

Git down and dirty

Bill Seremetis (@bserem)

Drupal Implementor

zehnplus.ch

Τι είναι το Git;

Note: Το Git είναι ένα σύστημα ελέγχου εκδόσεων (version control system, VCS) που αναπτύχθηκε από τον Linus Torvalds και χρησιμοποιείται ευρέως στην ανάπτυξη λογισμικού.

VCS? Τιν' τούτο;

Note: Τα VCS χρησιμεύουν στην παρακολουθήση αλλαγών σε αρχεία. Μια απλουστευμένη μορφή ενός VCS είναι το ιστορικό οποιασδήποτε σελίδας της Wikipedia (https://en.wikipedia.org/w/index.php?title=Version_control&action=history.)

Αυτόνομα προγράμματα ελέγχου εκδόσεων είναι το Git, SVN, CVS, Bazaar και άλλα.

Centralized vs Decentralized

Note: Το Git είναι ένα decentralized ή distributed VCS. Αυτό θα πει ότι μπορούν πολλοί developers να εργαστούν στο ίδιο έργο χωρίς να χρειάζεται να μοιράζονται το ίδιο δίκτυο. Η άλλη εκδοχή είναι τα centralized VCS.

The 3 states of insanity!



Όταν δουλεύουμε με το Git ζητάμε από αυτό να παρατηρεί κάποια αρχεία για αλλαγές. Τα αρχεία που παρατηρεί είναι τα `tracked files`, οτιδήποτε άλλο είναι `untracked`.

Note: Τα `tracked files` μπορούν να είναι στις εξής καταστάσεις:

- **unstaged**: δηλαδή υπάρχουν αλλαγές στο αρχείο που δεν έχουν γίνει **staged**
- **staged but not committed**: δηλαδή υπάρχουν αλλαγές που έχουν γίνει **stage**, αλλά όχι **commit**. Οι αλλαγές αυτές ξεχωρίζουν από τις **unstaged**. Ένα αρχείο μπορεί να έχει και **staged** και **unstaged** αλλαγές.
- **committed**: οι αλλαγές μας είναι αποθηκεύμενες και η παρούσα κατάσταση του αρχείου είναι αποθηκευμένη στο **Git**

Τι είναι όλα αυτά;;;

Πρέπει να αρχίσεις να σκέφτεσαι πολυδιάστατα, ευτυχώς είναι εύκολο. Όταν σε ένα αρχείο που το παρακολουθεί το Git γίνει μια αλλαγή, αυτή είναι αμέσως αναγνωρίσιμη από το Git. Μπορούμε να την αποθηκεύσουμε **προσωρινά** βάζοντας τη στο stage, ή μόνιμα κάνοντας commit.

Note: **ΠΡΟΣΟΧΗ:** ως αποθήκευση δεν αναφερόμαστε στην αποθήκευση του αρχείου από το πρόγραμμα επεξεργασίας, αλλά στην αποθήκευση των αλλαγών του αρχείου στο τοπικό repository του Git!

Το Git κάνει edit στα αρχεία;

Note: Σε γενικές γραμμές το Git δεν κάνει αλλαγές στα αρχεία, δηλαδή το Git δεν πρέπει να το μπερδέψεις με κάποιον editor. Το

Git απλά παρατηρεί τις αλλαγές στα αρχεία όπως αυτά βρίσκονται στο filesystem (δηλαδή στο σκληρό σου δίσκο). Ωστόσο, το Git μπορεί να αλλάξει τα αρχεία αυτά, από και προς καταστάσεις που ήδη αναγνωρίζει. Για παράδειγμα, μπορεί να φέρει το αρχείο σε μια παλαιότερη εκδοχή του, ή σε μια παράλληλη (περισσότερα σε λίγο) και διάφορα άλλα. Στις περιπτώσεις αυτές το Git επηρεάζει τα αρχεία στο filesystem.

Αρκετά με τη θεωρία, 'git it to me'

^προσπάθεια για catchy τίτλο, προσπεράστε τη...

Note: Για να αρχίσει το Git να κάνει τα μαγικά του για εμάς, πρέπει να του ζητήσουμε να παρακολουθεί ένα φάκελο (folder) στο σύστημα αρχείων (filesystem).

Δημιουργία ενός νέου repository

```
git init
```

Note: Φτιάξε ένα φάκελο, και μέσα σε αυτόν τρέξε: `git init`
για να φτιάξεις ένα νέο repository.

Αντιγραφή ενός υπάρχοντος repository

```
git clone /path/to/repository  
git clone username@host:/path/to/repository
```

Note: Αν θες να αντιγράψεις ένα τοπικό repository εκτελείς:

```
git clone /path/to/repository
```

Για ένα απομακρυσμένο (remote, συγκράτησε τη λέξη):

```
git clone username@host:/path/to/repository
```

Workflow

Το τοπικό repository μπορεί να έχει αρχεία στις 3 καταστάσεις που περιγράψαμε παραπάνω:

- Working Directory
- Index (stage)
- HEAD

Git Status

```
[git_down_and_dirty:master]$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

   modified:   README.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

   modified:   README.md
```

Note: Μόλις έμαθες και την δεύτερη εντολή για το git, την `git status` με την οποία το git σε ενημερώνει για οτιδήποτε συμβαίνει στο repository. Θα την πληκτρολογείς πολύ πολύ συχνά!

Προσθήκη αρχείων στο stage

```
git add <filename>  
git add *
```

Note: Σε ένα repository που μόλις κάναμε `init` δεν έχουμε αρχεία τα οποία παρακολουθεί (track) το git. Για να προσθέσουμε ένα ή πολλά δεν έχουμε παρά να κάνουμε: Κάνοντας `git add` βάζουμε και τις αλλαγές που έχουν τα αρχεία στο stage. **Αυτό είναι το πρώτο βήμα στο βασικό git workflow.**

Αποθήκευση αλλαγών στο Git

```
git commit -m "Your Commit Message"
```

Τώρα οι αλλαγές σου είναι στο **HEAD**, αλλά μόνο στο τοπικό σου repository.

Προβολή του ιστορικού

```
commit 3d73fd21a40502c0dc410c7e3398d98e7d7106c9
Author: Bill Seremetis <bill@seremetis.net>
Date:   Fri Feb 17 13:17:12 2017 +0200
```

```
    Prepare slides version of the repository.
```

```
commit 9bbb3762f5f7b8e65f713424d891be3bf682d961
Merge: 33f447b cc286e1
Author: Bill Seremetis <bill@seremetis.net>
Date:   Fri Feb 17 13:10:01 2017 +0200
```

```
    Merge branch 'master' into slides
```

Με την εντολή log:

```
git log
git log -p
git log --oneline
```

Push & Pull

```
git push origin master
```

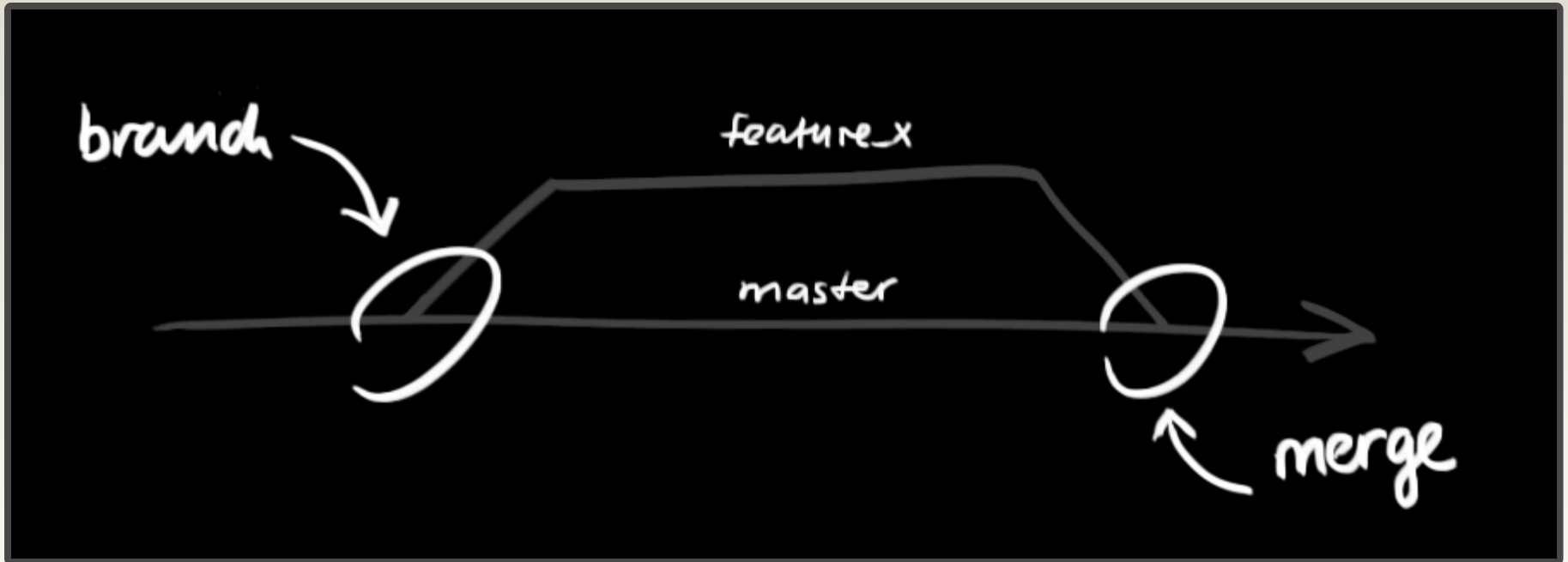
Note: Για να στείλεις τις αλλαγές σου σε κάποιο άλλο repository εκτελείς την `git push origin master` όπου `master` είναι το branch που θες να στείλεις και `origin` το repository στο οποίο θες να στείλεις τις αλλαγές σου.

Πολλαπλά remote repositories

```
git remote add <remote_name> <path_or_server>
```

Note: Το git σου επιτρέπει να εργάζεσαι με πολλά branches και remotes. Για τα branches θα μιλήσουμε σε λίγο. Για να προσθέσεις ένα remote (είτε επειδή δεν έχεις είτε επειδή θες περισσότερα) εκτελείς την: `git remote add <remote_name> <path_or_server>`.

Branches



Note: Χρησιμοποιούμε τα branches για να αναπτύξουμε χαρακτηριστικά, απομονώνοντας τις νέες αυτές προσθήκες από τον υπόλοιπο κωδικά μας. Το git ξεκινάει πάντα με ένα **master** branch.

Δημιουργία branch

```
git checkout -b feature_x
```

Note: Επιστροφή στο master: `git checkout master`

Push ενός branch

```
git push origin <branch>
```

Merging

```
git merge <branch>
```

Note: Στο branch στο οποίο θέλουμε να φέρουμε τα περιεχόμενα κάποιου άλλου, τρέχουμε την εντολή για το merge.

Ενημέρωση από repository

```
git pull
```

ΚΑΙ

```
git fetch
```

Ερωτήσεις;



Ευχαριστώ!

Βασίλης Σερεμέτης (bserem) - zehnplus.ch

Με ιδέες και εικόνες από: <http://rogerdudler.github.io/git-guide/>