

Davlatshoh Baxtiyorov



git

**Git va GitHub haqida
kichik qo'llanma**

GIT ASOSLARI

Git nima?

Git - bu har qanday kompyuter fayllari to'plamidagi o'zgarishlarni kuzatuvchi taqsimlangan versiya boshqaruv tizimi bo'lib, odatda dasturiy ta'minotni ishlab chiqish jarayonida manba kodini hamkorlikda ishlab chiqayotgan dasturchilar o'rtasida ishni muvofiqlashtirish uchun foydalaniladi. Uning maqsadlari tezlik, ma'lumotlar yaxlitligi va taqsimlangan, chiziqli bo'lmagan ish oqimlarini qo'llab-quvvatlashni o'z ichiga oladi.

Loyihaga git qo'shish uchun - "`git init`" ishlatiladi

Loyihadagi o'zgarishlarni ko'rish uchun - "`git status`" ishlatiladi

Ko'rsatilgan faylni stagingga qo'shish imkoniyati - "`git add <filename>`" va ("`git add .`" orqali esa barcha fayllarni stagingga qo'shish mumkin)

Ishchi katalogida bo'lgan barcha o'zgarishlarni staging maydoniga saqlash uchun - "`git add -A`" ishlatiladi

Staging maydonidan faylni olib tashlash uchun - "`git rm --cached <filename>`" ishlatiladi.

Loyihamizni ayni bir vaqtdagi holatini saqlab turishimiz uchun "`git commit`" buyrug'idan foydalanamiz.

Commit vaqtida qanaqa o'zgarishlar qo'shilganini biz "`git commit -m "xabar shu yerda yoziladi"`" buyrug'idan foydalanamiz.

Commit vaqtida "Author identify unknown" degan xatolik chiqishi mumkin. Bu gitga kimligingizni bildirishingiz kerak degani buni quyidagi:

`git config --global user.email "youremail@mail.com"` - email kiritish yoki

`git config --global user.name "Your Name"` - ism kiritish orqali bajarishingiz mumkin.

Gitimizda bo'lgan commitlarni “ `git log` ” orqali ko'rishimiz mumkin.

Loyihamizni ma'lum bir commit holatiga qaytish uchun - “ `git checkout <commit_hash>` ” buyrug'i yordamida amalga oshiramiz.

Qilgan o'zgarishlarni ortga qaytarish uchun - “ `git revert` ” dan foydalanamiz.

Git kuzatmaydigan kodalar **.gitignore** fayl ichida saqlanadi. Bu bizga boshqalar ko'rishi kerak bo'lmagan kalitlar, parollar va boshqalarni saqlab turishi mumkin.

“ `git remote` ” buyrug'i bu jamoada boshqa dasturchilar qilgan o'zgarishlarni ko'rish, yaratish va o'chirish imkoniyatini beradi.

Masofadagi repoga bog'lanish uchun “ `git remote add <name> <url>` ” - buyrug'idan foydalanamiz.

Local o'zgarishlarni masofaviy o'zgarishlarga o'tkazish uchun - “ `git push` ” ishlatamiz.

“ `git fetch` ” - yordamida masofaviy commitlarni mahalliy (local) repoga yuklab olamiz.

“ `fetch` ” - buyrug'i yordamida jamoadagi boshqa dasturchilar nimani ustida ishlashayotganini ko'rishimiz mumkin.

Remote repodan yuklab olingan ma'lumotlarni ko'rish uchun “ `git checkout` ” buyrug'idan foydalanamiz.

“ `git pull` ” - buyrug'i masofaviy repodan ma'lumotlarni local repoga yuklab olib va darhol local reponi yangilash uchun ishlatiladi. Bu asosan “git fetch” va “git merge” burug'larining birlashmasi hisoblanadi.

Gitda conflict ya'ni nizoni terminalda hal etish uchun “ `git mergetool` ” buyrug'idan foydalanamiz.

“ `:wq` ” - ID dan chiqib ketish;

“ `git checkout -b branch-name` ” - buyrug'i orqali tarmoq yaratish va unga o'tish imkoniyatini beradi.

“ `git push origin branch-name` ” - orqali remote'da branch yaratib, unga o'zgarishlarni yuklash imkonini beradi.

Oxirgi commit qilingan xabarni tahrirlash uchun - “ `git commit --amend -m 'Tahrirlangan xabar'` ” buyrug'ini ishlatamiz.

Mahalliy tarmoqni o'chirish uchun - “ `git branch -d feature/login` ”
Masofaviy tarmoqni o'chirish uchun - “ `git push origin --delete features/login` ” ishlatiladi.

Staging areaga qo'shilgan o'zgarishlarni olib tashlash uchun - “ `git reset index.html` ” buyrug'ini ishlatamiz.

Tarmoq nomini o'zgartirish uchun - “ `git branch -m <old-name> <new name>` ” ishlatiladi.

Davlatshoh Baxtiyorov | davlatshohbakhtiyorov@gmail.com | davlatshoh.uz