# Git Cheat Sheet

## Gitről általában

### Mi az a Git?

* **Git**: Egy verziókövető rendszer, amely lehetővé teszi a forráskód módosításainak nyomon követését és kezelését.

### Mi a verziókövetés?

* **Verziókövetés**: A forráskód módosításainak nyomon követése és kezelése, lehetővé téve a különböző verziók közötti visszaállítást és együttműködést.

### Mire jó a GitHub?

* **GitHub**: Egy online platform, amely támogatja a verziókövetést és az együttműködést, lehetővé téve a forráskód tárolását és megosztását.

### Ágak (Branches)

* **Ág**: Egy elkülönített fejlesztési vonal, amely lehetővé teszi párhuzamos fejlesztési munkák végzését anélkül, hogy a fő ágat befolyásolná.
* **Alapértelmezett ág**: Gyakran "main" vagy "master" néven ismert, ez az alapértelmezett fejlesztési vonal, amelyből más ágakat lehet létrehozni.

### Távoli és helyi repozitóriumok

* **Távoli repozitórium**: Egy központi repozitórium, amelyhez a fejlesztők csatlakozhatnak és változtatásokat küldhetnek.
* **Helyi repozitórium**: Az egyes fejlesztők saját gépén tárolt repozitórium, ahol a helyi munkát végzik.

### Commitok és Staging Area

* **Commit**: Egy mentett állapot a repozitóriumban, amely tartalmazza a forráskód aktuális állapotát és a hozzá tartozó üzenetet.
* **Staging Area**: Egy köztes terület, ahol a változtatások gyűjtésre kerülnek, mielőtt commitolnánk őket.

### Merge Konfliktusok

* **Merge konfliktus**: Akkor fordul elő, amikor két ág egyesítése során a változtatások ütköznek, és manuális beavatkozásra van szükség a probléma megoldásához.
* **Konfliktus feloldása**: A merge konfliktusokat kézzel kell feloldani, majd commitolni a megoldott változtatásokat.

### Git Ignore

* **.gitignore fájl**: Egy fájl, amely meghatározza, mely fájlokat és könyvtárakat kell figyelmen kívül hagyni a Git számára, így azok nem kerülnek verziókövetés alá.

### Szinkronizáció

* **Fetch vs. Pull**:
  + **Fetch**: Változtatások letöltése a távoli repozitóriumból anélkül, hogy azok beolvasztanánk a helyi repozitóriumba.
  + **Pull**: Változtatások letöltése és automatikus beolvasztása a helyi repozitóriumba.

### Git Flow Model

* **Git Flow**: Egy népszerű ágazási modell, amely meghatározza a fejlesztési folyamat lépéseit, beleértve a fejlesztési (develop), fő (main/master), kiadási (release), hotfix és feature ágakat.

### Verziókövetés előnyei

* **Visszakereshetőség**: Könnyen vissza lehet térni egy korábbi állapothoz, ha szükséges.
* **Együttműködés**: Több fejlesztő egyszerre dolgozhat ugyanazon a projekten anélkül, hogy ütköznének egymással.
* **Biztonság**: A változtatások mentése és követése megvédi a munkát az adatvesztéstől.

### Tippek és Trükkök

* **Rendszeres commitok**: Gyakori commitolás segít a változtatások nyomon követésében és a visszakereshetőségben.
* **Értelmes commit üzenetek**: Rövid, de leíró commit üzenetek megkönnyítik a változtatások követését és megértését.
* **Kísérleti ágak**: Új funkciók vagy javítások kifejlesztéséhez hozz létre külön ágakat, hogy ne befolyásold a fő fejlesztési vonalat.

## Fontosabb Git parancsok

### pull

* **pull**: Egy távoli repozitóriumból történő frissítés a helyi repozitóriumba.

### update

* **update**: A helyi repozitórium frissítése a legújabb változtatásokkal (megfelel a git pull parancsnak).

### commit

* **commit**: A helyi változtatások mentése a verziókövető rendszerben.

### cherry-pick

* **cherry-pick**: Egy specifikus commit alkalmazása egy másik ágon.

### push

* **push**: A helyi változtatások feltöltése egy távoli repozitóriumba.

### branch

* **branch**: Új ág létrehozása és kezelése.

### checkout

* **checkout**: Egy adott ág vagy commit munkakönyvtárba történő váltás.

### clone

* **clone**: Egy távoli repozitórium másolatának létrehozása a helyi gépen.

### init

* **init**: Egy új, üres Git repozitórium létrehozása.

### status

* **status**: A repozitórium aktuális állapotának ellenőrzése, beleértve a változtatásokat és a staging állapotot.

### add

* **add**: Fájlok hozzáadása a staging területhez a következő commithoz.

### merge

* **merge**: Két ág egyesítése, az egyik ágon végzett változtatások beolvasztása a másikba.

### log

* **log**: A commit történet megtekintése.

### diff

* **diff**: A változtatások megtekintése, amelyek még nincsenek commitolva.

### fetch

* **fetch**: Változtatások letöltése egy távoli repozitóriumból, anélkül, hogy beolvasztanánk őket a helyi repozitóriumba.

### rebase

* **rebase**: Az aktuális ág újraalapozása egy másik ágon, a commitok sorrendjének megváltoztatása nélkül.

### stash

* **stash**: A munkakönyvtár aktuális állapotának ideiglenes tárolása és az állapot visszaállítása későbbi használatra.