

Git Verziókezelés – Folyamatleírás

☞ Szükséges szoftver:

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

Általános tudnivalók

A Git egy parancssoros verziókezelő rendszer, amely segít a fejlesztési folyamat során a fájlok változásait nyomon követni, verziózni és visszaállítani. A GitHub pedig egy online platform, ahol a Git-repozitóriumokat tárolhatjuk, megoszthatjuk, és együttműködhetünk másokkal.

Git használatának lépései

0. Git telepítése

Töltsd le és telepítsd a Git-et a hivatalos oldalról:

☞ <https://git-scm.com/>

A telepítés után a **parancssor** (pl. Command Prompt, Git Bash) segítségével dolgozunk.

1. Bejelentkezés a GitHub fiókba

Állítsd be a GitHub felhasználódat a következő parancsokkal:

```
git config --global user.name "FelhasználóNév"
git config --global user.email "email@valami.com"
```

2. Projekthez tartozó GitHub repó lekérése (pull)

Mielőtt dolgozni kezdenél, **mindig először húzd le a legfrissebb változatot** a GitHubról:

```
git pull origin main
```

⚠ Ezzel elkerülöd, hogy más által végzett módosításokkal konfliktus alakuljon ki.

3. Navigálás a projekt mappába

Lépj be a projekt könyvtárba:

```
cd <elérési_út>
```

4. Git inicializálása

Ha új projektről van szó, indítsd el a Git-et:

```
git init
```

Ez létrehozza a helyi Git-repozitóriumot, ahol a változásokat nyomon lehet követni.

5. Fájlok hozzáadása a követéshez

```
git add <fájlnév>
```

Több fájl hozzáadásához használhatod a `git add .` parancsot, ami az összes változást hozzáadja.

6. Fejlesztői ág létrehozása

Hozz létre egy külön fejlesztői ágot:

```
git branch dev  
git switch dev
```

Itt dolgozol a fejlesztéseden, módosításaidat ide mented.

7. Változások mentése (commit)

Ha elkészültél, és a változások tesztelve lettek:

```
git commit -m "Változások leírása"
```

8. Visszaváltás a main ágra

```
git switch main
```

9. Frissítés a GitHub-ról

Újra húzd le a `main` ágot a GitHubról, hogy biztosan naprakész legyen:

```
git pull origin main
```

10. Merge – Fejlesztések egyesítése

```
git merge dev
```

Ez egyesíti a `dev` ágon végzett fejlesztéseidet a `main` ággal.

11. Feltöltés a GitHub-ra

```
git push origin main
```

Ezzel a változások felkerülnek a távoli GitHub repóba.

✓ Problémák elkerülve!

Ha követed ezt a folyamatot, elkerülheted a legtöbb verzióütközést, adatvesztést és fejlesztési problémát.

💡 Tipp

Mindig legyen mentésed, és mielőtt módosítasz, húzd le a legfrissebb verziót!

Koszta Félixről a csapatnak 😊