

Git va GitHub bilan ishlash bo'yicha to'liq qo'llanma

Mundarija

1. [Git nima?](#)
 2. [GitHub nima?](#)
 3. [Asosiy tushunchalar](#)
 4. [Git o'rnatish](#)
 5. [Birinchi sozlamalar](#)
 6. [Repository yaratish](#)
 7. [Asosiy buyruqlar](#)
 8. [GitHub bilan ishlash](#)
 9. [Amaliy misollar](#)
 10. [Keng tarqalgan xatolar](#)
-

Git nima?

Git – bu kod versiyalarini boshqarish tizimi. Oddiy qilib aytganda:

- Kodingizni saqlaydi
- O'zgarishlarni kuzatib boradi
- Eski versiyalarga qaytishga imkon beradi
- Bir necha kishi birgalikda ishlashiga yordam beradi

Misol: Siz essay yozayotgan bo'lsangiz va har safar saqlashda "essay_v1.docx", "essay_v2.docx" kabi nomlar berasiz. Git aynan shu ishni avtomatik qiladi, lekin ancha aqlli tarzda.

GitHub nima?

GitHub – bu internetdagi Git repositorylarini saqlash xizmati. Bu xuddi:

- Google Drive – fayllar uchun
- GitHub – kodlar uchun

GitHub nimalar beradi:

- Kodlaringizni internetda saqlash
 - Boshqalar bilan hamkorlik qilish
 - Loyihalaringizni ko'rsatish
 - Bepul hosting
-

Asosiy tushunchalar

Repository (Repo)

- Loyihangizning barcha fayllarini saqlaydigan papka
- `.git` papkasi bor bo'lsa, u repository

Commit

- O'zgarishlarni saqlash
- Xuddi o'yindagi "Save" tugmasi kabi

Branch

- Kodning alohida nusxasi
- Asosiy branch `main` yoki `master` deyiladi

Clone

- Internetdagi repositoryni kompyuteringizga ko'chirish

Push

- O'zgarishlarni internetga yuklash

Pull

- Internetdan yangi o'zgarishlarni olish
-

Git o'rnatish

Windows uchun:

1. git-scm.com ga kiring
2. "Download" tugmasini bosing
3. Yuklab olingan faylni ishga tushiring
4. Hammasi standart bo'yicha qoldirib o'rnatib

macOS uchun:

```
# Homebrew orqali  
brew install git
```

```
# Yoki Xcode Command Line Tools  
xcode-select --install
```

Linux uchun:

```
# Ubuntu/Debian  
sudo apt install git
```

```
# CentOS/RHEL  
sudo yum install git
```

Tekshirish:

```
git --version
```

Birinchi sozlamalar

Git o'rnatgandan keyin o'zingizni taniting:

```
git config --global user.name "Sizning Ismingiz"  
git config --global user.email "sizning@email.com"
```

Muhim: Bu ma'lumotlar har bir commitda ko'rinadi!

Repository yaratish

1-usul: Lokalda yaratish

```
# Yangi papka yarating
mkdir mening-loyiham
cd mening-loyiham

# Git repository qiling
git init
```

2-usul: GitHub'dan klonlash

```
git clone https://github.com/username/repository-name.git
```

Asosiy buyruqlar

Holat ko'rish

```
git status
```

Bu buyruq sizga:

- Qaysi fayllar o'zgargan
- Qaysi fayllar commit uchun tayyor
- Hozirgi branch nomini ko'rsatadi

Fayllarni qo'shish

```
# Bitta fayl qo'shish
git add fayl.txt

# Barcha fayllarni qo'shish
git add .
```

```
# Ma'lum turdagi fayllarni qo'shish  
git add *.js
```

Commit qilish

```
git commit -m "O'zgarishlar haqida qisqa tavsif"
```

Yaxshi commit xabarlari:

- "Login funksiyasi qo'shildi"
- "Xato tuzatildi"
- "CSS dizayni yangilandi"

O'zgarishlar tarixini ko'rish

```
git log  
git log --oneline # Qisqa ko'rinish
```

Fayllarni olib tashlash

```
# Faylni o'chirish va commitga qo'shish  
git rm fayl.txt  
  
# Faqat Git'dan olib tashlash (fayl qoladi)  
git rm --cached fayl.txt
```

GitHub bilan ishlash

GitHub'da repository yaratish

1. github.com ga kiring
2. Ro'yxatdan o'ting (bepul)
3. "New repository" tugmasini bosing
4. Nom bering (masalan: `mening-birinchi-repo`)
5. "Create repository" tugmasini bosing

Lokaldan GitHub'ga yuklash

```
# GitHub repository bilan bog'lash
git remote add origin https://github.com/username/repository-name.git

# Birinchi marta yuklash
git branch -M main
git push -u origin main

# Keyingi safalar
git push
```

GitHub'dan yangilanishlarni olish

```
git pull origin main
```

Amaliy misollar

Misol 1: Oddiy veb-sayt loyihasi

```
# 1. Papka yaratish
mkdir mening-sayti
cd mening-sayti

# 2. Git boshlash
git init

# 3. HTML fayl yaratish
echo "<h1>Salom Dunyo!</h1>" > index.html

# 4. Faylni qo'shish
git add index.html

# 5. Commit qilish
git commit -m "Birinchi HTML sahifa qo'shildi"

# 6. GitHub bilan bog'lash
git remote add origin https://github.com/username/mening-sayti.git
```

```
# 7. Yuklash
git push -u origin main
```

Misol 2: Mavjud loyihani GitHub'ga yuklash

```
# Loyiha papkasiga kirish
cd mavjud-loyiha

# Git boshlash
git init

# Barcha fayllarni qo'shish
git add .

# Commit qilish
git commit -m "Loyiha GitHub'ga yuklandi"

# GitHub bilan bog'lash
git remote add origin https://github.com/username/loyiha-nomi.git

# Yuklash
git push -u origin main
```

Misol 3: O'zgarishlar kiritish

```
# Faylni tahrirlash (masalan, index.html)
# ... kod yozish ...

# O'zgarishlarni ko'rish
git status
git diff

# Qo'shish va commit qilish
git add index.html
git commit -m "Dizayn yangilandi"

# GitHub'ga yuklash
git push
```

Branch'lar bilan ishlash

Yangi branch yaratish

```
git branch yangi-funksiya
git checkout yangi-funksiya

# Yoki qisqaroq:
git checkout -b yangi-funksiya
```

Branch'lar ro'yxati

```
git branch
```

Branch o'zgartirish

```
git checkout main
```

Branch'larni birlashtirish

```
# main branch'ga o'tish
git checkout main

# Boshqa branch'ni birlashtirish
git merge yangi-funksiya
```

Branch o'chirish

```
git branch -d yangi-funksiya
```

Foydali buyruqlar

O'zgarishlarni bekor qilish


```
# Fayl o'zgarishlarini bekor qilish
git checkout -- fayl.txt

# Oxirgi commitni bekor qilish
git reset --soft HEAD~1

# Barcha o'zgarishlarni bekor qilish (EHTIYOT!)
git reset --hard HEAD
```

Fayllarni e'tiborsiz qoldirish

`.gitignore` fayli yarating:

```
# Node.js
node_modules/
*.log

# Python
__pycache__/*
*.pyc

# IDE
.vscode/
.idea/

# OS
.DS_Store
Thumbs.db
```

Qisqa ma'lumot

```
git show      # Oxirgi commit haqida
git diff      # O'zgarishlarni ko'rish
git blame fayl.txt # Kim nimani o'zgartirgan
```

Keng tarqalgan xatolar

1. "fatal: not a git repository"

Sabab: Siz git repository bo'lmagan papkada turibsiz

Yechim:

```
git init # yoki to'g'ri papkaga o'ting
```

2. "Author identity unknown"

Sabab: Git sozlanmagan

Yechim:

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

3. "Permission denied (publickey)"

Sabab: SSH kaliti sozlanmagan

Yechim: HTTPS ishlatng:

```
git remote set-url origin https://github.com/username/repo.git
```

4. "Your branch is ahead of origin/main"

Sabab: Lokalda commitlar bor, GitHub'da yo'q

Yechim:

```
git push
```

5. "Merge conflict"

Sabab: Bir xil faylni ikki kishi o'zgartirgan

Yechim:

1. Faylni oching
2. <<<<<< , ===== , >>>>>> belgilarini toping
3. Kerakli kodni qoldiring
4. `git add fayl.txt`
5. `git commit -m "Konflikt hal qilindi"`

GitHub'ning qo'shimcha imkoniyatlari

Issues (Masalalar)

- Bug'larni qayd qilish
- Yangi funksiyalar so'rash
- Muhokama qilish

Pull Request (PR)

- O'zgarishlarni taklif qilish
- Kod tekshirish
- Muhokama va birlashtirish

GitHub Pages

- Bepul veb-sayt hosting
- [Settings](#) > [Pages](#) bo'limida yoqing

Actions

- Avtomatik test va deploy
- CI/CD pipeline

Tezkor cheat sheet

Asosiy buyruqlar

`git init`

`git add .`

`git commit -m "xabar"`

`git push`

`git pull`

`git status`

`git log`

Repository boshlash

Barcha fayllarni qo'shish

Commit qilish

GitHub'ga yuklash

GitHub'dan olish

Holat ko'rish

Tarix ko'rish

Branch'lar

`git branch`

`git checkout -b yangi`

`git merge branch-nomi`

Branch'lar ro'yxati

Yangi branch yaratish

Branch birlashtirish

```
# Remote
git remote add origin URL      # GitHub bilan bog'lash
git clone URL                  # Repository klonlash

# Bekor qilish
git reset --soft HEAD~1        # Oxirgi commitni bekor qilish
git checkout -- fayl.txt       # Fayl o'zgarishini bekor qilish
```

Keyingi qadamlar

1. **Amaliyot qiling:** Har kuni biror narsa commit qiling
 2. **Open Source'ga qo'shiling:** Boshqalarning loyihalariga hissa qo'shing
 3. **Advanced Git:** Rebase, stash, hooks kabi mavzularni o'rganing
 4. **Git GUI:** GitKraken, SourceTree kabi vizual dasturlarni sinab ko'ring
-

Yordam kerak bo'lsa

- **Git dokumentatsiyasi:** git-scm.com/docs
 - **GitHub qo'llanma:** docs.github.com
 - **Savollar:** Stack Overflow, GitHub Discussions
-

Eslatma: Bu qo'llanma bilan siz Git va GitHub'ning 95% imkoniyatlarini o'rganasiz. Qolgan 5% tajriba bilan keladi!

Git va GitHub bilan omad tilaymiz!