Σεμινάριο Git & GitHub

Θέμης Παπαμελετίου

Διονύσης Ζήνδρος

ΕΜΠ 2015

Ποιοι είμαστε

- Θέμης Παπαμελετίου
  - ΗΜΜΥ ΕΜΠ
  - reEmbed
  - Ex-Googler
- Διονύσης Ζήνδρος
  - ΗΜΜΥ ΕΜΠ
  - Ex-{Googler,Twitter,deviantART}

Βοηθοί & Συντελεστές

- Κωνσταντίνος Σαμαράς-Τσακίρης
  - ΗΜΜΥ ΑΠΘ, συντήρηση git-class.gr
- Δημήτρης Λαμπρινός
  - Πληροφορική ΑΠΘ, βοηθός
- Βιτάλης Σαλής
  - ΗΜΜΥ ΑΠΘ, βοηθός
- Νικόλας Κορασίδης
  - ΗΜΜΥ ΕΜΠ, βιντεοσκόπηση & ηχοληψία
- The Cube Athens

Τι θα μάθουμε

- Σήμερα
  - Τι είναι το git
  - Βασική χρήση git
  - Δουλεύοντας τοπικά με git
  - Δουλεύοντας απομακρυσμένα με git
  - Συνεργασία μέσω git{,hub}

Τι θα μάθουμε

- Τετάρτη
  - Προχωρημένες τεχνικές git
    - Undo
    - Blame
    - Tag
    - Cherry pick
    - Rebase
  - Workflows και συνεργατικές τεχνικές

Το πρόβλημα

- Ως προγραμματιστές έχουμε ανάγκες
  - Νέος κώδικας μερικές φορές είναι buggy
  - Δουλεύουμε πολλοί ταυτόχρονα στον ίδιο κώδικα
  - Διαγράφουμε κώδικα που μπορεί να χρειαστεί ξανά
  - Χρειαζόμαστε back-ups για τη δουλειά μας

Πώς λύνουμε αυτά τα προβλήματα;

- Πώς κρατάμε πολλές εκδόσεις ενός αρχείου;
- Πώς επιστρέφουμε σε μία παλιά έκδοση;

EVERY DESIGNER IN THIS WORLD

Version control

- Μπορούμε να...
  - κρατάμε εκδόσεις στα αρχεία
  - κάνουμε undo αλλαγές
  - συνεργαζόμαστε με άλλους
  - κρατάμε backups των αρχείων μας
  - μοιραζόμαστε εύκολα τον κώδικα με την ομάδα
  - ξέρουμε ποια είναι η «τελευταία» έκδοση

Ιστορία του version control

- CVS – 1990
  - Από τα πρώτα πλήρη version control systems
- Subversion (SVN) – 2000
  - Διορθωμένο CVS για project-wide management
- git – 2005
  - Distributed version control system
- GitHub – 2008
  - Συνεργατικό περιβάλλον version control

Τι είναι το git?

- Πρόγραμμα που τρέχεις στον υπολογιστή σου
- Εργαλείο στο command line
- Το αφήνεις να χειριστεί τον κώδικά σου

Εγκατάσταση του git

- Linux (Debian, Ubuntu)
  - `apt-get install git`
- Mac
  - Τρέξε git και ακολούθα οδηγίες εγκατάστασης
- Windows
  - format c:

Εγκατάσταση του git

- Linux (Debian, Ubuntu)
  - `apt-get install git`
- Mac
  - Τρέξε git και ακολούθα οδηγίες εγκατάστασης
- Windows
  - Κατέβασμα από το https://git-scm.com/download

Βιβλιογραφία

- git-οβιβλίο: https://git-scm.com/book
- https://try.github.io/
- https://pcottle.github.io/learnGitBranching/

Ρύθμιση του git

- Πρέπει να πεις στο git ποιος είσαι
  - Το όνομα θα φαίνεται στον δημόσιο κώδικά σου
- `git config --global`
  - Αλλάζει τις ρυθμίσεις του git

```
dionyziz@erdos ~ % git config --global user.name "Dionysis Zindros"
dionyziz@erdos ~ % git config --global user.email "dionyziz@gmail.com"
```

Merge pull request #118 from pkakelas/readme ···

pkakelas authored 15 days ago

11c44ae

Start using winston's 'warn' method instead of undefined 'warning'

VitSalis authored 15 days ago

528969e

Make Django use sqlite3 instead of mysql5.5 for testing

VitSalis authored 15 days ago

4e6810c

Commits on Sep 16, 2015

Enable Travis tests for django

VitSalis authored 18 days ago

ce9c6e0

Commits on Sep 12, 2015

Merge pull request #122 from mbalamat/serverside_username_validation ···

mbalamat authored 19 days ago

9f97082

Add serverside username validation

mbalamat authored 23 days ago

bad8240

Commits on Sep 10, 2015

Make ting set-up more descriptive in README

dionyziz authored 24 days ago

7b4e2b2

Ας ξεκινήσουμε ένα project!

- Ανοίξτε όλοι τα laptops σας

git init

- Άνοιξε τη γραμμή εντολών
- Δημιούργησε ένα φάκελο
- Μέσα στο φάκελο τρέξε:
- git init
  - Αναθέτει στο git τη διαχείριση του κώδικά σου

```
dionyziz@erdos ~/workspace/git-class-example % git init
Initialized empty Git repository in /Users/dionyziz/workspace/git-class-example/.git/
dionyziz@erdos ~/workspace/git-class-example (master) %
```

Ο φάκελος .git

- Υπάρχει μέσα στο project μας
- Το project που ελέγχεται από το git λέγεται “repository” ή “repo”
- Μέσα αποθηκεύει μετα-δεδομένα σχετικά με το repository μας

git status

- Δείχνει την κατάσταση του κόσμου
- Τρέξτο αν δεν ξέρεις σε τι κατάσταση είναι το repository σου

```
dionyziz@erdos ~/workspace/git-class-example (master) % git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

git add

1. Ζητάει από το git να παρακολουθεί ένα νέο αρχείο
   - Πρέπει να ακολουθείται από ‘commit’
   - Παίρνει ως παράμετρο το νέο αρχείο
   - ή ένα γενικό μοτίβο

     . όλα τα αρχεία

     '*.txt' όλα τα αρχεία .txt

```
dionyziz@erdos ~/workspace/git-class-example (master) % vim README
dionyziz@erdos ~/workspace/git-class-example (master*) % cat README
Hello, git world!
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README

nothing added to commit but untracked files present (use "git add" to track)
dionyziz@erdos ~/workspace/git-class-example (master*) % git add README
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README
```

git commit

- Δημιουργεί ένα “commit”

Τι είναι ένα «commit αντικείμενο»?

- Ένα στιγμιότυπο του κόσμου
  - Το σύνολο όλων των αρχείων και φακέλων του project μας
  - Με τα περιεχόμενά τους σε μία στιγμή του χρόνου
- Περιλαμβάνει ένα περιγραφικό μήνυμα
- Καταγράφει μετα-δεδομένα
  - Ημερομηνία
  - Δημιουργό
- Έχει ένα μοναδικό αναγνωριστικό
  - π.χ. 5e0dc079899ef4b13f9fa78a53952310f94

```
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README

dionyziz@erdos ~/workspace/git-class-example (master*) % git commit -m 'Added README file'
[master (root-commit) da0e7b3] Added README file
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

git add

1. Ζητάει από το git να παρακολουθεί ένα νέο αρχείο
   - Πρέπει να ακολουθείται από ‘commit’
2. Προετοιμάζει τις αλλαγές ενός αρχείου να γίνουν commit
   - Πρέπει να ακολουθείται από ‘commit’

```
dionyziz@erdos ~/workspace/git-class-example (master) % vim README
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master
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

no changes added to commit (use "git add" and/or "git commit -a")
dionyziz@erdos ~/workspace/git-class-example (master*) % git add README
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README

dionyziz@erdos ~/workspace/git-class-example (master*) % git commit -m 'Modified README file'
[master 5e0dc07] Modified README file
 1 file changed, 1 insertion(+), 1 deletion(-)
```

git help

- Βοήθεια σχετικά με κάποιο git command
- `git help add`
- `git help commit`

Βασικό git workflow

- `vim README`
- `git add README`
- `git status`
- `git commit -m "Changed README"`

`git commit -a`

`git commit -a`

σημαίνει:

`git add + git commit`

- Δεν σας το προτείνουμε
- Τρέξτε add ξεχωριστά για να έχετε καλύτερο έλεγχο του τι γίνεται commit

.gitignore

- Αρχείο στον κεντρικό φάκελο του repo
- Μέσα γράφεις μία λίστα αρχείων
  - Ένα αρχείο ανά γραμμή
- Τέτοιου είδους αρχεία αγνοούνται από το git
  - Δεν προστίθεται με `git add .`
  - Δεν φαίνονται στο `git status`
- Μπορεί να περιέχει μοτίβα αρχείων
  - `*.swp`

Παράδειγμα .gitignore

```
node_modules/
bower_components/
*.pyc
*.swp
*.log
config/local.json
nohup.out
client/dist/
API/venv/
```

`git rm`

- Διαγράφει ένα αρχείο και ενημερώνει το git
  - Πρέπει να ακολουθείται από commit
- Κατά μία έννοια το «αντίθετο» του add

```
dionyziz@erdos ~/workspace/git-class-example (master) % git status
On branch master
nothing to commit, working directory clean
dionyziz@erdos ~/workspace/git-class-example (master) % git rm README
rm 'README'
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    README

dionyziz@erdos ~/workspace/git-class-example (master*) % git commit -m 'Removed file README'
[master 20b826e] Removed file README
 1 file changed, 1 deletion(-)
 delete mode 100644 README
```

`git mv`

- Μεταφέρει ένα αρχείο και ενημερώνει το git
  - Πρέπει να ακολουθείται από commit
- `git mv` = `git rm` το παλιό + `git add` το νέο

Ιστορικό

- Ο κώδικας αποτελείται από μία σειρά από commits
- Τα commits βρίσκονται σε χρονολογική σειρά

git log

- Δείχνει το ιστορικό
- Με το log βλέπουμε:
  - Τη λίστα του ιστορικού με όλα τα commits
  - Ποιος έκανε το κάθε commit
  - Περιγραφή
  - Αντίστροφη χρονολογική σειρά

git log

Αναγνωριστικό του commit

```
commit 5e0dc079899ef4b13f9fa78a53952310f943e34b
Author: Dionysis Zindros <dionyziz@gmail.com>
Commit: Dionysis Zindros <dionyziz@gmail.com>

    Modified README file

commit da0e7b303c53accd7d4fefb8e033210b5fda028e
Author: Dionysis Zindros <dionyziz@gmail.com>
Commit: Dionysis Zindros <dionyziz@gmail.com>

    Added README file
```

git log

Δεύτερο commit

```
commit 5e0dc079899ef4b13f9fa78a53952310f943e34b
Author: Dionysis Zindros <dionyziz@gmail.com>
Commit: Dionysis Zindros <dionyziz@gmail.com>

    Modified README file
```

```
commit da0e7b303c53accd7d4fefb8e033210b5fda028e
Author: Dionysis Zindros <dionyziz@gmail.com>
Commit: Dionysis Zindros <dionyziz@gmail.com>

    Added README file
```

Πρώτο commit

Ένα καλύτερο `git log: git lg`

Το git είναι παραμετροποιήσιμο. Μπορούμε να φτιάξουμε τις δικές μας εντολές.

π.χ. προτείνουμε την εντολή `git lg`

```
git config --global alias.lg "log --
color --graph --pretty=format:'%Cred%h
%Creset -%C(yellow)%d%Creset %s
%Cgreen(%cr) %C(bold blue)<%an>
%Creset' --abbrev-commit"
```

git lg

```
* 20b826e - (4 minutes ago) Removed file README - Dionysis Zindros (HEAD -> master)
* 8436b9f - (4 minutes ago) Add 3 lols - Dionysis Zindros
* 6013540 - (4 minutes ago) Added file test - Dionysis Zindros
* 5e0dc07 - (15 hours ago) Modified README file - Dionysis Zindros
* da0e7b3 - (15 hours ago) Added README file - Dionysis Zindros
```

git show

- Δείχνει τι έκανε ένα συγκεκριμένο commit
- Τρέχει με παράμετρο ένα αναγνωριστικό
- Με το show βλέπουμε:
  - Όλα όσα βλέπαμε με το git log
  - Όλα όσα άλλαξε ένα συγκεκριμένο commit (diff)

git show

```
commit 5e0dc079899ef4b13f9fa78a53952310f943e34b
Author: Dionysis Zindros <dionyziz@gmail.com>
Commit: Dionysis Zindros <dionyziz@gmail.com>

    Modified README file

diff --git a/README b/README
index cec196b..77060e5 100644
--- a/README
+++ b/README
@@ -1 +1 @@
-Hello, git world!
+Hello, GitHub world!
(END)
```

Αναγνωριστικά των commits

- Hash του περιεχομένου και των μεταδεδομένων του commit
- Μοναδικό για κάθε commit
  - Διαφορετικό για κάθε commit ακόμη και μεταξύ διαφορετικών repositories!
- π.χ. 8436b9f2457b55b1c81edf50d03ff48283
- Μπορούμε να αναφερθούμε και με ένα πρόθεμα (τουλάχιστον 4 χαρακτήρες)
  - 8436b
  - Αρκεί να είναι μοναδικό μέσα στο repo

`git diff`

- Δείχνει τι άλλαξε στο φάκελό μας για το οποίο το git δεν έχει ενημερωθεί ακόμα
  - Μας λέει τι θα γίνει add αν τρέξουμε git add
- Μπορούμε να το τρέξουμε μόνο του
- Ή να του δώσουμε παράμετρο συγκεκριμένα αρχεία που μας ενδιαφέρουν
- Το output μοιάζει με το `git show`

`git diff --staged`

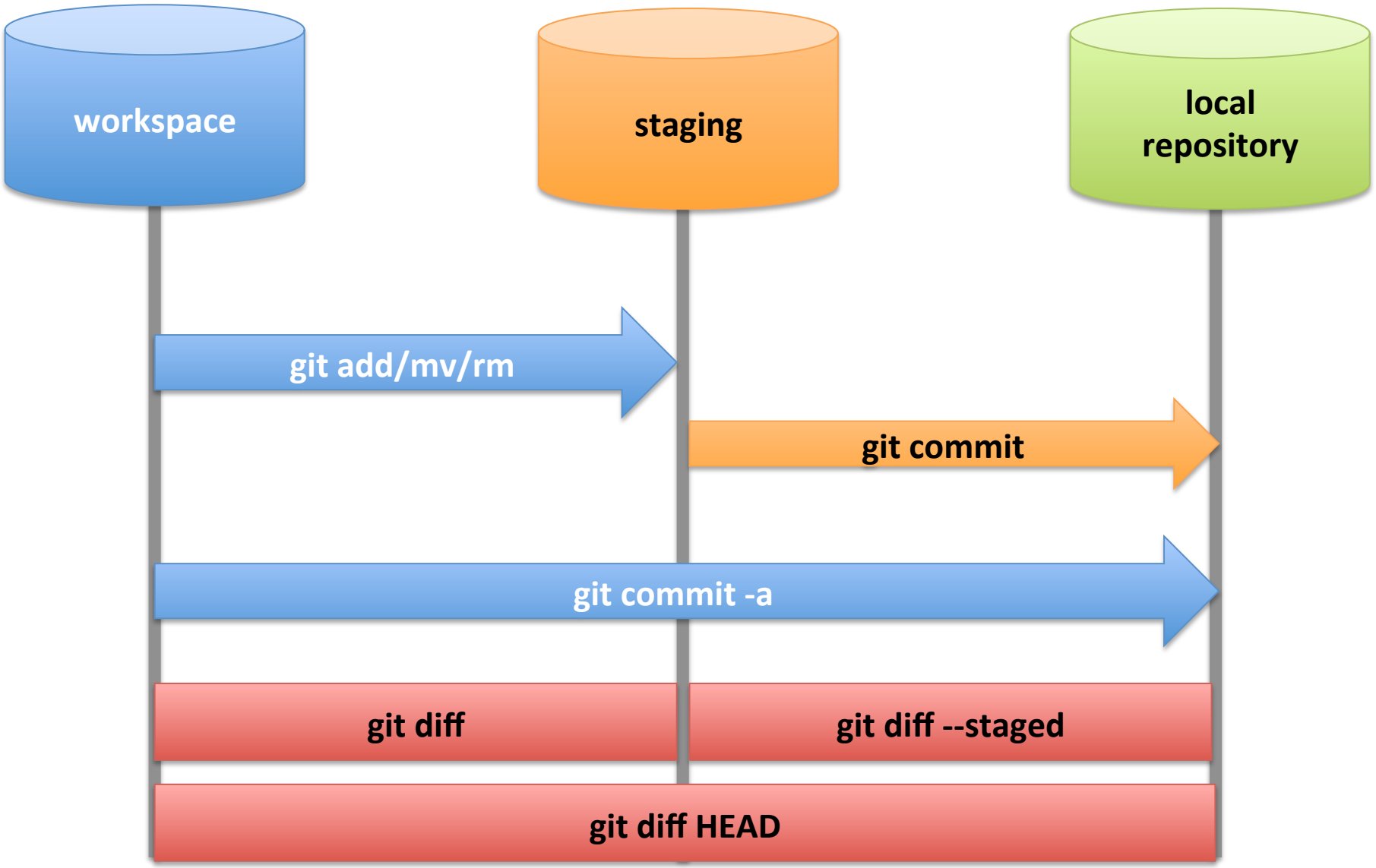
- Δείχνει τι αλλαγές έχουν γίνει που θα καταγραφούν στο επόμενο commit
  - Μας λέει τι έχει γίνει ήδη git add

`git diff HEAD`

- Δείχνει τι αλλαγές έχουν γίνει στο σύστημα από το τελευταίο commit

Staging area

- Ο εικονικός «χώρος» στον οποίο μπαίνουν οι αλλαγές μας όταν κάνουμε git add
- Μας επιτρέπει να προετοιμάσουμε ένα commit


workspace
staging
local repository
git add/mv/rm
git commit
git commit -a
git diff
git diff --staged
git diff HEAD

Quiz: Τι δείχνει αυτό το git status?

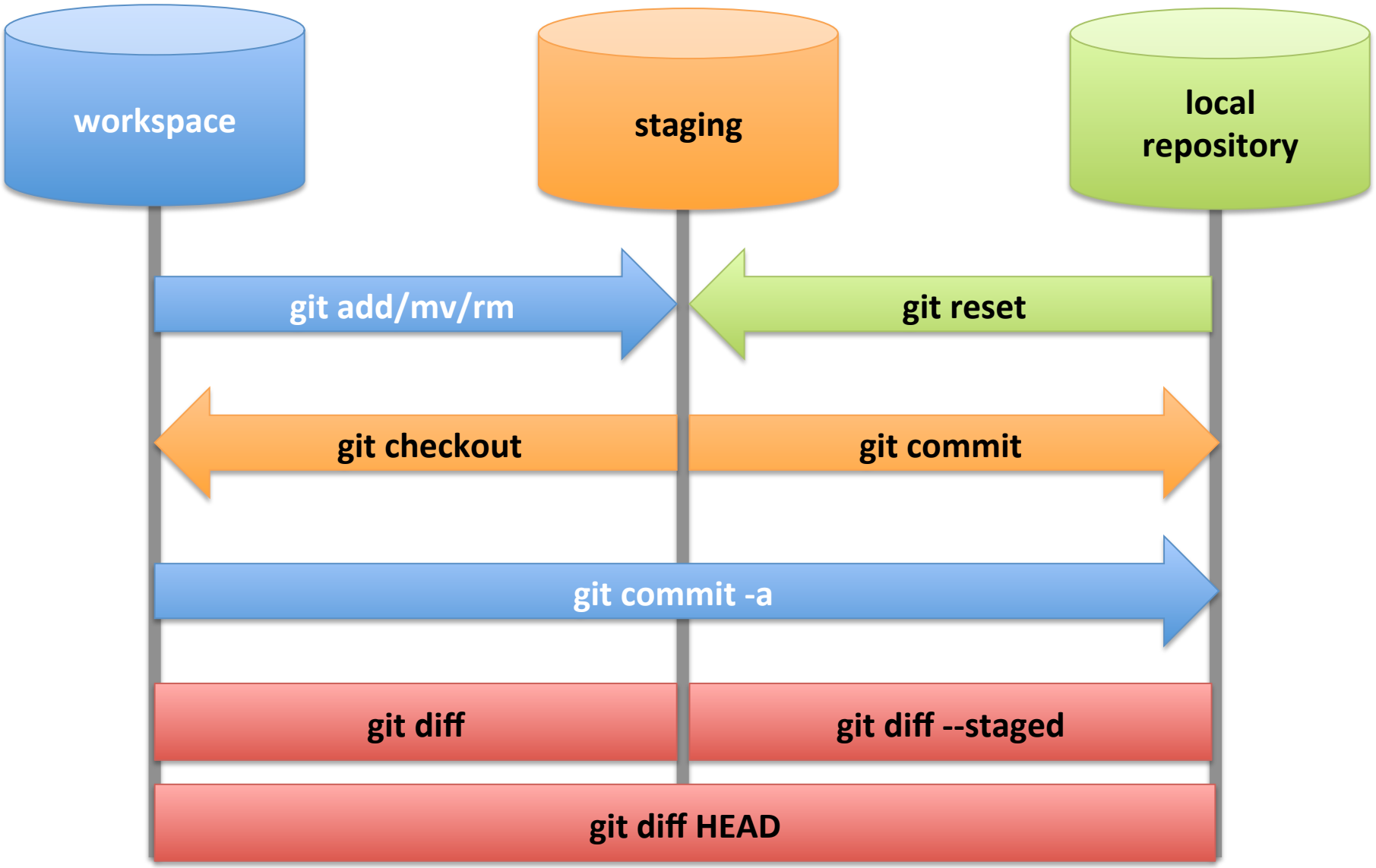
```
dionyziz@erdos ~/workspace/git-class-example (master*) % git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   test

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
```

git checkout

- Ακυρώνει τις αλλαγές μας στον κώδικα
  - που δεν έχουν γίνει stage
  - αντιγράφοντας την κατάσταση του staging area (συνήθως ίδια με το τελευταίο commit) στο working copy

`git reset`

- Αφαιρεί πράγματα που έχουν μπει στο staging area
  - αντιγράφοντάς τα από το πιο πρόσφατο commit
- Αναιρεί μία ενέργεια `git add`


workspace
staging
local repository
git add/mv/rm
git reset
git checkout
git commit
git commit -a
git diff
git diff --staged
git diff HEAD

Διάλλειμα

- φτιάξτε GitHub accounts
- γνωριστείτε μεταξύ σας!

Branches

- Η βασικότερη λειτουργία του git
- Επιτρέπει να διατηρούμε διαφορετικές εκδόσεις του κώδικά μας
- Κάθε branch
  - Έχει ένα **όνομα**
  - **Δείχνει σε ένα commit**
- Περιέχει διαφορετικό ιστορικό, με διαφορετικά commits
- Ενδεχομένως κάποια commits να είναι κοινά ανάμεσα σε branches

git branch

`git branch <name>`

- Δημιουργεί ένα νέο branch
- Του δίνουμε ως παράμετρο το όνομα του νέου branch που θέλουμε
- Το νέο branch είναι πανομοιότυπο με το υπάρχον τρέχον (δείχνει στο ίδιο commit)

`git branch`

- Δείχνει τι branches υπάρχουν
- Το τρέχον branch σημειώνεται με *

`git checkout <branch>`

- Αλλάζει το τρέχον branch
- Το τρέχον branch στον κόσμο του git αναφέρεται με το όνομα “HEAD”
- Το git checkout θέτει το HEAD στο branch που δίνεται ως παράμετρος

`git branch -d <branch>`

- Διαγράφει το branch που περνάς ως παράμετρο
- Τα commits δεν διαγράφονται, μόνο το branch που δείχνει σε αυτά

master

- Το προεπιλεγμένο branch όταν δημιουργούμε ένα νέο repository (με `git init`)
- Στα περισσότερα projects, το branch που περιέχει τον ‘τρέχοντα’ κώδικα
- Συνήθως φτιάχνουμε νέο branch ως αντίγραφο του master

```
dionyziz@erdos ~/workspace/git-class-example (master) % git branch
* master
dionyziz@erdos ~/workspace/git-class-example (master) % git branch readme_typos
dionyziz@erdos ~/workspace/git-class-example (master) % git branch
* master
  readme_typos
dionyziz@erdos ~/workspace/git-class-example (master) % git checkout readme_typos
Switched to branch 'readme_typos'
dionyziz@erdos ~/workspace/git-class-example (readme_typos) % git branch
  master
* readme_typos
```

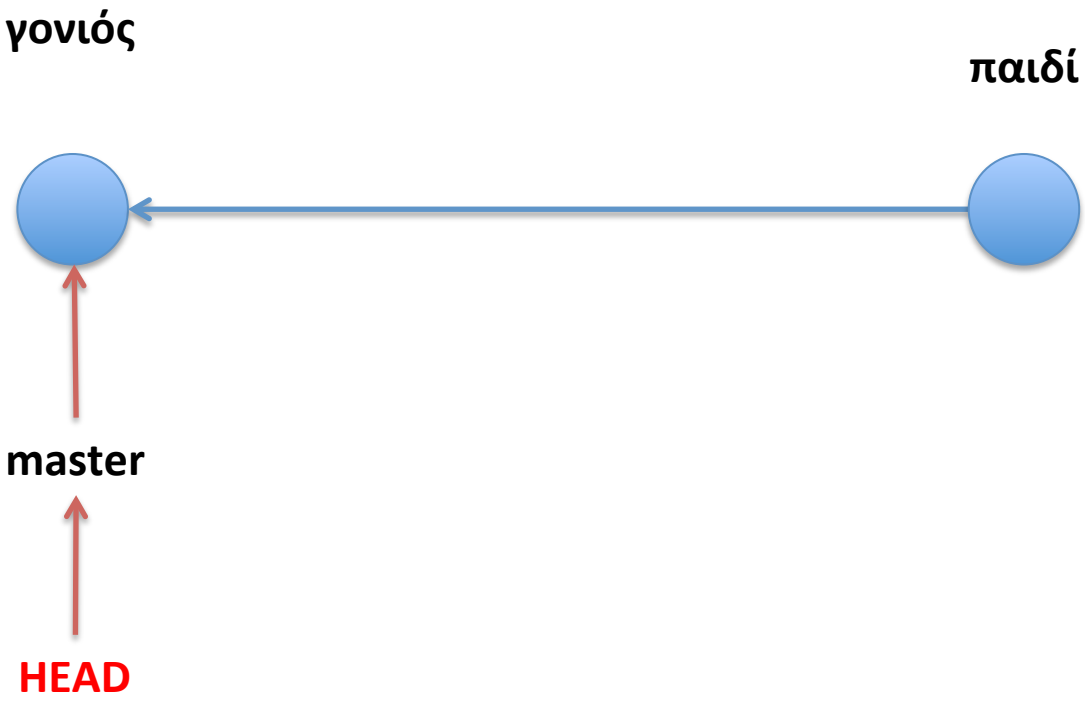
git checkout -b <branch>

- Δημιουργεί νέο branch και αλλάζει το τρέχον branch σε <branch>
- git checkout -b <branch> σημαίνει:
  - git branch <branch>
  - git checkout <branch>

Ο γράφος του git

- Το git είναι ένα **σύστημα επεξεργασίας γράφων**
- **Κάθε commit είναι ένας κόμβος**
- Κάθε commit έχει **γονιό** το προηγούμενο commit του
- Ένα branch δείχνει σε ένα commit
- Το HEAD δείχνει στο τρέχον branch
- Ένα branch επιτρέπει τη δημιουργία **διακλαδώσεων** στο γράφο

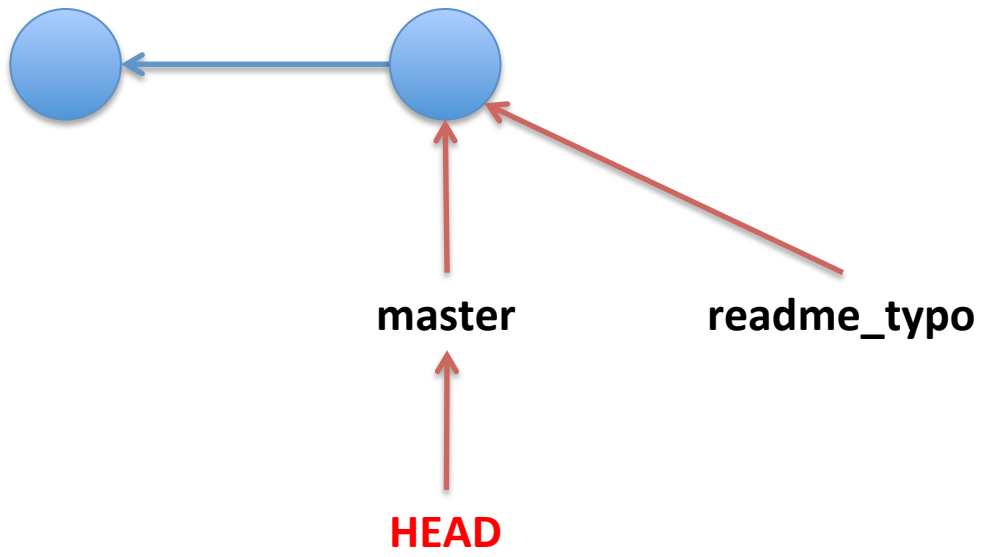
git commit



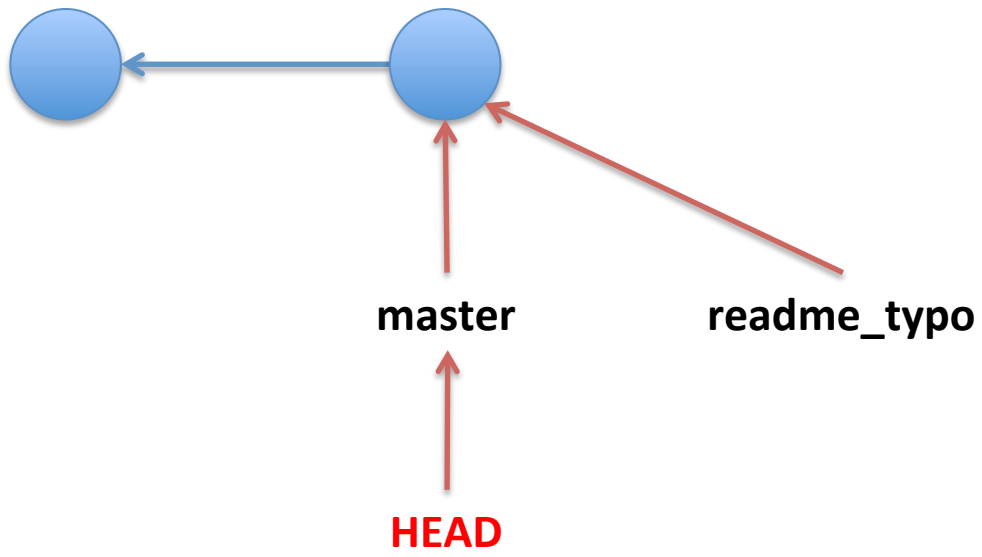
git commit

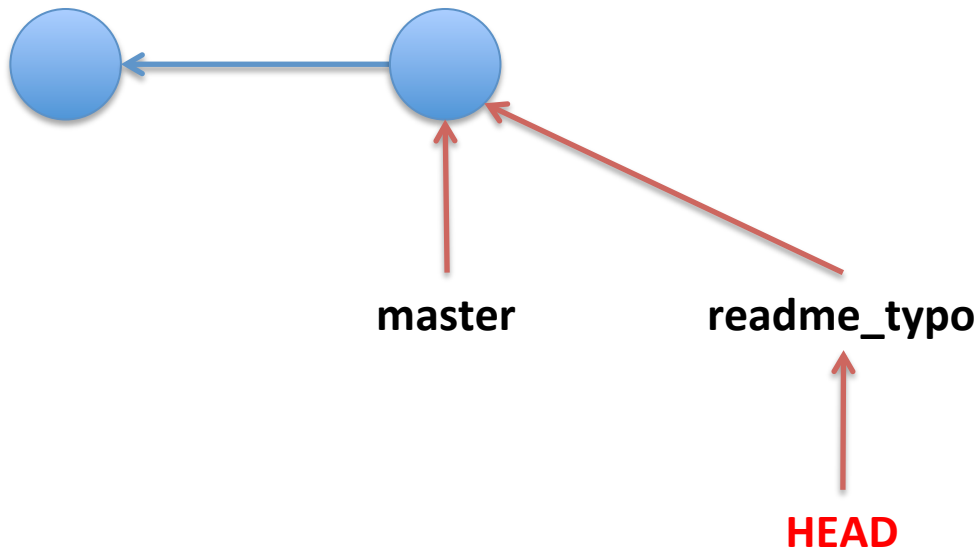
- Τώρα μπορούμε να μιλήσουμε για commits με όρους γράφων
- git commit
  - Δημιουργεί ένα νέο commit αντικείμενο
  - Ορίζει τον **γονιό** του να είναι το commit αντικείμενο στο οποίο δείχνει το τρέχον branch (δηλαδή το branch στο οποίο δείχνει HEAD)
  - Μεταφέρει το τρέχον branch στο νέο commit

git branch readme_typo

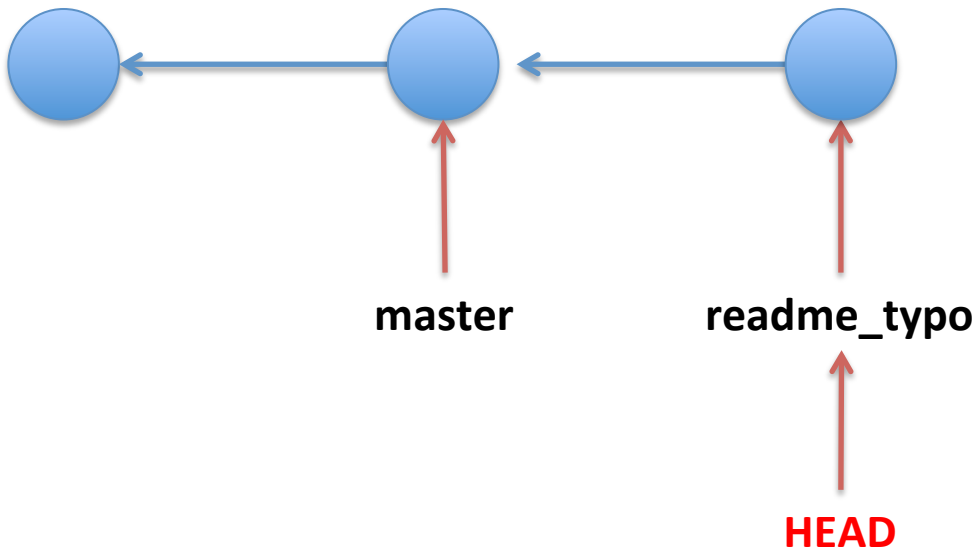


git checkout readme_typo

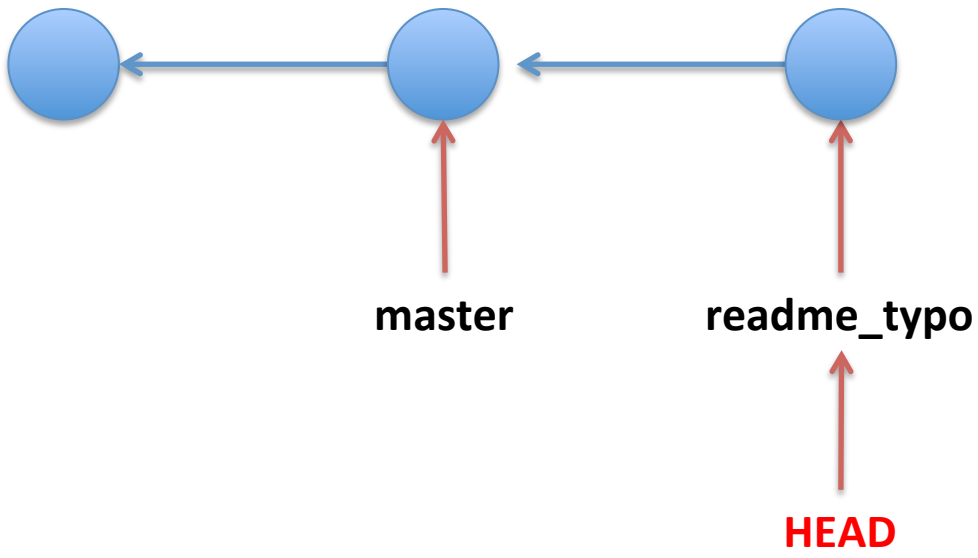



master
readme_typo
HEAD

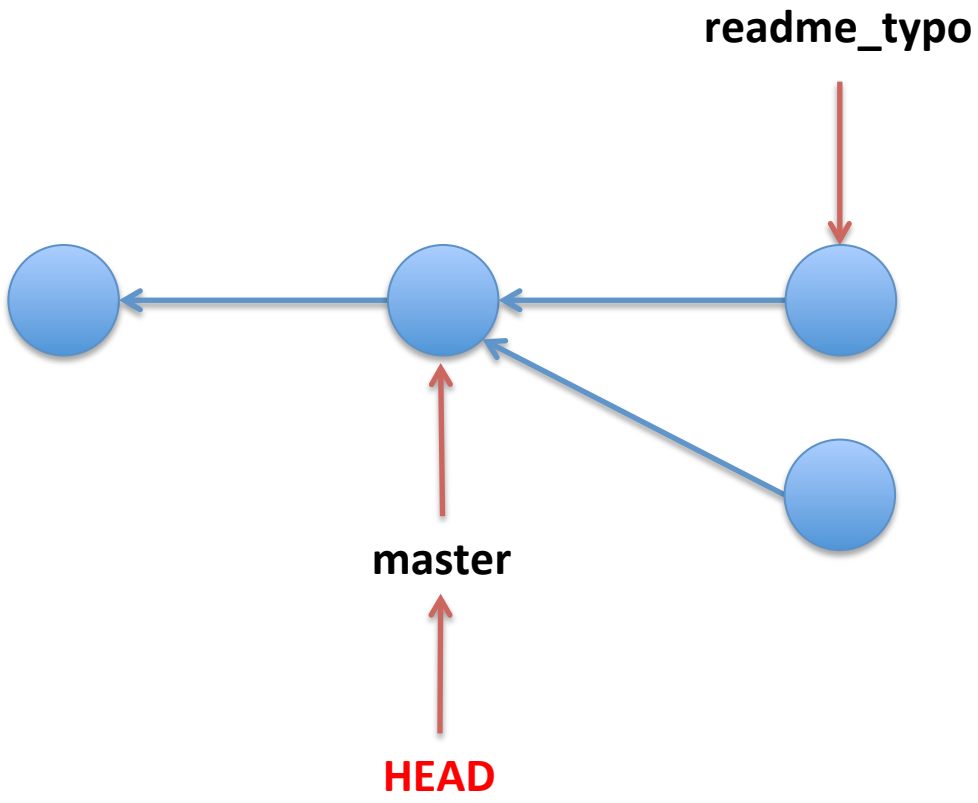
git commit



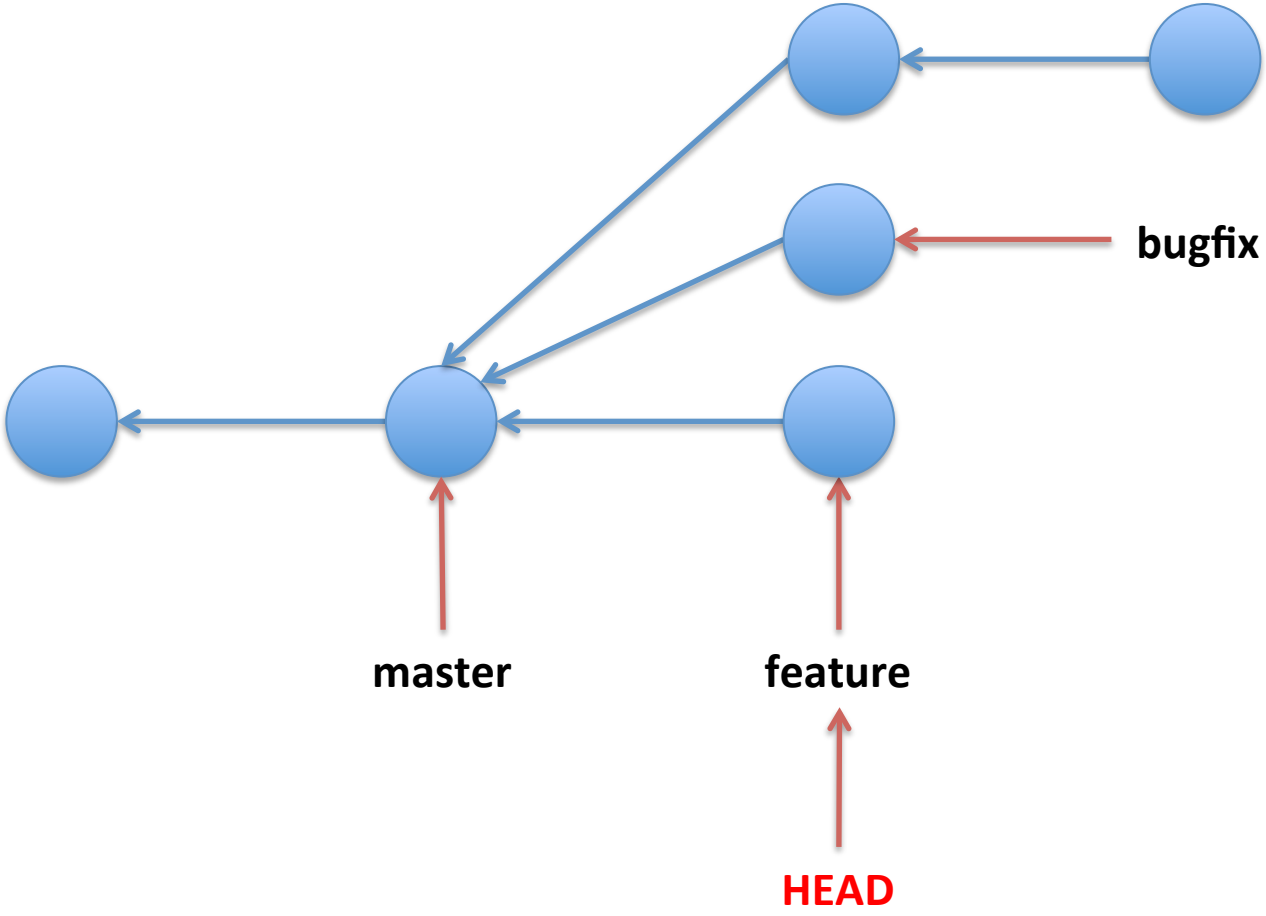
git checkout master



git commit



Quiz: Δημιούργησε τον γράφο



```
git init
git commit
git commit
git checkout -b bugfix
git commit
git checkout master
git checkout -b koko
git commit
git commit
git checkout master
git branch -D koko
git checkout -b feature
git commit
```

`git merge <branch>`

- Ενοποιεί το ιστορικό του branch που περνάς ως παράμετρο με το τωρινό branch
- Προσπαθεί να ενώσει τις αλλαγές στα αρχεία και από τα δύο branches
- Δημιουργεί ένα commit με 2 γονιούς:
  - Το τρέχον branch
  - Το branch που δίνεται ως παράμετρος

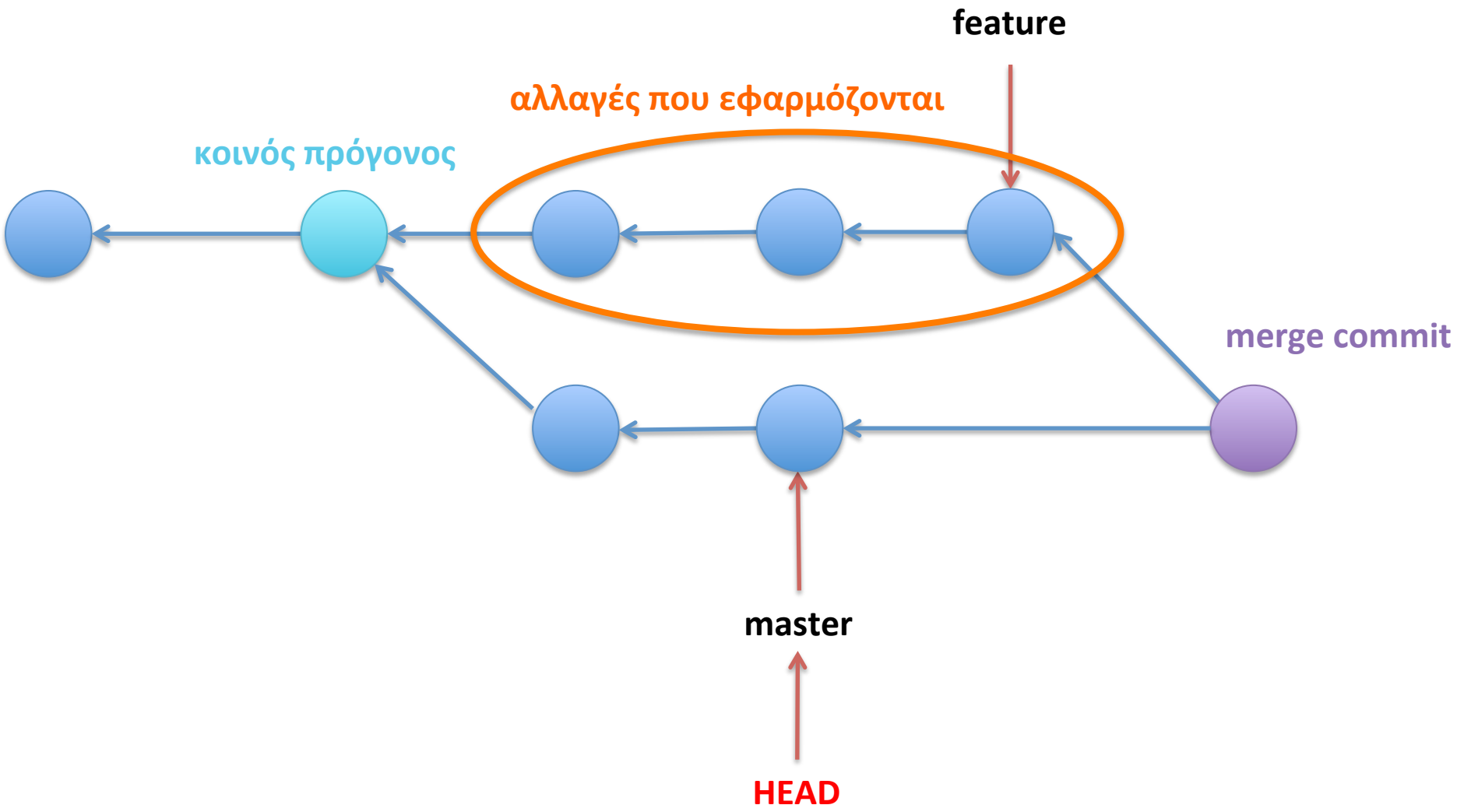
Αλγόριθμος merging

- Έστω ότι βρισκόμαστε στο master
- Τρέχουμε: `git merge feature`
- Αυτό δημιουργεί στο master τις αλλαγές που έγιναν εντωμεταξύ στο feature branch
- Δημιουργεί ένα νέο “merge commit” με δύο γονιούς:
  1. master
  2. feature

Αλγόριθμος merging

- Εντοπίζεται ο πιο πρόσφατος κοινός πρόγονος ανάμεσα σε master και feature
- Τα commits που πρέπει να εφαρμοστούν για να γίνει το merge είναι όλα τα commits ανάμεσα σε αυτόν τον πρόγονο και το feature branch
- Οι αλλαγές σε αυτά τα commits εφαρμόζονται στο master
- Το master μεταφέρεται στο νέο merge commit

(master) git merge feature



Branching workflow

- Προτείνουμε να δημιουργείς **ένα branch ανά feature**
- Για **κάθε** μικρό feature (π.χ. αλλαγή χρώματος ενός κουμπιού, διόρθωση ενός bug κλπ.) δημιουργούμε ένα νέο branch
- Κάνουμε τις αλλαγές μας στο νέο branch
- Κάνουμε όσα commits χρειάζονται
- Κάνουμε merge στο master
- Διαγράφουμε το νέο branch

Branching workflow

git checkout master

git checkout -b feature

vim

git add

git commit

πολλές φορές

git checkout master

git merge feature

git branch -d feature

Stash

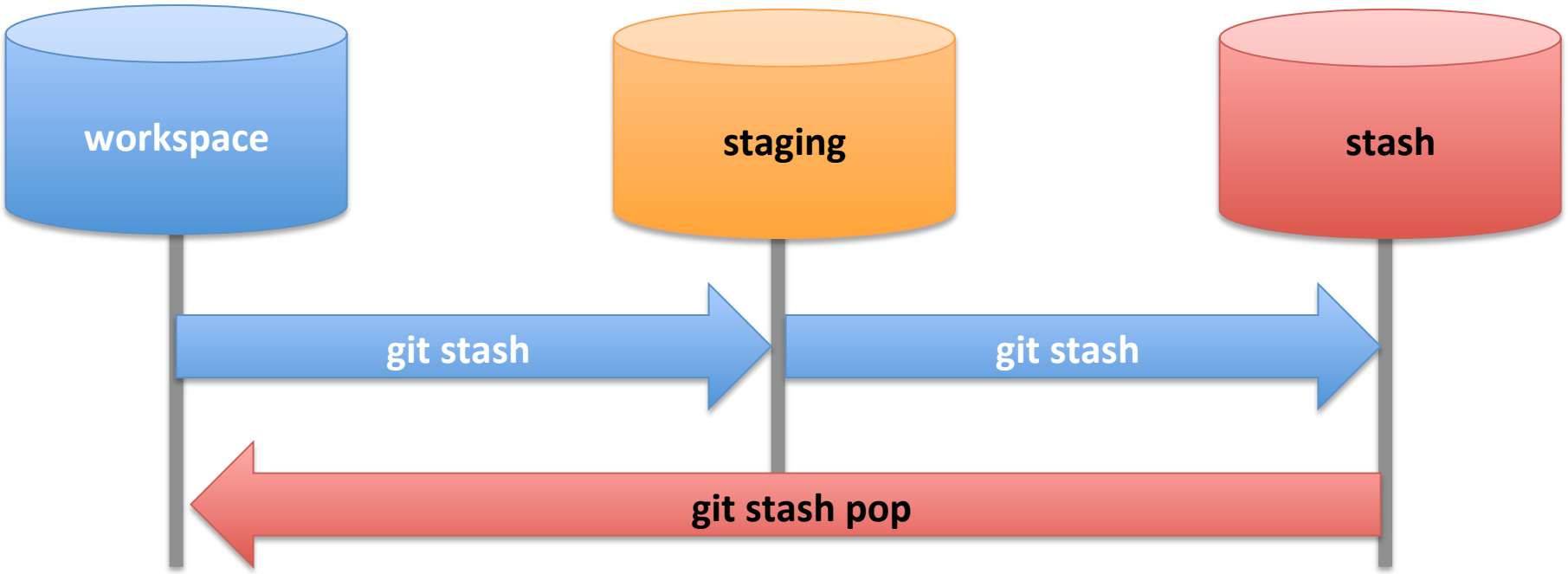
- Μερικές φορές χρειάζεται να αλλάξουμε branch για κάτι έκτακτο (π.χ. hotfix)
- Εντωμεταξύ μπορεί να έχουμε κάνει αλλαγές στο working copy μας
- Επιλογές:
  - commit… όμως δε θέλουμε να κάνουμε commit κάτι τσαπατσούλικο
  - checkout… όμως δε θέλουμε να χάσουμε τις αλλαγές μας

git stash

- Κρατάει στην άκρη τις αλλαγές:
  - στο working copy
  - στο staging area
- Καθαρίζει το working copy και το staging
- Πλέον μπορούμε να κάνουμε checkout ένα άλλο branch χωρίς να πρέπει να κάνουμε commit ή να χάσουμε τις αλλαγές μας

`git stash pop`

- Επανεφαρμόζει στο working copy τις αλλαγές που είναι αποθηκευμένες στο stash


workspace
staging
stash
git stash
git stash
git stash pop

Remotes

- Ο κώδικας είναι συνεργατικό πράγμα
- Μία ομάδα χρειάζεται πολλά αντίγραφα του κώδικά της, ένα για κάθε προγραμματιστή
- Μερικές φορές χρειάζεται να ανταλλάξουμε κώδικα με αυτά τα αντίγραφα
- Κάθε αντίγραφο με το οποίο ανταλλάσσουμε κώδικα ονομάζεται “remote”

GitHub

GitHub

- GitHub != git
- Ένα website που μπορούμε να ανεβάσουμε αντίγραφα των repo μας και να συνεργαστούμε
- Προσφέρει εργαλεία για συνεργασία

Ας συνεργαστούμε!

- Μπείτε στο https://github.com/dionyziz/git-class
- Πατήστε “Fork” πάνω δεξιά
- Τώρα έχετε φτιάξει ένα αντίγραφο του δικού μας repo το οποίο σας ανήκει και μπορείτε να το αλλάξετε

Cloning

- Τώρα που έχεις ένα δικό σου αντίγραφο στο GitHub, μπορείς να φτιάξεις ένα αντίγραφο στον υπολογιστή σου

git clone

git@github.com:*to-onoma-sou*/git-class.git

cd git-class

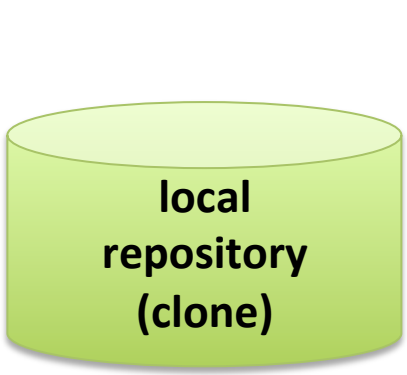
Forks & Clones

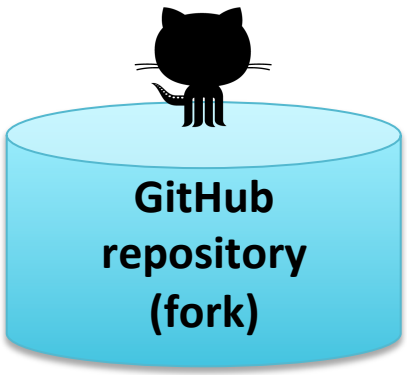
Πλέον έχεις **2 δικά σου repos**, αντίγραφα του δικού μας:

- Fork: Το αντίγραφο του repo μας εντός του λογαριασμού σου στο GitHub
- Clone: Το αντίγραφο στον υπολογιστή σου

Remotes

- Αυτά τα δύο repos συνδέονται:
  - το clone στον υπολογιστή σου
  - το fork σου εντός του GitHub
- Το repo στον υπολογιστή σου έχει ως **remote** το repo σου στο GitHub





Remotes

- Κάθε αντίγραφο του repo έχει τα δικά του commits, branches, και ιστορικό
- Κάποια από αυτά μπορεί να είναι κοινά
- Κάθε remote έχει:
  - το δικό του URL
  - ένα όνομα

git remote

`git remote`

- Δείχνει ποια είναι τα remotes του repo μας

`git remote add <name> <url>`

- Προσθέτει ένα καινούργιο remote με το οποίο σκοπεύουμε να συνεργαστούμε

`git remote rm <name>`

- Διαγράφει το remote

origin

- Όταν κάνουμε git clone δημιουργείται αυτόματα ένα remote με το όνομα “origin” το οποίο είναι το δικό μας αντίγραφο του repo στο GitHub
- Αυτό είναι το remote που θα χρησιμοποιείς περισσότερο

git push

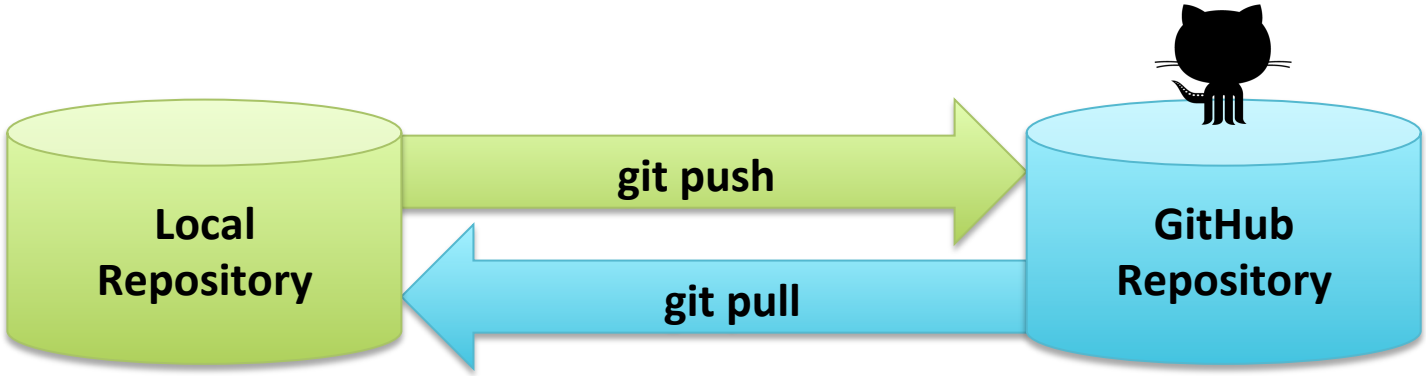
```
git push origin master
```

- Στέλνει στο branch master του remote με όνομα “origin” τα commits που έχουμε στο branch master τοπικά
- Έτσι δημοσιεύουμε τον κώδικά μας

git pull

```
git pull origin master
```

- Φέρνει στο τοπικό branch master τα commits που υπάρχουν στο master branch του remote με όνομα “origin”
- Έτσι κατεβάζουμε τις αλλαγές των άλλων


Local
Repository
git push
git pull
GitHub
Repository

Remote workflow

git checkout master

git pull origin master

git checkout -b feature

vim && git add && git commit

git checkout master

git merge feature

git push origin master

“Outdated”

```
dionyziz@erdos ~/workspace/ting2 (master*) % git push origin master
To git@github.com:dionyziz/ting.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:dionyziz/ting.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Outdated

- Κάποιος άλλος έκανε push commits εντωμεταξύ
- Για να μην γράψουμε **πάνω από τις αλλαγές των άλλων**, το git ζητάει να κάνουμε pull
- Με το pull γίνεται το εξής:
  - Κατεβαίνει ο νέος κώδικας (git fetch)
  - Ενοποιείται με τις δικές μας αλλαγές (git merge)
- Στη συνέχεια μπορούμε να κάνουμε push

Μάθαμε

- Τοπική χρήση του git
  - init
  - add/rm/mv
  - commit
  - log/lg/show
  - diff/diff --staged/diff HEAD
  - checkout

Μάθαμε

- Συνεργατική χρήση του git
  - branches
  - remotes
  - push/pull

Στο επόμενο...

- Περισσότερα για συνεργασία μέσω GitHub
  - Pull requests
  - Reviews
  - Issues
- Προχωρημένες τεχνικές git
  - Undo
  - Blame
  - Tag
  - Cherry pick
  - Rebase
- Workflows και συνεργατικές τεχνικές


Ευχαριστούμε!