

Git 1: Bevezetés, Telepítés és Alap Parancsok

1.1 Git Telepítés

A Git telepítéséhez látogass el a hivatalos weboldalra:

<https://git-scm.com/downloads>

- **Windows:** A telepítő végigvezet a folyamaton, érdemes a "Git "-t is telepíteni.
- **Linux:** Telepíthető csomagkezelővel (pl. Ubuntu/Debian: `sudo apt install git`).
- **MacOS:** Telepíthető Homebrew segítségével: `brew install git`.

Telepítés után ellenőrizheted a verziót:

```
git --version
```

Első beállítások:

```
git config --global user.name "Saját Név"  
git config --global user.email "email@domain.com"
```

Ezek az adatok minden commitnál szerepelni fognak.

1.2 Alapvető Git Parancsok

1 Új Git repository létrehozása

```
git init
```

Ez a parancs létrehoz egy `.git` könyvtárat, amely tárolja a verziókezeléshez szükséges adatokat.

2 Jelenlegi státusz lekérése

```
git status
```

Megmutatja a fájlok állapotát (módosított, új stb.).

3 Fájlok hozzáadása a verziókezeléshez

```
git add <fájlnév>      # Egy fájl hozzáadása  
git add .               # Az összes módosított fájl hozzáadása
```

4 Commit készítése (változások rögzítése)

```
git commit -m "Első commit"
```

A `-m` után egy rövid üzenetet kell megadni.

5 Commitok előzményeinek megtekintése

```
git log
```

Az összes eddigi commitot listázza.

6 Egy commit részleteinek megtekintése

```
git show <commit_hash>
```

A commit hash kódját a `git log` parancsból másolhatjuk.

1.3 Git GUI

A Githez több grafikus felület is elérhető, például:

- **GitKraken** (<https://www.gitkraken.com/>)
- **SourceTree** (<https://www.sourcetreeapp.com/>)
- **GitHub Desktop** (<https://desktop.github.com/>)

Ezek vizuális felületek a commitok, branch-ek és távoli repository-k kezelésére.

Git 2: Branch, Checkout, Stash, GitHub Használata

2.1 Branch és Checkout

A **branch-ek** lehetővé teszik, hogy párhuzamos fejlesztéseket végezzünk anélkül, hogy az alapprojektet érintenénk.

1 Új branch létrehozása és átváltás

```
git branch feature-uj-funkcio # Új branch létrehozása  
git checkout feature-uj-funkcio # Átváltás az új branch-re
```

Vagy egyben: `git checkout -b feature-uj-funkcio`

2 Branch-ek listázása

```
git branch
```

A csillag jelzi az aktív branch-et.

3 Branch törlése

```
git branch -d feature-uj-funkcio
```

2.2 Stash – ideiglenes változások mentése

Ha módosításokat végeztél, de még nem akarsz commitolni, **stash-be** teheted őket:

```
git stash
```

Ezzel a változások eltűnnek a working directory-ből, de nem vesznek el.

Visszaállítás:

```
git stash pop
```

Vagy listázás és visszaállítás:

```
git stash list  
git stash apply stash@{0}
```

2.3 GitHub Regisztráció és Alapok

Regisztráció

Menj a <https://github.com/> oldalra, hozz létre egy fiókot.

SSH kulcs beállítása

Ha nem akarsz minden push-nál megadni a jelszót, használhatsz SSH-kulcsot:

```
ssh-keygen -t rsa -b 4096 -C "email@domain.com"
```

A létrejött ~/.ssh/id_rsa.pub tartalmát add hozzá a GitHub profilodhoz (Settings → SSH Keys).

Új GitHub Repository Klónozása

```
git clone https://github.com/felhasznalo/repo.git
```

Ez letölti a távoli repository tartalmát.

Helyi repository feltöltése GitHubra

```
git remote add origin https://github.com/felhasznalo/repo.git  
git push -u origin main
```

A `-u origin main` beállítja az alapértelmezett távoli branch-et.

Frissítések letöltése távolról

```
git pull origin main
```

2.4 Alkalmazás Installálása

A GitHub lehetőséget ad arra, hogy egyes alkalmazásokat integrálj a repository-hoz (pl. **GitHub Actions**, **CodeQL**, **Dependabot**).

🔗 **GitHub Actions:** Automatizált workflow-k beállítása (CI/CD).

🔗 **Dependabot:** Automatikusan frissíti a függőségeket.

Ezeket a GitHub **Settings** → **Integrations & Services** menüben lehet aktiválni.

Összegzés

Git telepítése és első beállításai

Alapvető parancsok: `init`, `add`, `commit`, `log`, `show`

Branch kezelés: `branch`, `checkout`

Ideiglenes mentés `stash` segítségével

GitHub regisztráció és repository feltöltése