Git Conflict téma

Már volt szó a pull, és a fetch műveletről. Tudjuk, hogy a kettő között a különbség „csak” annyi, hogy a pull automatikusan elvégzi a merge műveletet is.

Egy tanulságos történet

Amikor egy remote szerveren lévő változásokat pull-ozom, vagy branch-eket olvasztok össze, conflict keletkezhet. Ez abban az esetben történhet, ha egy file ugyanazon sorát többen szerkesztették, azaz két különböző commit a file ugyanazon sorának módosítását tárolja.

**Például:** két fejlesztő ugyanazon a development branchen fejleszt, és az index.html 1. sorát mindketten módosítják.

Az első fejlesztő ezt írj az első sorba:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

A másikuk viszont ezt:

<!DOCTYPE html>

Mindketten elmentik a file-t. Mindketten commitolnak, majd az első fejlesztő felpusholja a változtatások, majd amikor a második fejlesztő fel akarja pusholni a saját változtatásait akkor kap egy figyelmeztetést, hogy először pullozza le a szerveren lévő változásokat.

Elvégzi a pull műveletet. Ilyenkor kapja meg a conflict részleteit. A git nem tudja eldönteni, hogy melyik verzió a helyes. Ilyenkor manuálisan kell feloldanunk a conflit-ot. Ekkor három lehetőségünk van:

* A pullozott verziót tartjuk meg
* A saját verziót tartjuk meg
* Összefésüljük a kettőt, és akár mindegyik kódsort megtarthatjuk.

Ezért is érdemes push előtt mindig pull-ozni, vagy először csak fetch-elni.

Konklúzió

Tehát a következő legyen a sorrend:

* add – commit – pull (feloldás, commitolás ha kell) – push

Vagy a másik megoldás: stash -pull - stash apply- add -commit -push

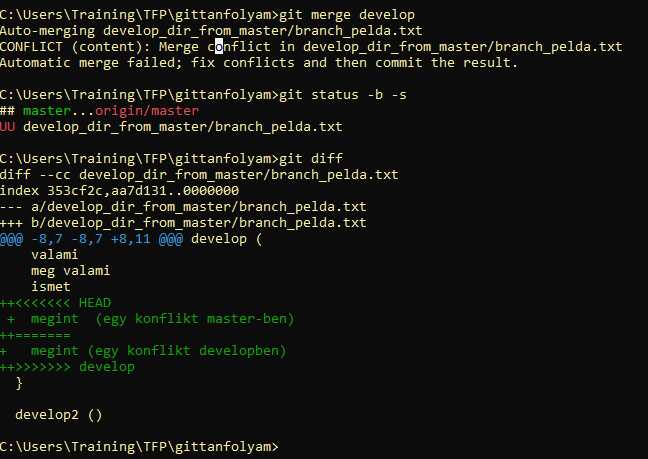
**Mindig nézzünk status-t.**

Önálló feladatok

* Idézz elő a fent leírt módon egy conflict -ot és oldd fel tetszés szerint.

**Master és develop egyazon sorban változást mutat.**

**git merge develop nem lehetséges**



Felodás: