

FOSSCOMM

THESSALONIKI

2024



git

Εισαγωγή στο Git | Introduction to Git

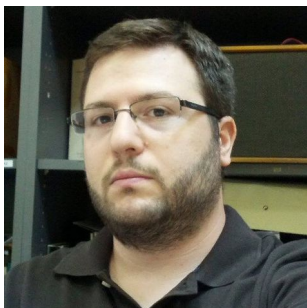


GreekLUG



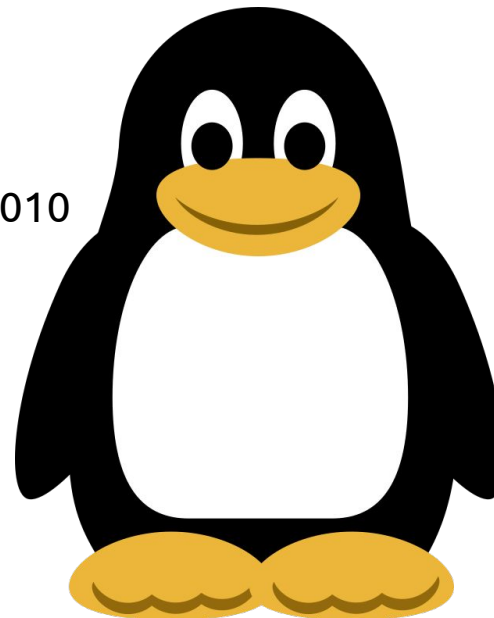
Εισαγωγή στο Git

who



~/Ζήσης Μιχάλης | System Administrator

- Μέλος της κοινότητας ΕΛ/ΛΑΚ από περίπου το 2010
- Μέλος του GreekLUG (ΔΣ)



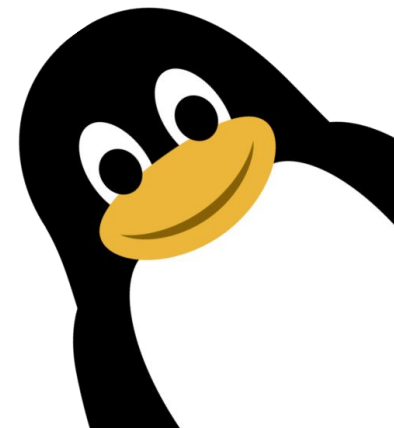


Εισαγωγή στο Git



Ενότητες Workshop

- Εισαγωγή στο versioning,
- τι είναι το Git,
- τα πρώτα βήματα,
- πως το χρησιμοποιώ,
- παραδείγματα χρήσης και
- εργασίες σε αποθετήρια.





1) Εισαγωγή στο versioning

Μίνι ιστορικό

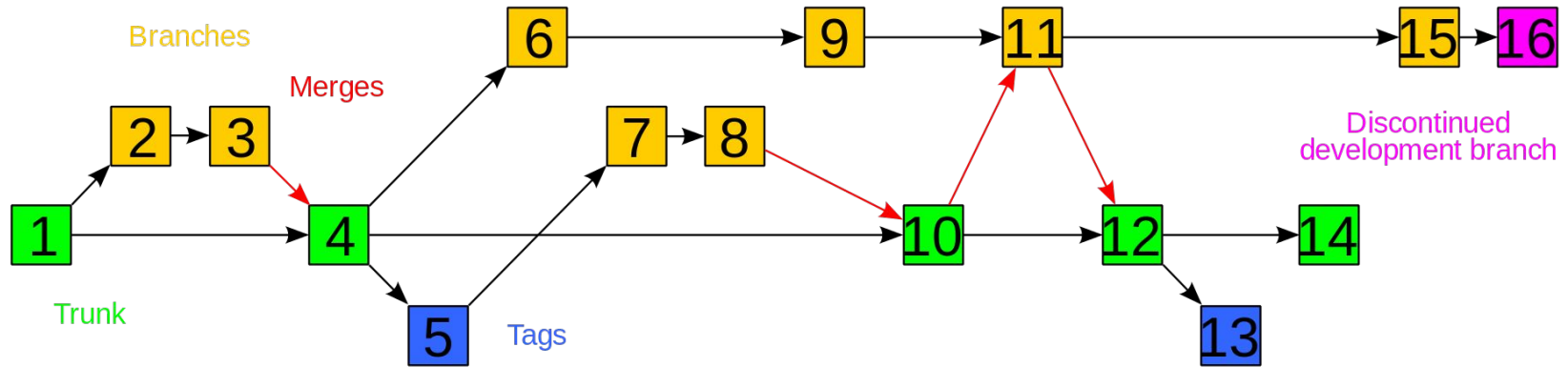
- Ο έλεγχος αναθεώρησης (**revision control**) είναι κάθε είδους πρακτική που παρακολουθεί και παρέχει έλεγχο σε αλλαγές στον **πηγαίο κώδικα**.
- Οι προγραμματιστές λογισμικού χρησιμοποιούν μερικές φορές λογισμικό ελέγχου αναθεώρησης για να διατηρούν αρχεία τεκμηρίωσης και διαμόρφωσης, καθώς και τον πηγαίο κώδικα.
- Παλαιά ιστορία... που ξεκινάει πίσω στο 1962 !
- Οι νεώτερες υλοποιήσεις είναι το **Apache Subversion (SVN)** [2000], **Mercurial** [2005] και το **Git** [2005].



1) Εισαγωγή στο versioning

Όροι

Repo | Trunk | Branches | Merge | Push | Pull | Commit | Fork



2) Τι είναι το Git

Είναι ένα δωρεάν λογισμικό **ανοικτού κώδικα** για έλεγχο εκδόσεων.

Προσφέρει

- παρακολούθηση αλλαγών σε οποιοδήποτε σύνολο αρχείων
- συντονισμό της εργασίας μεταξύ προγραμματιστών που αναπτύσσουν συνεργατικά τον πηγαίο κώδικα



Οι στόχοι του περιλαμβάνουν την ταχύτητα, την ακεραιότητα δεδομένων και υποστήριξη για κατανεμημένες, μη γραμμικές ροές εργασίας.



Ξεκίνησε το 2005 από τον **Linus Torvalds** ως εργαλείο για την ανάπτυξη του πυρήνα Linux και βασικός του συντηρητής είναι ο **Junio Hamano**.

2) Τι είναι το Git

Ιδιαίτερο χαρακτηριστικό του είναι ότι ο **κάθε κατάλογος Git είναι ένα πλήρες αποθετήριο με πλήρες ιστορικό και πλήρεις δυνατότητες παρακολούθησης εκδόσεων, ανεξάρτητα από την πρόσβαση στο δίκτυο ή έναν κεντρικό διακομιστή!**



Τελευταία σταθερή έκδοση: **2.47.0**

Αποθετήριο: **git.kernel.org/pub/scm/git/git.git**

Κώδικας: στηρίζεται κυρίως στην C, με το GUI και πρόσθετα scripts σε Shell script, Perl, Tcl και Python

Λ/Σ: POSIX (Linux, macOS, Solaris, AIX), Windows

Διανέμεται μόνο με άδεια **GPL-2.0**

Url: **<https://git-scm.com>**



3) Τα πρώτα βήματα

Εγκατάσταση Git

- GNU/Linux:

```
sudo + apt install git || dnf install git || pacman -S git || zypper install git
```

- MacOS:

εγκαθιστούμε το homebrew (<https://brew.sh>)

```
brew install git
```

- Windows:

<https://gitforwindows.org/> | κατεβάζουμε & εγκαθιστούμε το Git-X.XX.X-YY-bit.exe

περιέχει: git bash + git gui + Shell Integration

- Επαληθεύουμε με: `git --version`

3) Τα πρώτα βήματα

GNU/Linux

μέσω τερματικού

Debian/Ubuntu/Mint

- **sudo apt install git**

RHEL/Fedora/CentOS

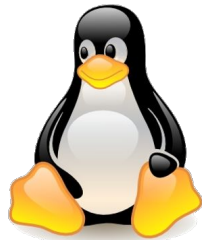
- **sudo dnf install git**

Arch Linux

- **sudo pacman -S git**

openSUSE

- **sudo zypper install git**



Εγκατάσταση Git

MacOS

εγκαθιστούμε το
homebrew

**αν δεν το έχουμε ήδη:*

- <https://brew.sh>

μέσω τερματικού:

- **brew install git**



Windows

μεταβαίνουμε στη σελίδα:
<https://gitforwindows.org>

κατεβάζουμε / εγκαθιστούμε
το **Git-X.XX.X-YY-bit.exe**

- όπου X.XX.X η έκδοση
- όπου YY η αρχιτεκτονική του Η/Υ μας
πχ Git-2.47.0-64-bit.exe



ελέγχουμε με: `git --version`

3) Τα πρώτα βήματα

Βασική ρύθμιση

- Εντολή **git config**

Τοποθεσίες

#**Local** → μέσα στο κάθε αποθετήριο | `git config --local`

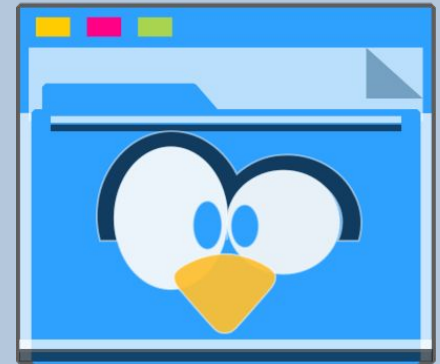
`.git/config`

#**User** → επίπεδο χρήστη | `git config --global`

`/home/user/.gitconfig`

#**System** → επίπεδο συστήματος | `git config --system`

`/etc/gitconfig`



*Από προκαθορισμένα, το `git config` θα γράψει στο local level



3) Τα πρώτα βήματα

Βασική ρύθμιση

```
git config --global user.name "GreekLUG"  
git config --global user.email "fosscomm2024@linux.edu.gr"  
git config --global core.editor "nano -w"
```

Εμφάνιση ρυθμίσεων:

```
git config --list  
git config --list -show-origin
```

Επεξεργασία αρχείου ρυθμίσεων:

```
git config -e -global
```

Βοήθεια / πρόσθετα options:

```
git config --help
```

4) Πως το χρησιμοποιώ





4) Πως το χρησιμοποιώ

Τοπικό αποθετήριο





4) Πως το χρησιμοποιώ

Δημιουργία τοπικού αποθετηρίου

```
#Δημιουργία αποθετηρίου σε έναν φάκελο
```

```
#git init <φάκελος>
```

```
πχ
```

```
git init fosscomm2024-1-local
```

```
#Πλοηγούμαστε σε αυτό και βλέπουμε τα περιεχόμενά του:
```

```
cd fosscomm2024-1-local/
```

```
ls -la
```

```
σύνολο 12
```

```
drwxrwxr-x 3 greeklug greeklug 4096 Νοε 17 19:05 .
```

```
drwxrwxr-x 3 greeklug greeklug 4096 Νοε 17 19:05 ..
```

```
drwxrwxr-x 7 greeklug greeklug 4096 Νοε 17 19:05 .git
```

```
greeklug@gnulinuxlab-pc:~/fosscomm2024/fosscomm2024-1-local$
```



4) Πως το χρησιμοποιώ

Προσθήκη περιεχομένου

#Δημιουργούμε/μεταφέρουμε τα δεδομένα μας στο αποθετήριο

πχ

δημιουργούμε το αρχείο "notes.txt" με χρήση του επεξεργαστή nano, όπου εισάγουμε την παρακάτω γραμμή ως περιεχόμενο

```
nano notes.txt
```

```
-----
```

```
# Workshop Git
```

```
-----
```



4) Πως το χρησιμοποιώ

Κατάσταση αποθετηρίου

#Ελέγχουμε την κατάσταση του αποθετηρίου. Εμφανίζεται μια λίστα με τα αρχεία που βρίσκονται #ή μη στο "Staging Area" ή δεν παρακολουθούνται για αλλαγές
git status

```
greeklug@gnulinuxlab-pc:~/fosscomm2024/fosscomm2024-1-local$ git status
```

```
Στον κλάδο master
```

```
Δεν υπάρχουν ακόμα υποβολές
```

```
Μη παρακολουθούμενα αρχεία:
```

```
(χρησιμοποιήστε "git add <αρχείο>..." για να συμπεριληφθεί σε αυτά που θα υποβληθούν)
```

```
notes.txt
```

```
δεν προστέθηκε τίποτα στην υποβολή αλλά υπάρχουν μη παρακολουθούμενα αρχεία (χρησιμοποιήστε "git add " για να παρακολουθήσετε)
```

```
greeklug@gnulinuxlab-pc:~/fosscomm2024/fosscomm2024-1-local$ █
```




4) Πως το χρησιμοποιώ

Προσθήκη δεδομένων στο Staging Area

#Προσθήκη ενός φακέλου ή αρχείου στην Staging Area

#git add <φάκελος / αρχείο>

#git add . → εισάγουμε αρχεία που εργαζόμαστε και παρακολουθούνται από το git

#git add -A → εισάγουμε όλα τα δεδομένα, ακόμη και νέα αρχεία

πχ

```
git add notes.txt
```

#Ελέγχουμε δίνοντας εκ νέου με git status

Δεν υπάρχουν ακόμα υποβολές

Αλλαγές προς υποβολή:

(χρησιμοποιήστε "git rm --cached <αρχείο>..." για να αποπροετοιμάσετε)

νέο αρχείο: notes.txt



4) Πως το χρησιμοποιώ

Καταχώριση αλλαγών

#Δημιουργία μίας καταχώρησης με τις αλλαγές μας

git commit → ανοίγει τον επεξεργαστή κειμένου για να εισάγουμε το μήνυμά μας, εναλλακτικά

git commit -m "μήνυμα" → εισάγουμε άμεσα μήνυμα

πχ

```
git commit -m "το 1ο μας commit"
```

```
[master (root-commit) 1ca1e0e] το 1ο μας commit
1 file changed, 1 insertion(+)
create mode 100644 notes.txt
```

#Επιβεβαιώνουμε δίνοντας εκ νέου git status

```
Στον κλάδο master
δεν υπάρχει τίποτα προς υποβολή, το δέντρο εργασίας είναι καθαρό
```



4) Πως το χρησιμοποιώ

Αλλαγή περιεχομένου

#Τροποποιούμε το αρχείο “notes.txt” με χρήση του επεξεργαστή nano, όπου εισάγουμε τα παρακάτω ως περιεχόμενο

```
nano notes.txt
```

```
-----
```

```
# Workshop Git
```

```
# FOSSCOMM2024
```

```
Παρουσιαστής: GreekLUG
```

```
-----
```



4) Πως το χρησιμοποιώ

Εξέταση αλλαγών

#Ελέγχουμε τις αλλαγές μας από το τελευταίο commit με χρήση

git diff

```
diff --git a/notes.txt b/notes.txt
index 9b255c4..1018173 100644
--- a/notes.txt
+++ b/notes.txt
@@ -1 +1,3 @@
 # Workshop Git
+# FOSSCOMM2024
+Παρουσιαστής: GreekLUG
```

*Με **πράσινο** χρώμα εμφανίζονται οι προσθήκες

*Με **κόκκινο** χρώμα εμφανίζονται οι αλλαγές

*Με **γκρι** χρώμα εμφανίζονται δεδομένα που παρέμειναν ίδια



4) Πως το χρησιμοποιώ

Καταχώριση αλλαγών

```
#Προσθέτουμε τις αλλαγές μας στην Staging Area
```

```
git add .
```

```
#Δημιουργία μίας νέας καταχώρησης με τις αλλαγές μας
```

```
git commit -m "το 2ο μας commit"
```

```
[master b087230] το 2ο μας commit  
1 file changed, 2 insertions(+)
```



4) Πως το χρησιμοποιώ

Εμφάνιση ιστορικού

#Βλέπουμε το ιστορικό των commit μας με χρήση

git log

```
commit b0872307f29187153d8b7cbe415d1ac3184edea6 (HEAD -> master)
```

```
Author: GreekLUG <fosscomm2024@linux.edu.gr>
```

```
Date: Sun Nov 17 19:44:08 2024 +0200
```

το 2ο μας commit

```
commit 1ca1e0e65bbdbdf4e1824725808decd253b71b3b
```

```
Author: GreekLUG <fosscomm2024@linux.edu.gr>
```

```
Date: Sun Nov 17 19:28:06 2024 +0200
```

το 1ο μας commit



4) Πως το χρησιμοποιώ

Βοηθητικά

#Συνηθίζεται να δημιουργούμε ένα αρχείο πληροφοριών "README"

#[!] Δεν πρέπει να ξεχνάμε την δημιουργία τεκμηρίωσης/σημειώσεων

πχ σε markdown → README.md

#Μπορούμε να εξαιρέσουμε δεδομένα με το αρχείο ".gitignore"

#[!] το ίδιο το αρχείο .gitignore γίνεται track από το Git

#μπορούμε να χρησιμοποιήσουμε wildcard χαρακτήρες

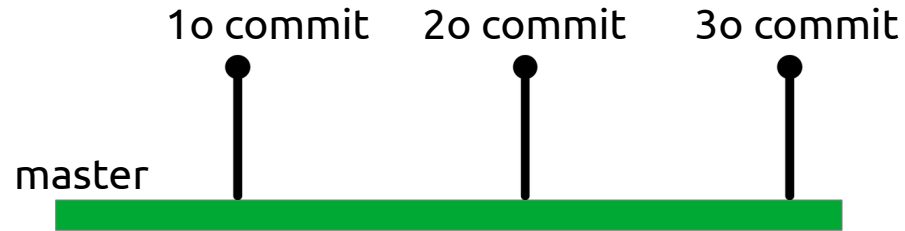
πχ για εξαιρέσουμε όλα τα αρχεία που τελειώνουν σε .tmp μπορούμε να εισάγουμε το ακόλουθο:

*.tmp



4) Πως το χρησιμοποιώ

Τοπικό αποθετήριο





4) Πως το χρησιμοποιώ

Επαναφορά σε προηγούμενη κατάσταση

#Μπορούμε να μεταβούμε στην κατάσταση του αποθετηρίου σε οποιαδήποτε καταχώριση με

#χρήση της εντολής `git checkout <αριθμός καταχώρησης commit>`

πχ

```
git checkout 1ca1e0e65bbdbdf4e1824725808decd253b71b3b
```

#Αναίρεση (`reset`) (επαναφέρει σε προηγούμενη κατάσταση χωρίς να κρατάει τις αλλαγές που

έγιναν ενδιάμεσα) → [!] ΕΠΙΚΙΝΔΥΝΟ [!]

```
git reset <αριθμός καταχώρησης commit>
```

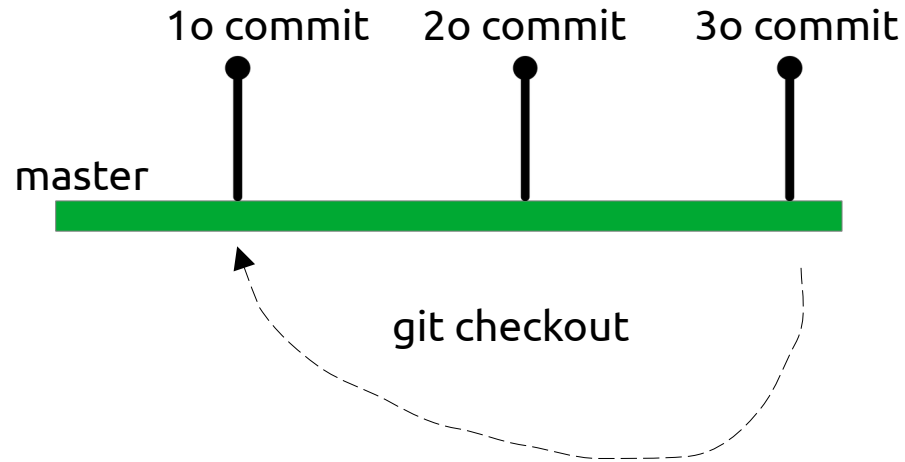
#Επαναφορά (`revert`) (διατηρεί το ιστορικό αλλά επαναφέρει σε προηγούμενη κατάσταση)

```
git revert <αριθμός καταχώρησης commit>
```



4) Πως το χρησιμοποιώ

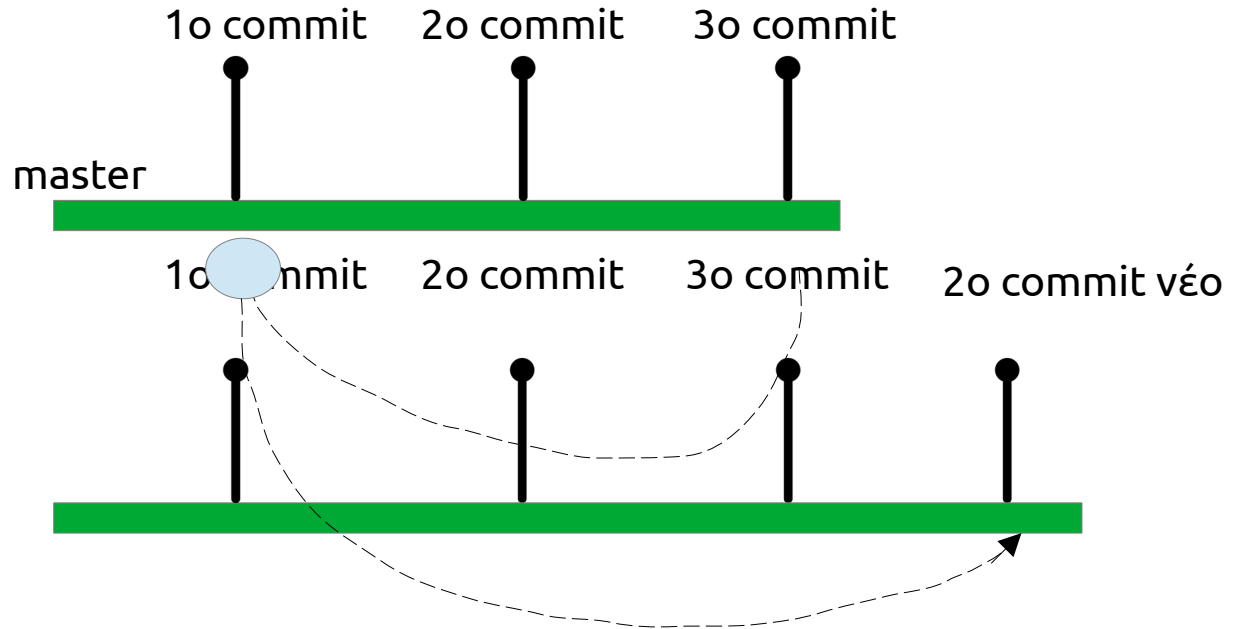
Τοπικό αποθετήριο





4) Πως το χρησιμοποιώ

Τοπικό αποθετήριο





5) Εργασίες σε αποθετήρια

Δημιουργία κλειδιού SSH

Κλειδί με ED25519:

```
ssh-keygen -t ed25519 -C "fosscomm2024@linux.edu.gr"
```

Εισάγουμε κάποιο passphrase:

```
*****
```

id_ed25519 | **ιδιωτικό** τμήμα κλειδιού

id_ed25519.pub | **δημόσιο** τμήμα κλειδιού

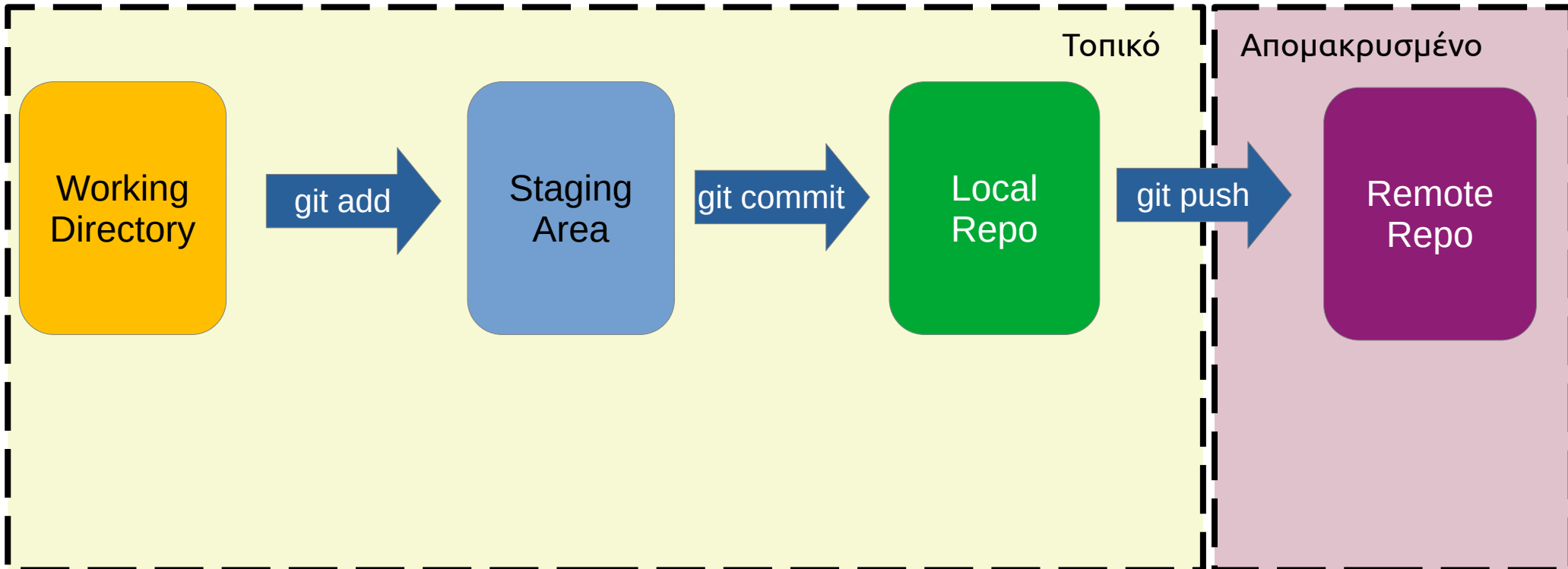


Αντιγράφουμε το (**δημόσιο** | .pub) στην πλατφόρμα/αποθετήριο, πχ GitLab



5) Εργασίες σε αποθετήρια

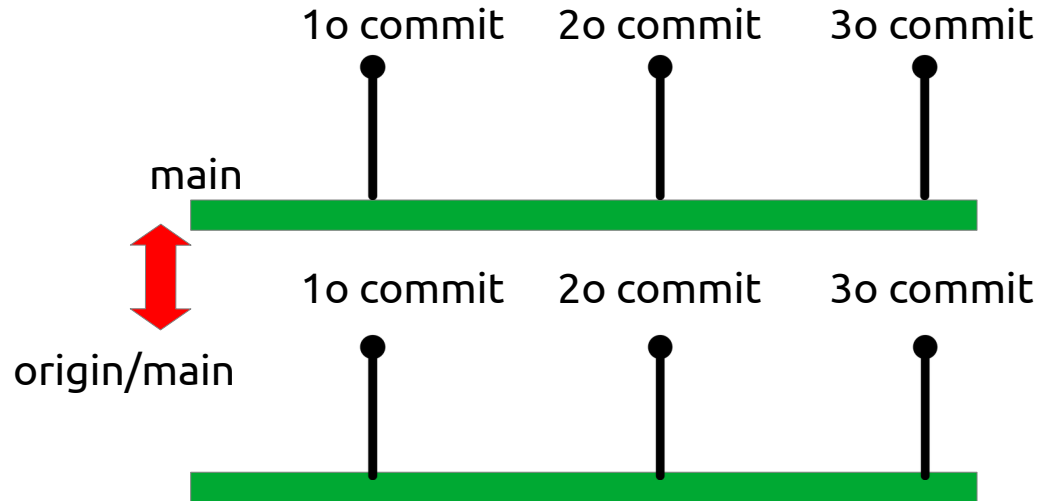
Απομακρυσμένο αποθετήριο





5) Εργασίες σε αποθετήρια

Απομακρυσμένο αποθετήριο





5) Εργασίες σε αποθετήρια

Ανάκτηση απομακρυσμένου αποθετηρίου

#Δημιουργία απομακρυσμένου αποθετηρίου σε έναν φάκελο

```
#git clone <αποθετήριο>
```

πχ επιστρέφουμε στον κεντρικό μας φάκελο και εκτελούμε

```
cd ~
```

```
git clone git@gitlab.com:fosscomm2024/fosscomm2024-2-online.git
```



5) Εργασίες σε αποθετήρια

Προσθήκη περιεχομένου

#Δημιουργούμε το αρχείο “notes-live.txt” με χρήση του επεξεργαστή nano, όπου εισάγουμε τα
#παρακάτω ως περιεχόμενο

```
nano notes-live.txt
```

```
-----
```

```
# Workshop Git
```

```
# FOSSCOMM 2024
```

```
Παρουσιαστής: GreekLUG
```

```
Σάββατο, 09/11/2024
```

```
-----
```




5) Εργασίες σε αποθετήρια

Καταχώριση αλλαγών

#Προσθέτουμε τις αλλαγές μας στην Staging Area

```
git add .
```

#Δημιουργία μίας καταχώρησης με τις αλλαγές μας

```
git commit -m "το 1ο online commit μας"
```

#Αποστολή των δεδομένων του κλάδου <main> στο απομακρυσμένο αποθετήριο <origin>

```
git push
```

```
Απαρίθμηση αντικειμένων: 4, ολοκληρώθηκε.  
Καταμέτρηση αντικειμένων: 100% (4/4), ολοκληρώθηκε.  
Delta compression using up to 4 threads  
Συμπίεση αντικειμένων: 100% (3/3), ολοκληρώθηκε.  
Writing objects: 100% (3/3), 433 bytes | 433.00 KiB/s, ολοκληρώθηκε.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
To gitlab.com:fosscomm2024/fosscomm2024-2-online.git  
85cbb2a..8aae865 main -> main
```



5) Εργασίες σε αποθετήρια

Εμφάνιση πληροφοριών απομακρυσμένου αποθετηρίου

#Μπορούμε να δούμε τα συνδεδεμένα απομακρυσμένα αποθετήρια με χρήση

```
git remote show
```

```
origin
```

#Στην περίπτωσή μας υπάρχει μόνο το <origin>. Για να δούμε τις πληροφορίες του δίνουμε

```
git remote show origin
```

```
* remote origin
```

```
URL ανάκτησης: git@gitlab.com:fosscomm2024/fosscomm2024-2-online.git
```

```
URL δημοσίευσης: git@gitlab.com:fosscomm2024/fosscomm2024-2-online.git
```

```
HEAD branch: main
```

```
Απομακρυσμένος κλάδος:
```

```
main tracked
```

```
Local branch configured for 'git pull':
```

```
main merges with remote main
```

```
Local ref configured for 'git push':
```

```
main pushes to main (πλήρως ενήμερο)
```



5) Εργασίες σε αποθετήρια

Ανάκτηση αλλαγών απομακρυσμένου αποθετηρίου

#Μπορούμε να ανακτήσουμε τα απομακρυσμένα δεδομένα με χρήση

`git fetch` → ανακτά τις πληροφορίες των αλλαγών

`git merge` → συγχωνεύει τις αλλαγές στο τοπικό αντίγραφο

#Εναλλακτικά, εκτελούμε την εντολή `pull`, που πρακτικά εκτελεί `fetch+merge`

#[!] Προσοχή, καθώς μπορεί να χάσουμε τις τοπικές αλλαγές μας

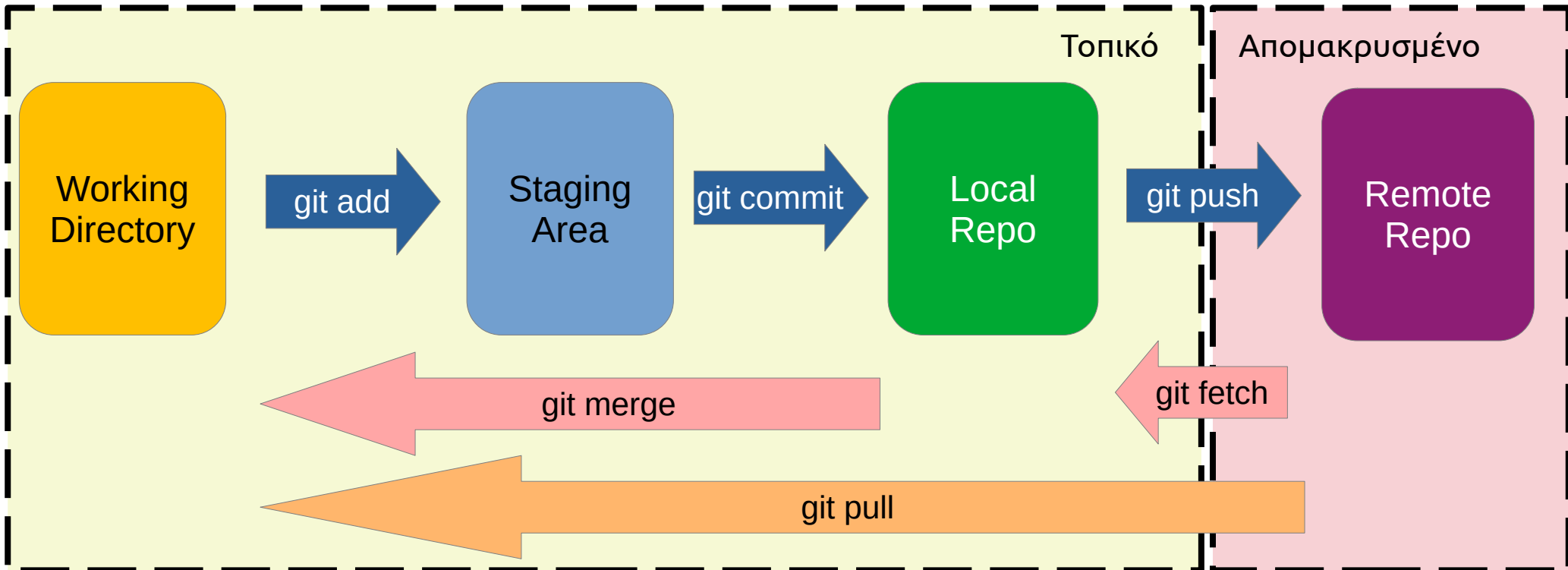
`git pull`

*Σε περίπτωση που οι αλλαγές είναι συμβατές, συνήθως εκτελείται αυτόματο `merge`, διαφορετικά λαμβάνουμε ειδοποίηση διένεξης και θα πρέπει να ελέγξουμε χειροκίνητα τις αλλαγές



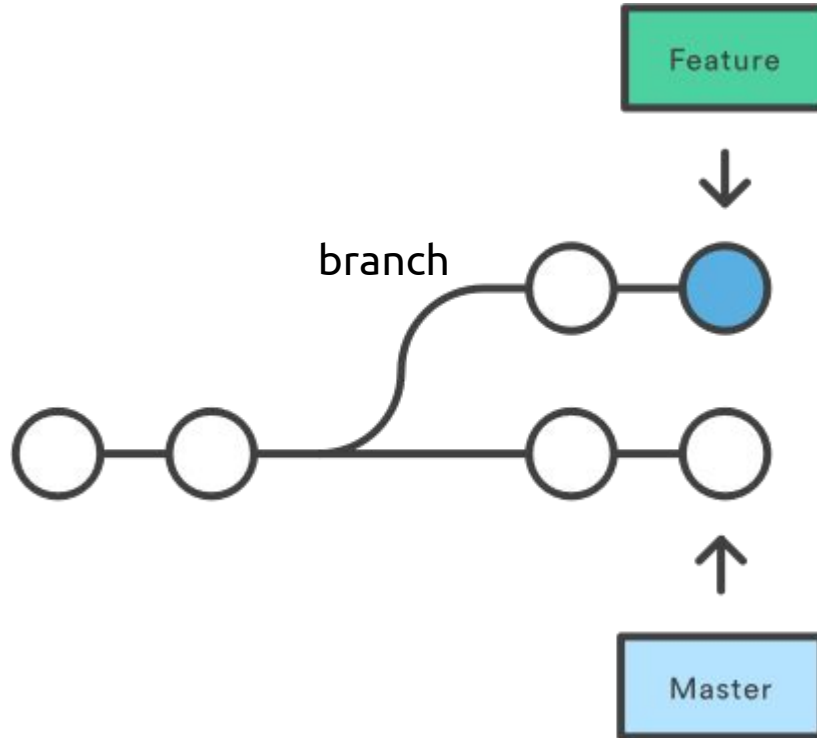
5) Εργασίες σε αποθετήρια

Απομακρυσμένο αποθετήριο





5) Εργασίες σε αποθετήρια





5) Εργασίες σε αποθετήρια

Δημιουργία Branch

#Εμφάνιση branch --> Συνήθως υπάρχει το "master" ή "main"

```
git branch --list
```

```
git branch -a
```

```
* main  
remotes/origin/HEAD -> origin/main  
remotes/origin/main
```

#Δημιουργία branch, με git branch <όνομα-branch>, πχ δημιουργούμε το branch "dev"

```
git branch dev
```

#Μετάβαση στο branch, με χρήση checkout

```
git checkout dev
```

#Απ' ευθείας Δημιουργία + Μετάβαση στο branch

```
git checkout -b dev
```



5) Εργασίες σε αποθετήρια

Εργασίες σε Branch

#Για συγχώνευση ενός branch στο κύριο, αρχικά μεταβαίνουμε σε αυτό και έπειτα συγχωνεύουμε

```
git checkout main
```

```
git merge dev
```

#Για διαγραφή τοπικού branch, χρησιμοποιούμε το `git branch -d <όνομα-branch>`

```
git branch -d dev
```

#Για διαγραφή απομακρυσμένου branch, χρησιμοποιούμε

```
#git push <απομακρυσμένο_αποθετήριο> --delete <όνομα-branch>
```

```
git push origin -d dev
```



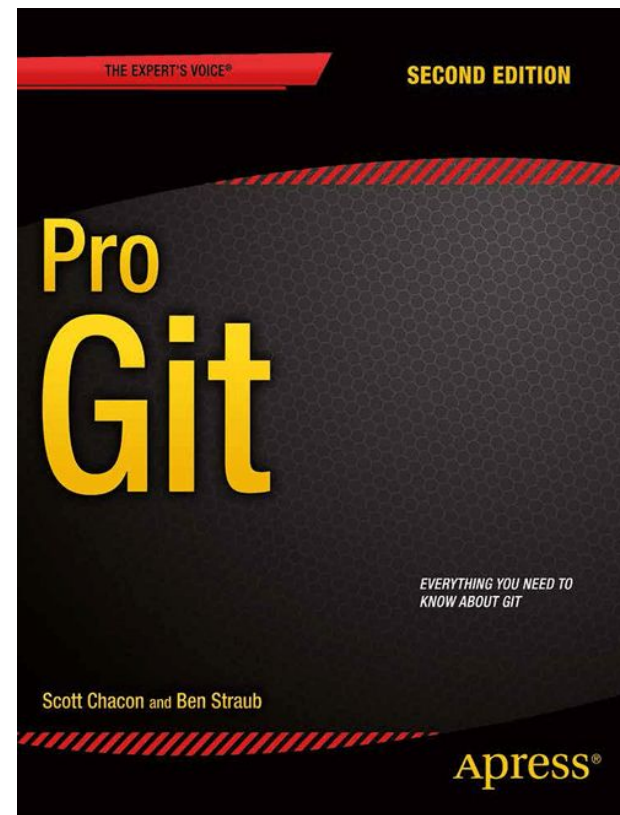
GreekLUG

Υλικό

🚩 repo.greeklug.gr

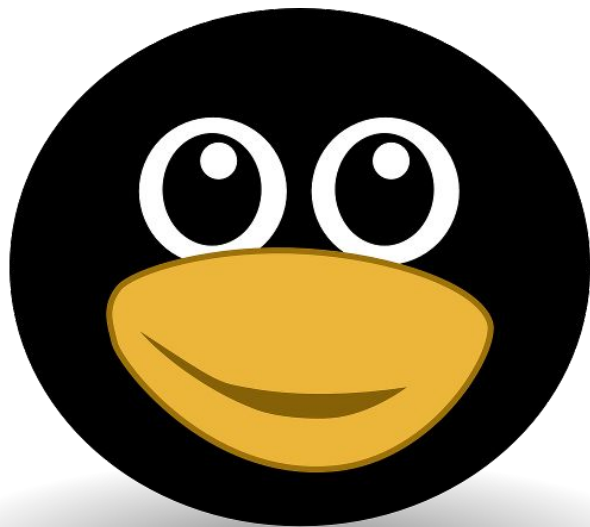
📦 Book - Git SCM

<https://git-scm.com/book/en/v2>





GreekLUG



Ευχαριστούμε!



Το αρχείο της παρουσίασης από την
Ελληνική Ένωση Φίλων ΕΛ/ΛΑΚ (GreeklUG) διέπεται από την άδεια

Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 Διεθνές
(CC BY-NC-SA 4.0)

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.el>

Ελληνική Ένωση Φίλων Ελεύθερου Λογισμικού | GreeklUG

<https://www.greeklug.gr/>