duge, kao i neke boje koje duga ne sadrži. Ovo nije realan primer projekta koji se obično verzionira pomoću Gita; to je pojednostavljen primer koji će vam omogućiti da se fokusirate na učenje kako Git radi.

Pošto je glavni cilj uzorka projekta da nabroji boje duge, vašem projektu ćete dati ime direktorijum `rainbow` (duga). Ovaj projekt direktorijum ćete kreirati u `desktop` direktorijumu kako biste ga lako videli na radnoj površini vašeg računara. Idite na Korake 1-11 da biste kreirali svoj direktorijum.

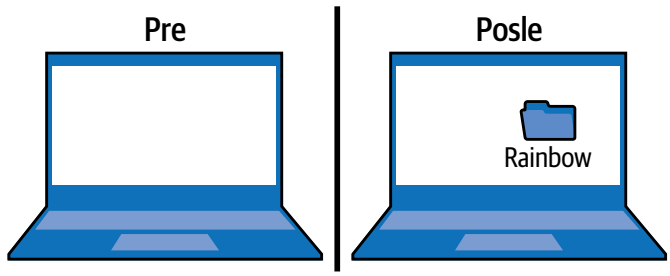
[KORACI 1-11]

```
1 $ cd desktop
2 desktop $ mkdir rainbow
3 desktop $ ls
  rainbow
```

Obratite pažnju:

- U koraku 3, izlaz naredbe `ls` će prikazati direktorijum projekta `rainbow` koji ste upravo kreirali, zajedno sa ostalim direktorijumima ili datotekama koje imate u svom `desktop` direktorijumu. Ovo nije prikazano u primeru izlaza, jer će sadržaj na vašem računaru biti drugačiji od mog.

Ako pogledate na radnu površinu vašeg računara, trebalo bi da vidite direktorijum projekta `rainbow` koji ste upravo kreirali, kao što je prikazano na slici 1-4.



SLIKA 1-4
Primer vaše radne površine pre i posle kreiranja direktorijuma projekta `rainbow`

Zatim ćete uočiti da samo zato što ste kreirali direktorijum projekta `rainbow`, to ne znači da ste se prebacili u njega u naredbenom redu. Idite na Korake 1-12 da biste se eksplicitno prebacili u direktorijum projekta `rainbow` u naredbenom redu.

[KORACI 1-12]

```
1 desktop $ cd rainbow
2 rainbow $ pwd
  /Users/annaskoulikari/desktop/rainbow
```

Obratite pažnju:

- U koraku 2, odzivnik i izlaz iz `pwd` komande pokazuju da se nalazite u direktorijumu `rainbow`.

Kreirali ste i otišli u direktorijum `rainbow`. Sada, šta se dešava ako zatvorite prozor komandne linije i ponovo ga otvorite?

Zatvaranje komandne linije

Podrazumevano, ako zatvorite prozor komandne linije i ponovo ga otvorite, lokacija direktorijuma će se resetovati na tekući korisnički direktorijum. Zbog toga ćete morati ponovo da odete do direktorijuma u kojem želite da radite. Idite na Korake 1-13 da biste ovo testirali.

[KORACI 1-13]

```
1 rainbow $ pwd
  /Users/annaskoulikari/desktop/rainbow
2 Zatvorite prozor komandne linije, a zatim otvorite novi prozor komandne linije.
3 $ pwd
  /Users/annaskoulikari
4 $ cd desktop
5 desktop $ cd rainbow
6 rainbow $ pwd
  /Users/annaskoulikari/desktop/rainbow
```

U ovom trenutku, pokrili smo nekoliko osnovnih naredbi komandne linije. Sledi postavljanje nekih osnovnih Git konfiguracija.

Postavljanje Git konfiguracija

Git konfiguracije su podešavanja koja vam omogućavaju da prilagodite rad Gita. Sastoje se od promenljivih i njihovih vrednosti i čuvaju se u nekoliko različitih fajlova. Da biste radili sa Gitom, morate postaviti nekoliko konfiguracionih promenljivih koje se odnose na korisnička podešavanja.

Pre postavljanja bilo kakvih promenljivih, proverite da li globalni Git konfiguracioni fajl postoji u vašem fajl sistemu i, ako postoji, koje su promenljive postavljene. To ćete uraditi koristeći `git config` komandu, prosleđujući opcije `--global` i `--list`.

Obratite pažnju da `git config` komanda sa opcijom `--global` predstavlja primer komande gde nije bitno u kom se tekućem direktorijumu nalazite prilikom izvršavanja; ona će uvek prikazati informacije o ili menjati informacije u globalnom Git konfiguracionom fajlu. Ovaj fajl je skriven i naziva se `.gitconfig` koji je obično kreiran u tekućem korisničkom direktorijumu.

[ZAPAMTITE KOMANDU]

git config --global --list
Listanje promenljivih u globalnom Git konfiguracionom fajlu i njihove vrednosti

Idite na Korake 1-14 da isprobate ovo.

[KORACI 1-14]

```
1 $rainbow $ git config --global --list
  fatal: unable to read config file '/Users/annaskoulikari/.
  gitconfig': No such file or directory
```

Obratite pažnju:

- Ovde prikazan izlaz ukazuje na to šta ćete videti ako niste postavili nikakve promenljive u svom globalnom Git konfiguracionom fajlu. U tom slučaju, dobićete grešku koja ukazuje da fajl ne postoji. Ako ste postavili promenljive u svom globalnom Git konfiguracionom fajlu, izlaz će prikazati promenljive koje ste postavili i njihove vrednosti.

Dve promenljive koje su nam važne za ovu knjigu su `user.name` i `user.email`. Svaki put kada neko sačuva verziju projekta (ili, drugim rečima, napravi komit, tj. upis promene), Git će zabeležiti ime i imejl adresu osobe i povezati ih sa tom sačuvanom verzijom. Promenljive

`user.name` i `user.email` se koriste da postave ime i imejl adresu koja će biti sačuvana za komite koje pravite. To znači da možete videti ko je radio na čemu u Git projektu. Konfigurisanje ovih promenljivih je obavezno kako biste radili sa Gitom. Imajte na umu da svako ko može da vidi listu vaših promena u bilo kom projektu može videti i vašu imejl adresu, pa se pobrinite da koristite adresu za koju vam ne smeta da je drugi vide.

Da biste postavili ove promenljive u svom globalnom Git konfiguracionom fajlu, prosledite ih kao argumente komandi `git config`, unesite željene vrednosti unutar navodnika (ne zaboravite da ne treba da uključite uglaste zagrade oko vrednosti).

[ZAPAMTITE KOMANDU]

git config --global user.name "<ime>"
Postavlja vaše ime u globalnom Git konfiguracionom fajlu

git config --global user.email "<email>"
Postavlja vašu imejl adresu u globalnom Git konfiguracionom fajlu

Ako u izlazu Koraka 1-14 ove promenljive imaju vrednosti koje želite, možete preskočiti Korake 1-15. Ako ove promenljive ne postoje ili nemaju vrednosti koje želite, odlučite koje ime i imejl adresu želite povezati sa radom na projektu Rainbow, i idite na Korake 1-15. Obavezno zamenite vrednostima koje želite koristiti umesto mog korisničkog imena i imejl adrese.

[KORACI 1-15]

```
1 rainbow $ git config --global user.name "annaskoulikari"
2 rainbow $ git config --global user.email
  "gitlearningjourney@gmail.com"
3 rainbow $ git config --global --list
  user.name=annaskoulikari
  user.email=gitlearningjourney@gmail.com
```

Obratite pažnju:

- U koraku 3, u izlazu `git config --global --list`, promenljive `user.name` i `user.email` postavljene su na vrednosti koje ste uneli.

Sada kada ste instalirali Git i pripremili svoje korisničke postavke, poslednji alat koji vam je potreban za rad na Git projektu je tekst editor.

Priprema tekst editora

Git projekat se sastoji od datoteka i direktorijuma koji su pod kontrolom verzija. Git može pratiti verzije svih vrsta datoteka. U projektu Rainbow, koristićete tekst editor za rad sa jednostavnim tekstualnim datotekama (koje su tipa `.txt`).

Tekst editor je program koji omogućava korisniku da uređuje običan tekst. Za mnoge vežbe koje ćete raditi u ovoj knjizi, trebaće vam tekst editor. Tekst editor se razlikuje od programa za obradu teksta, koji se uglavnom koriste za uređivanje bogatog teksta. Primeri programa za obradu teksta su Microsoft Word i Google Docs; oni se ne mogu koristiti za upravljanje datotekama u Git projektu. *Bogat tekst* (rich text) je tekst sa pridruženim ili ugrađenim stilovima. Ako vidite da je tekst podebljan (bold) ili kurziviran (italic), onda je to bogat tekst.

Neki tekst editori olakšavaju rad na Git projektima više nego drugi. Ova knjiga je napisana na način koji vam omogućava da koristite bilo koji tekst editor koji volite. Ako već imate instaliran tekst editor koji ste koristili za rad na Git projektima, slobodno ga koristite za vežbe u ovoj knjizi. Ako niste sigurni koji tekst editor da koristite, visoko preporučujem Visual Studio Code (*https://code.visualstudio.com*): to je popularan tekst editor i onaj koji sam koristila pri radu na projektu Rainbow u ovoj knjizi. Idite na Korake 1-16 kako biste pripremili tekst editor.

[KORACI 1-16]

1. Odaberite svoj omiljeni tekst editor. Ako već nemate jedan instaliran, preuzmite tekst editor.
2. Otvorite `direktorijum` projekta rainbow u prozoru tekst editor.

Integrisani terminali

Neki napredni tekst editori (*integrisana razvojna okruženja* – integrated development environments, IDE), kao što je Visual Studio Code, uključuju verziju komandne linije unutar sebe, obično nazvanu *integrisan terminal*, u kom možete izvršavati komande koje biste obično koristili u svom prozoru komandne linije.

Kada koristite integrisan terminal, može biti lakše upravljati prostorom na ekranu, jer je terminal već deo prozora tekst editora. Međutim, to zavisi od ličnih preferenci. Ako vaš tekst editor ima integrisan terminal, možete odlučiti da ga koristite umesto zasebnog prozora komandne linije kako biste izvršili Git komande u preostalim vežbama u ovoj knjizi. I jedno i drugo će raditi odlično, pa je na vama da odlučite šta vam više odgovara.

U ostatku ove knjige, kad god želim da se odnosi na mesto gde treba da izvršite komande, nazvaću ga prozorom komandne linije; međutim, imajte na umu da ovo takođe uključuje integrisan terminal.

Sada, sa instaliranim Gitom i vašim tekst editorom spremnim za rad, spremni ste da započnete rad na projektu Rainbow!

Rezime

U ovom poglavlju ste naučili nešto malo o Gitu i videli da je koristan alat za praćenje istorije projekta i saradnju sa drugima. Pripremili ste se za rad na projektu koristeći Git tako što ste instalirali ažuriranu verziju na svom računaru i postavili neke osnovne Git konfiguracione promenljive.

Naučili ste i neke osnove komandne linije, kao što je pregled sadržaja direktorijuma, kako da se krećete unutar njih i izlazite iz njih, i kako da ih kreirate. Pritom ste kreirali direktorijum projekta koji ćete koristiti tokom ovog procesa učenja, nazvan `rainbow`.

Napokon, pripremili ste tekst editor tako da možete kreirati i uređivati datoteke za bilo koji Git projekat na kojem želite da radite.

Sada ste spremni da pređete na Poglavlje 2, gde ćete zapravo pretvoriti svoj direktorijum projekta `rainbow` u Git repozitorijum (skladište) i početi da učite o najvažnijim oblastima prilikom rada sa Gitom.