1. Előadás: Verziókezelés

Regisztráció GitHub-ra

https://github.com/join

Git telepítése

Git letöltése: https://git-scm.com/downloads
Asztali kliens: https://desktop.github.com/

Git konfigurálása

Parancssorban

Név hozzáadása:

```
git config --global user.name "Gipsz Jakab"
```

E-mail cím hozzáadása:

```
git config --global user.email "gipsz.jakab@gmail.com"
```

Asztali kliensen

File / Options / Accounts / Sign in / Itt hitelesíteni kell a Githubot a böngészővel

Új tárhely létrehozása

Asztali kliensen

File / New repository

1. **Name**: a tárhely neve
2. **Description**: a tárhely rövid leírása
3. **Local path**: a tárhely mappája a helyi számítógépen
4. **Git ignore**: a gitignore-hoz adott fájlok nem lesznek feltöltve a tárhelyre. Itt érdemes kiválasztani az adott programnyelvhez tartozó ignore konfigurációt. A .gitignore a tárhely gyökerében lévő szöveges állomány, amihez saját magunk is hozzáírhatunk ha vannak olyan fájlok, amelyeket nem szeretnénk feltölteni a tárhelyre. Ilyenek lehetnek a nagyméretű állományok vagy személyes információt / konfigurációt tartalmazó állományok.
5. **License**: milyen másolhatósági / továbbadhatósági szabályok vonatkoznak a tárhelyre. Ezt legtöbb esetben elhagyhatjuk.

Ezután a tárhely létrejön a számítógépen. Ha megadtuk az e-mail címet és nevet a Git parancssoros területén és ez egyezik a GitHub-on használt e-mail címmel és jelszóval, a parancssor fel fogja ismerni a felhasználót, és lehetséges parancssorból elvégezni a Git műveleteket.

A legtöbb (de nem minden) műveletet ugyanúgy el lehet végezni parancssoron mint az asztali felületen. A kurzus során a hallgató választja meg, melyiket használja. A továbbiakban viszont az egyszerűség kedvéért a parancssoros műveleteket fogjuk taglalni.

Gitignore szerkesztése

Belépés a tárhely gyökerébe:

```
cd c/Users/Jakab/GitHub/desktop-tutorial
```

Új ignore fájl létrehozása:

```
nano .gitignore
```

Ezitán minden sorban egy ignorált fált lehet megadni, vagy a * karakterrel egy általános definíciót megadni:

```
videos/big_video.mp4
passwords/*
*.zip
*.rar
*.jpg
```

Mentés és kilépés a szerkesztőből:

```
ctrl + o
ctrl + x
```

Ezek a definíciók példa jellegűek. A valósan ignorált fájlok és típusok tárhelytől függően változhatnak.

Változtatások elmentése és feltöltése a tárolóra

A tárhely gyökerébe navigálás:

```
cd c/Users/Jakab/GitHub/desktop-tutorial
```

Változtatások mutatása a parancssoron:

```
git status
```

Változtatások hozzáadása a Git indexhez (megjelölés commit-ra):

```
git add .
```

Commit valamilyen üzenettel (a -m a message szóra utal):

```
git commit -m "valamilyen üzenet"
```

Feltöltés a távoli tárolóra, adott fejlesztési ágra:

```
git push origin main
```

Ahol az origin a távoli tárhelyet jelöli, a main pedig a fejlesztési ágat

Változtatások egy új fejlesztési ágon

Új fejlesztési ág létrehozása:

```
git checkout -b new_branch
```

Jelenlegi fejlesztési ág lekérdezése (itt a new_branch kell legyen megjelölve):

```
git branch
```

Opcionálisan vissza lehet váltani egy már létező ágra:

```
git checkout main
git checkout new_branch
```

Változtatások hozzáadása és feltöltése az új ágra:

```
git add .
git commit -m "valamilyen üzeenet"
git push origin new_branch
```

Fejlesztési ágak összeolvasztása

Ebben az esetben az előfeltételezés, hogy legalább 2 ág van a tárhelyen, és ezek mindegyikén már legalább 1 commit van.

github.com webes felületen

1. Belépés a tárhelyre www.github.com/jakab/desktop-tutorial
2. Fejlesztési ágak megtekintése: **2 branches**
3. Itt a new_branch mellett: **New pull request**
4. A branchek beállítása után: **Create pull request**
5. Ha az összefésülés lehetséges: **Merge pull request**
6. **Confirm merge**
7. Fejlesztési ág törlése: **Delete branch**

Ha a fejlesztési ágakat nem lehetséges összeolvasztani, a konfliktusokat kézzel kell feloldani.