

# Git képzés

2. nap

Rostagni Csaba

2022. november 25.

# Tartalom

- 1 Git és GitHub
- 2 SSH
- 3 Markdown
- 4 merge, pull, push
- 5 Conflict

# Git

- Nyílt forráskódú **szoftver**
- Elosztott verziókezelő rendszer
  - **SVN**-nel szemben nincs szükség központi gépre, **minden megtalálható mindenkinél**
- Gyors, egyszerű és hatékony

# GitHhb

- Jogosultság kezelés
- Problémák, ötletek nyomonkövetése
- Automatizálás (tesztelés)
- Biztonság (automatikus ellenőrzés)
- Statisztikák
- Használható ingyenesen

Hasonló szolgáltatások:

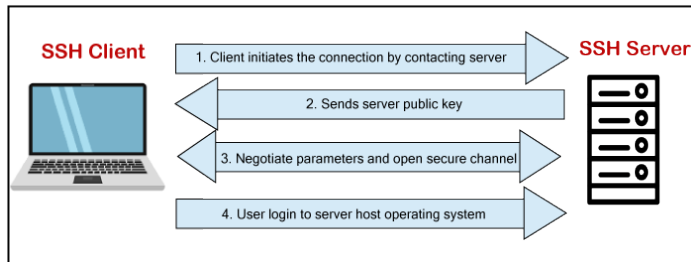
- GitHUB
- BitBucket
- GitLab

# Tartalom

- 1 Git és GitHub
- 2 SSH**
- 3 Markdown
- 4 merge, pull, push
- 5 Conflict

# SSH

- **Secure Shell**
- Az SSH protokoll egy biztonságos kapcsolatot tesz lehetővé
- kliens-szerver architektúra
- publikus és privát kulcspárokat használ



# PS parancsok

- `cd mappa` - mappa váltás
- `cd ..` - váltás a szülő mappa
- `ls` - mappa listázása
- `clip` - vágólap
- `cat` - fájl tartalmának kiírása

# Tartalom

- 1 Git és GitHub
- 2 SSH
- 3 Markdown**
- 4 merge, pull, push
- 5 Conflict



# Markdown

- Egyszerű szöveges leíró nyelv

Linkek:

- [Markdown - wikipedia](#)
- [Eredeti markdown](#)
- [GitHub féle Markdown](#)
- [markdownguide.com](#)

# Címsor

```
#_Címsor_1
```

```
##_Címsor_2
```

```
#####_Címsor_6
```

- A sor első karaktere legyen n darab kettőskereszt, utána szóközzel a cím
- A kettőskeresztek száma adja meg, milyen címsorról van szó
- Előtte és utána legyen egy-egy üres sor

# Számozatlan felsorolás

```
-□alma  
-□dinnye  
-□barack
```

- A felsorolást a sor elejére kitett szóközt követő kötőjellel jelölhetjük
- A felsorolás előtt és után legyen egy-egy üres sor

# Számozott felsorolás

```
1. □alma  
2. □dinnye  
3. □barack
```

- A számozott felsorolást a sor elejére kitett szóközt követő sorszámmal jelölhetjük
- A felsorolás előtt és után legyen egy-egy üres sor

# Automatikus számozás



```
1. □alma  
1. □dinnye  
1. □barack
```

- Összefüggő felsorolás esetében elegendő **1.**-est kirakni minden sor elejére
- Így új pont közbeszúrásakor nem kell újra számozni az egészet

# Kód a szövegben

A ``ctrl` + `s`` billentyűkombináció menti a dokumentumot.

Állítsa be az oldal háttérszínét ``#CCCCCC`` színkódúra.

- A soron belüli kódrészletet egyszerűen backtick-ek (  +  ) közé kell tenni



# Kódblokk

```
Állítsd be az oldal háttérét pirosra
```

```
```
```

```
body {background-color: red}
```

```
```
```

- A kódblokkokat három backtick-et (  +  ) tartalmazó sor határolja
- A blokk előtt és után is legyen egy üres sor

# Szintaxis kiemelése

```
```html
<h1>Címsor 1</h1>
```
```

```
```css
h1 {color: darkblue;}
```
```

- A bevezető három karakter után közvetlen megadható a nyelv típusa
- A felismert nyelveket átalakításkor kiszínezi.



# Karakterformázás

**\*Dőlt\***

**\*\*Félkövér\*\***

**\*\*\*Félkövér és dőlt\*\*\***

- Csillag karakterekkel körbevett szöveg lehet félkövér, dőlt vagy mind a kettő
- A csillag tapadjon hozzá a szöveghez, azaz ne legyen szóköz

# Linkek

```
[Google.com] (http://google.com)
```

```
[feladatok] (./feladatok.html)
```

```
[feladatok] (./feladatok.html "Itt vannak a feladatok")
```

- A linkek szögletes zárójellel kezdődnek.
- A szögletes zárójelben a linkként megjelenő szöveget kell elhelyezni
- A kerek zárójelben adható meg hova mutat a link (href)
- Utána opcionálisan idézőjelben megadható a title értéke.

# Kép

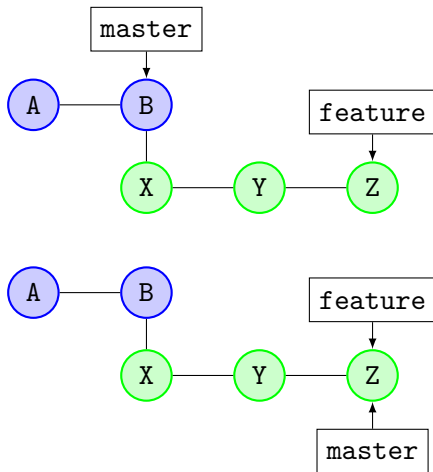
```
![Piros_alma](./alma.png"A_képen_egy_piros_alma_látható")
```

- A képeket felkiáltójellel kell kezdeni
- A szögletes zárójelben az alternatív szöveg adható meg (alt)
- A kerek zárójelben pedig a fájl neve
- Utána opcionálisan idézőjelben megadható a title értéke.

# Tartalom

- 1 Git és GitHub
- 2 SSH
- 3 Markdown
- 4 merge, pull, push**
- 5 Conflict

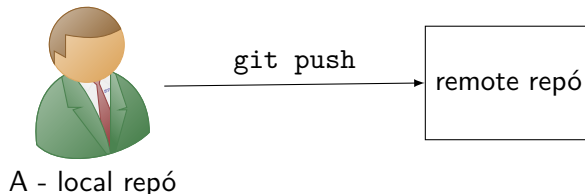
# fast-forward



- A master branch-en nem történt változtatás
- A végén a master és a feature is ugyanoda mutat

# Kezdés lokális repóval

```
git init  
git config user.name "..."  
git config user.email "..."  
git add .  
git commit -m "init"  
git branch -M main  
git remote add origin "..."  
git push -u origin main
```



# Projekt kezdése githubon

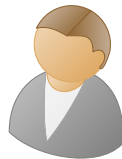
```
git clone  
git add .  
git commit -m "init"  
git push
```



A - local repó

git push

remote repó



B - local repó

# Projekt folytatása (másik személy/gép)

```
git clone  
git add .  
git commit -m "elso"  
git push
```



A - local repó

remote repó

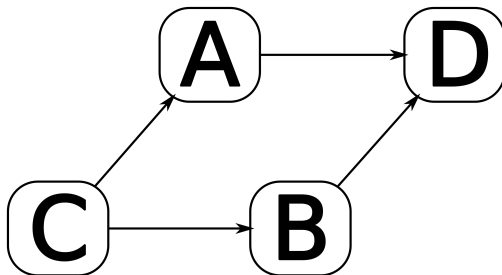
git clone



B - local repó



## 3 way merge

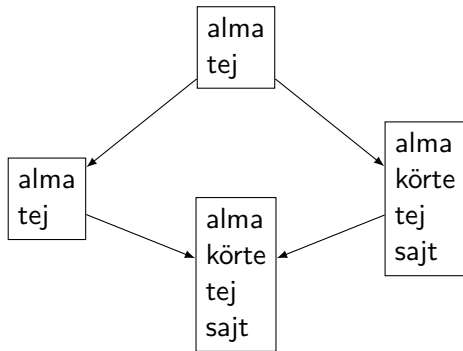


- Az eredet fájl a "C". Az "A" és "B" belőle származnak
- Olyan szakaszokat keres, melyek háromból két fájlban azonosak
- "C" és "A" egyezésekor a "B" érték lesz a kimenet
- "C" és "B" egyezésekor a "A" érték lesz a kimenet

# 3 way merge

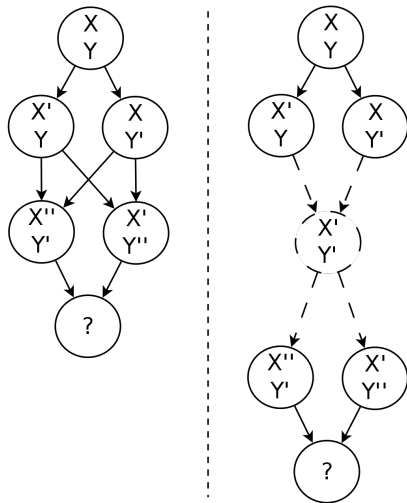


Andrea

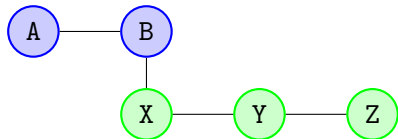


Peti

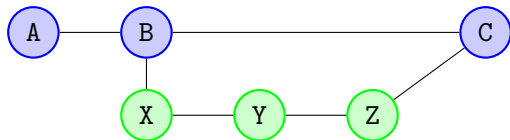
# 3-way (recursive)



# 3-way merge



Utána



- A master branch-en nem történt változtatás
- A végén a master és a feature is ugyanoda mutat

# merge-strategies

- **recursive**

- 3-way merge
- Átnevezéseket kezeli

- **ort**

- Ostensibly Recursive's Twin, a recursive lecserélésére készült
- alapértelmezett

- **ours** ...

Linkek:

- Merge strategies - git dokumentáció

## ort

```
git merge -s ort -Xtheirs
```

```
git pull -s ort -Xtheirs
```

- A -s opcióval adható meg a merge stratégia
- A -X kezdetű opciók az adott stratégiának szolgáltatnak további paramétereket

## ort

```
git pull -s ort -Xtheirs
```

```
git pull -s ort -Xours
```

```
git pull -s ort -Xignore-space-change
```

- A `-s` szerint az `ort` stratégiát alkalmazza
- A `-Xtheirs` szerint a távoli repóból érkezett változásokat fogadja el
- A `-Xours` szerint a távoli repóból érkezett változásokat fogadja el
- A `-Xignore-space-change` szerint a szóközök módosítását figyelmen kívül hagyja

# Tartalom

- 1 Git és GitHub
- 2 SSH
- 3 Markdown
- 4 merge, pull, push
- 5 Conflict**



# Conflicts



Andrea

alma  
banán  
tej  
liszt

alma  
tej

alma  
körte  
tej  
sajt



Peti

- Ugyanazt a fájlt szerkesztette két felhasználó
- Azonos pontotn szerkesztették
- A módosítások automatikus kezelése nem sikerült, ütközés keletkezett.

# Ütközés jelölése a fájlban

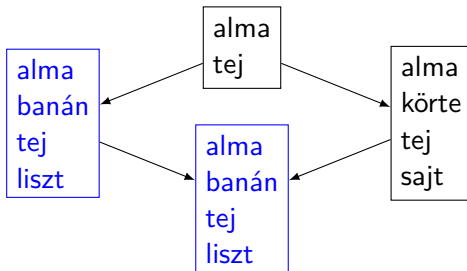
```

alma
<<<<<< HEAD
körte
tej
sajt
=====
banán
tej
liszt
>>>>>> fd6348274291f0a67b034904e35c99ee6b858e32

```

- A feloldandó konfliktusokat  
 <<<<<<,  
 ">>>>>>" és  
 "======" jelek határolják
- A fájl akár teljesen manuálisan is szerkeszthető

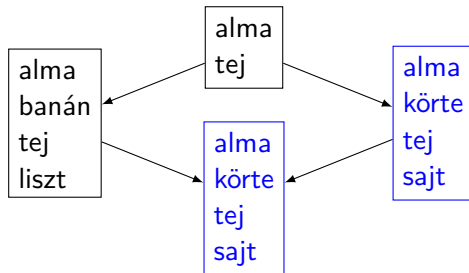
# Saját módosításaink elfogadása



```
git pull  
git merge -s ort -Xours
```

- Andrea elfogadta a **saját** módosításait, ezzel felülírva a többiekét

# Mások módosításaink elfogadása



```
git pull  
git merge -s ort -Xours
```

- Andrea elfogadta a többiek módosításait, ezzel felülírva a sajátját

# Véglegesítés

```
git add .  
git diff --cached  
git commit -m "merge"  
git push
```

- Adjuk hozzá a módosításokat a staging areához
- Utolsó ellenőrzése commitra felkészített módosításoknak
- Commit
- Módosítások pusholás a távoli repóba