

# GIT-et bemutató weboldal feladatai

Git jelentősége

Telepítése

Röviden a GIT-ről

Parancssor

VS Code

Repository

Local repository

Remote repository



## GIT verziókezelő rendszer

Linus Torvalds

A GIT-et még Linus Torvalds kezdte el fejleszteni a Linux kernel fejlesztéséhez, mert 2005-ben nem volt olyan ingyenesen használható verziókezelő rendszer, amely elég gyors lett volna.

Verziókezelő

A Git egy nyílt forráskódú, elosztott verziókezelő szoftver, vagy másképpen egy szoftverforráskód-kezelő rendszer, amely a sebességre helyezi a hangsúlyt. A GIT Windows változatának karbantartásáért a Microsoft felel.

Igen elterjedt

Számos nagy volumenű projekt használja; a legfontosabbak ezek közül: Linux-rendszermag, GNOME, Samba, MediaWiki, VLC media player, Ruby on Rails és az Android platform. A Git karbantartásának felügyeletét jelenleg Junio Hamano látja el.

### A GIT jelentősége

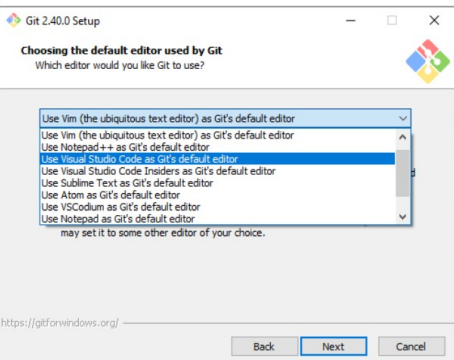
A verziókezelés nem más, mint amikor projektünk egy állapotáról, amihez egy címkét is illeszthetünk (nem kötelező címkét hozzárendelni), készítünk egy mentést, hogy ha szükséges lenne, egyszerűen vissza tudjunk állni arra az állapotra, arra a verzióra. A GIT ebben nyújt nekünk segítséget.

- Mindig tudjuk, melyik a forráskód legfrissebb verziója. Akkor is, ha egyszerre nagyon sokan dolgoznak rajta.
- Több verziót is karban tudunk tartani. Például egy fejlesztői, egy tesztelési, valamint egy éles használatra kiadott verziót. És akkor sem lesz kaotikus, ha az éles verzió gyorsan ki kell valamit javítani, amit aztán később nyilván a fejlesztői és tesztelési verziókba is át kell venni.
- Bármikor vissza tudunk térni korábbi verzióhoz. Például ha a nagy bemutató előtt 4 órával hirtelen semmi nem működik, akkor a pánikhangulat helyett bármikor vissza lehet térni egy korábban agyontesztelt verzióhoz, mely ugyan nem tud még mindent, de akkor legalább működött.

### A GIT telepítése

Telepítése rendkívül egyszerű. A <https://git-scm.com/> oldalról le kell tölteni az operációs rendszerhez illeszkedő változatot, és az alapértelmezett beállításokat elfogadva installálni.

Egyedül az alapértelmezett szövegszerkesztőnél ajánlott egy, a *vim* helyett egy kezdők számára, könnyebben kezelhető változatot kijelölni.



### A GIT-nek 3 fő oszlopa van

A munkakönyvtár (**working directory**) – Ez az a könyvtár, amely a projektünket tartalmazza, annak aktuális állapotát. Ez egy teljesen hagyományos könyvtár, GIT-től független.

**Verzió napló (STAGE)** – Ebben tartjuk nyilván, hogy az adott (aktuális) verzióban milyen fájlok, milyen tartalommal szerepelnek. Ez a napló egy szöveges fájl.

Mentett verziók (**local repository**) – Itt található az összes mentett állapot (verzió), amelyek közül szükség esetén vissza is térhetünk.

### Inicializálás

A GIT telepítésével a Git BASH parancsértelmező is a gépünkre került. Ajánlott belépni a munka könyvtárba, a helyi menüből elindítani a BASH-t. Ezután ki kell adnunk a „git init” parancsot. A parancs előkészíti a könyvtárunkat a GIT használatára. Létezőn a „git reinit” könyvtár.

### Konfiguráció

Mielőtt még használni kezdenénk a GIT-et meg kell tennünk pár beállítást:

```
git config --global user.name "saját nevünk"
git config --global user.email "saját email címünk"
```

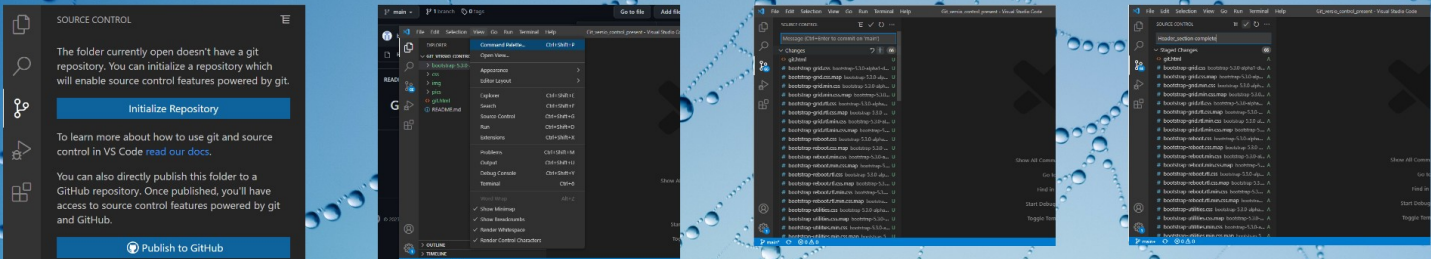
### Alap parancsok

Amikor elkészültünk egy részfeladattal, akkor a **git add** fájlnevén révén felvesszük a fájlokat a verziónaplóba. A **git commit -m "Üzenet"** fájlnevén pedig létrejön egy verzió a projektünkről.

### A Visual Studio Code és a GIT

A Visual Studio Code program némileg egyszerűbbé teszi a git használatát. A GIT eszköztára az ablak bal oldalán található **Source control** ikonra kattintva érhető el.

Eőször felkínálja az adott könyvtár inicializálását. Az add parancsot a + jellel adhatjuk ki. A commit parancs jele egy pipa, a parancshoz tartozó message helye pedig a pipa alatti bevitteli mező.



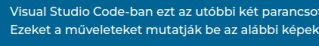
### Local és remote repository

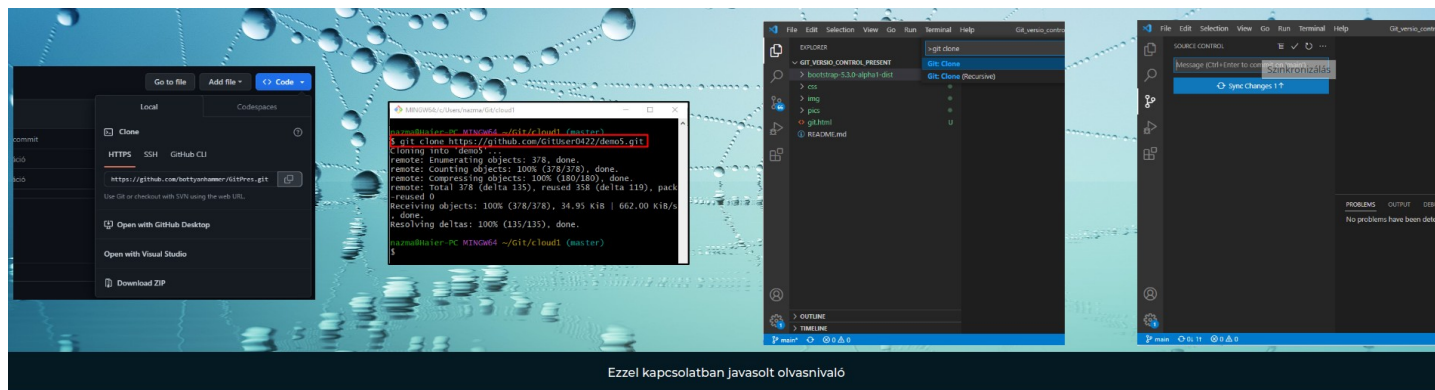
A GIT verziókezelő rendszer lehetővé teszi, hogy a helyi gépen lévő repositoryt feltöltsük egy távoli szerverre. Ezt hívjuk remote repositorynak.

Emellett ezt a távoli repositoryt képes több felhasználóval is megosztani, hogy azt közösen használják. A változtatásokat szinkronizálhatjuk, és így a GIT megteremtí a **csapatmunka** lehetőségét is.

Javasolt, hogy a helyi tárolót a távoli tároló klónozásával hozzuk létre a **git clone url-cím** paranccsal. Így minden beállításra kerül a háttérben. Ezután már egyszerűen használhatjuk a **git pull** parancsot a szinkronizáláshoz, illetve a **git push** parancsot a fájlok feltöltéséhez.

Visual Studio Code-ban ezt az utóbbi két parancsot a Sync Changes gombra kattintva valósíthatjuk meg. Ezeket a műveleteket mutatják be az alábbi képek:





A feladatok előtt kérlek, klónozd az itt található url-t a github-ról!

Url:

Ezek után hozz létre a saját gitHub-odon egy GIT\_feladat\_vezeteknevKezdoobetuj\_keresztnev nevű repositoryt!

Példa: **GIT\_feladat\_H\_Peter**

Ezt is klónozd, majd ebbe az utóbbi mappába másold az én gitHub-omról letöltött állományokat!

Ezek az állományokon dolgozz!

Ha kész vagy, akkor add, majd commit a tanult módokon, és szinkronizáld a local repositoryt a remote repositoryval!

A végén töltsd fel a megoldást zip-pelve a classroomba is, illetve ide linkeld be a Te gitHubos linkedet is, hogy ha szeretném, akkor én is klónozhassam a Te projektet. (Ez csak ellenőrzés lesz.)

**Erre szánjatok legalább 10 percet!**

Ezeket a feladatokat a git.html állományban végezd el!

1. Az oldal nyelve legyen magyar (Karakterkészlet: utf-8)!
2. A címsorban megjelenő cím: *Git verziókezelő* legyen!
3. Helyezz el hivatkozást a css/style.css állományra úgy, hogy az utasításai érvényesülhessenek!
4. A menüsorban a Parancssor menünek hiányoznak az almenüi. Készítsd el azokat a következő módon!
  1. A menüpontok rendre a következő azonosítókra hivatkozzanak: init, conf, com!
  2. Az almenüpontok a következők legyenek: Inicializálás, Konfigurálás, Alap parancsok!
5. A header-be helyezd el a *Linus-Torvald.jpg* képet! Amennyiben rávisszük a kurzort, vagy a kép nem jeleníthető meg, akkor a Linus Torvald szöveg jelenjen meg!
6. A kép alá kerüljön egy egyes címsor! Tartalma: *GIT verziókezelő rendszer*
7. A fejléc alatt 3 kis kártyának kellene lennie, de csak kettőt látsz, mert az utolsó hiányzik. A többi mintájára készítsd el a harmadikat!
  1. A kártya címe: Igen elterjedt
  2. Bekezdése:  
Számos nagy volumenű projekt használja; a legfontosabbak ezek közül: Linux-rendszermag, Samba, MediaWiki, VLC media player, Ruby on Rails és az Android platform. A Git karbantartásának felügyeletét jelenleg Junio Hamano látja el.
8. A *GIT telepítése* (2.row) kommentnél, a row osztályú keretbe helyezz el egy kártyát, a képen látható módon!
  1. A kártyához használd fel a bootstrap oldalon található kártyák közül az elsőt! A felesleges részeket töröld a kártyából!
  2. A képet helyezd el alulra, és a *card-img-top* helyett sorold a *card-img-bottom* osztályba! A kép neve: install.png
  3. Amennyiben a kép nem jelenne meg, vagy a kurzort a kép fölé visszük, a *Szövegszerkesztő kiválasztása* szöveg jelenjen meg!
  4. A kártya szövegei a forras.txt -ben találhatóak.
  5. A kártya címét formázd az *install* azonosítóval!
  6. A <https://git-scm.com> szövegre készíts hivatkozást úgy, hogy az egy új üres oldalon nyíljon meg! A hivatkozás url-címe: <https://git-scm.com>
  7. A <https://git-scm.com> szöveg legyen dőlt betűs! Használd a megfelelő htm tag-et!

8. A hivatkozás szövege legyen fekete színű! Ezt egy inline formázással érd el!
9. A kártyát helyezd egy col osztályú keretben! Ezt a keretet újabb bootstrap osztálynevek használatával formázd a következő módon:
  1. 991 px-nél nagyobb kijelző esetén a kártya a vele egy sorban található kártyával együtt 4:8 arányban foglaljon helyet a böngésző ablakában! Tehát a most készítendő kártya 4 oszlopot foglaljon el!
  2. 992 px méret alatt a kért kártya egymás alá kerüljön! Mindkét kártyánál végezz beállítást!
10. A megfelelő bootstrap osztálynév használatával érd el, hogy a kártya fölött 4 bootstrap egységnyi külső eltartás legyen!

A következő feladatokat a style.css állományban végezd el!

1. A weboldal háttere a body-bg.jpg kép legyen! Ne az univerzális szelektort használd!
  1. A kép ne mozduljon el az oldal görgetésével!
  2. Legyen kifeszítve, és ne ismétlődjön!
2. A navigáció legyen flex tulajdonságú keret!
  1. A megfelelő szelektorról gondoskodj a body szelektor alatt!
  2. Engedélyezd, hogy kisebb kijelzőméret esetén az elemek egymás alá törjenek!
  3. A háttérszíne 0, 100, 148 legyen!
3. A navigáció ikonját kerekítsd a tanult módon!
4. A navigáció listaelemei előtt ne látszódjanak a „kis korongok”!
5. A navigáció szövege legyen fehér! A szöveg mérete legyen 1.1rem!
6. A fejléc címsora legyen vastag (félkövér), a mérete pedig 3rem!
7. Linus képének legyen árnyéka!
  1. Minden irányban 2px-es legyen!
  2. A színe: 0, 53, 84
8. A sectionRow osztályú elemet igazítsd vízszintesen középre!
9. A sectionRow osztályú keret kis kártyái bal oldalukon kapjanak egy 0, 100, 148-as értékű kék szegélyt!
10. Érd el media query (töréspont) használatával, hogy a sectionRow osztályú keret elemei 1440px méretű vagy annál kisebb méretű kijelző esetén vízszintesen középre igazodjanak! Vigyázz ez egy flex osztályú konténer! Kommenteltem a helyét!