



## Lo: Fejlesztői eszközök és kellékek

Ezen laborgyakorlat célja a fejlesztői eszközök telepítése és példaprogramok kipróbálása. A laborgyakorlatok megoldásához használhatunk Windows, Linux vagy Mac OS X operációs rendszert.

### Fejlesztői eszközök

A webprogramozás tantárgy laborfeladatainak sikeres megoldásához szükségünk lesz különböző szoftver eszközökre, amelyeket otthoni használat céljából telepítenünk kell a saját számítógépünkre. A legfontosabb eszközök listája a következő:

1. Legalább két modern legfrisebb verziójú böngésző: [Firefox](#), [Chrome](#), [Safari](#), [Opera](#), stb. Mivel a HTML oldalak megjelenítésében adódnak különbségek böngészőtől függően, mindenképpen érdemes a házikat több böngészőben is ellenőrizni. Legalább a legelterjedtebb böngészőket érdemes figyelembe venni.
2. Szintaxis kiemeléssel (syntax highlighting) rendelkező szövegszerkesztő: leginkább ajánlott a [Visual Studio Code](#).
3. [node.js](#) JavaScript runtime
  - ajánlott a legfrissebb *LTS* (long term support) verzió használata
  - kipróbálható konzolról a [node](#), [npm](#) és [npx](#) parancsok segítségével
4. [git verziókövető](#). Használati opciók:
  - konzolból a velejáró *Git Bash* segítségével;
  - a szerkesztő pluginja segítségével – [git VS Code](#)-ból;
  - külső git kliensalkalmazás, pl. [SourceTree](#), [GitKraken](#), stb.
5. HTTP hívások tesztelésére [Postman](#)
6. [MySQL Community](#) adatbáziskezelő rendszer (későbbi feladatoknál szükséges).

## Példaprogramok

Az előadás különböző tematikájához tartozó példaprogramok elérhetőek egy [publikus git tárolón](#) keresztül. **További részletek a gittel való munkáról a [Konvenciók/ubb-web-git-examples.pdf](#) állományban.**

---

### Szerkezet

- A példaprogramok tematika szerint vannak csoportosítva. Minden tematikának egy külön könyvtár felel meg, ezen belül megtalálhatóak az aktuális példaprogramok.
- Ahol megjelennek, az előző évek példaprogramjai a webes standardok gyors változásának köszönhetően több esetben is elavultak, ezért használatuk **nem** ajánlott.

---

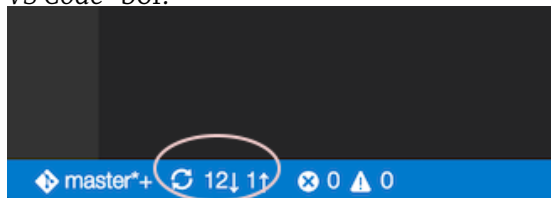
### Használat

#### Első letöltés

- git konzolparancs:
  - navigáljunk azon mappába, ahova letöltenénk a példákat: `cd <celpont>`
  - letöltés: `git clone https://gitlab.com/ubb-webprog/examples` (létrehoz egy `examples` almappát)
- VS Code-ból:
  - Command palette: `Ctrl+Shift+P`
  - „Git: Clone”
  - Repo URL: `https://gitlab.com/ubb-webprog/examples`

#### A tartalom frissítése

- git konzolparancs:
  - navigáljunk azon mappába, ahova korábban letöltöttük a példákat: `cd <celpont>/examples`
  - frissítés: `git pull`
- VS Code-ból:



#### Hiba esetén

Hiba léphet fel, hogyha a lokális tárolóban levő tartalom *megváltozott* (ha kézzel változtattok a példaprogramok tartalmán). A könyvtár *tisztításáért* hívjátok a következőt ugyanazon `examples` mappából (**Vigyázat!**: Ezen parancsok meghívása a laborfeladatos tárolóban kitörölheti annak módosításait):

```
git reset --hard HEAD
git clean -xfd
```

## Saját laborfeladatos tároló

- A laborfeladatokat a GitLab verziókövető portálra kell feltölteni. Minden diáknak automatikusan készül egy-egy projekt, melynek URL-je: <https://gitlab.com/ubb-webprog/labs2021/lab-<abcd1234>> ahol az [abcd1234](#) megfelel a felhasználói SSID-toknak. Ide töltsétek fel gittel a [develop](#) ágra (ez lesz használva automatikusan) minden megoldást.
- Ahhoz hogy a projekthez elérést nyerjeteek, [készítsetek felhasználót a GitLabon](#), majd adjátok le az itt használt **e-mail címeteket** az oktatónak.
- A laborfeladatokhoz tartozik egy *merge request* (automatikusan létrehozva általunk). A feladatokat **inkrementálisan** ajánljuk megoldani a kurzusok mentén. A találkozási alkalmak között opcionálisan lehet kérni review-t az aktuális tartalomra.
- Az értékelésnél kizárólag a GitLabra felpusholt állományokat vesszük figyelembe. A leadás dátumának az *utolsó commit* időpontja számít.

## Gyakorlat

1. Klónozzuk mind a saját projektet, mind a példaprogramokat tartalmazó git projektet. **Ezek ne ágyazódjanak egymásba - ne a példarepó belsejébe klónozzuk a sajátot, vagy fordítva!** Extra információ: [Konvenciók/ubb-web-git-examples.pdf](#)
2. A fenti eszközök telepítése után telepítsük a [serve](#) globális npm csomagot: `npm install --global serve`.
3. Másoljuk le a példaprojekt [1-htmlcss/html5-basic/index.html](#) állományát a saját repónkba. Írjuk bele saját információinkat és tetszőleges tartalmat.
4. Konzolban navigáljunk ebbe a folderbe, vagy indítsuk el a [serve](#) parancsot. Bizonyosodjunk meg róla, hogy **mind böngészőből, mind Postmanből** sikerül az [index.html](#) állományt lekérjünk. Alapértelmezetten az 5000-es porton figyel a [serve](#) alkalmazás.
5. Használjuk a böngészőnk fejlesztői konzolját az oldal szerkezetének kielemezésére (általában [F12](#)). Próbáljuk ki az egyes elemek tulajdonságainak módosítását a fejlesztői konzolon keresztül.

## Bónusz: SSH használata gittel

Annak érdekében, hogy ne kelljen minden pull/push operációkor *felhasználót és jelszót* megadni a gitnek, konfigurálható az SSH-s elérés. Ennek lépései:

1. Lokálisan a *Git Bash*-ből: `ssh-keygen` - minden kért argumentumot **üresen hagyva**.
2. `cat ~/.ssh/id_rsa.pub` - ez kiírja a publikus SSH kulcsot.
3. A profilodon a *Settings* -> *SSH Keys* részbe másold be az SSH kulcsot *teljességében* (`ssh-rsa` kulcsszó az elején, a géped neve a végén).
4. A projekt oldalán másold le az SSH-s elérési linket - ennek formája `git@gitlab.com:ubb-webprog/...`
5. A fenti HTTP-s `clone` parancs helyett: `git clone <link>` Ha már klónoztad a repót korábban, akkor annak folderén belül: `git remote set-url origin <link>`