

# HOW TO → git

Software Libre Society  
Πανεπιστήμιο Πειραιώς



SOFTWARE **LIBRE** SOCIETY  
UNIVERSITY OF PIRAEUS

# Credits:

- Κυριάκος Γιαννάκης – Διαφάνειες, Παρουσίαση
- git-class.gr
  - Θέμης Παπαμελετίου
  - Διονύσης Ζήνδρος
- Πέννυ Θωμοπούλου – QC
- Δημήτρης Τζιλιβάκης – Artwork, Posters, Banners
- Αλέξανδρος Μπαμπούνης Τσάτσος, Νεφέλη Βασιλάκη - Οργάνωση
- Σωτήρης Καραγεωργόπουλος, Νικόλας Ζαφειρόπουλος - Εγγραφές

# Το πρόβλημα

- Ως προγραμματιστές έχουμε ανάγκες
  - Ο 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;

# Enter Version Control

- Κρατάμε «εκδόσεις» σε 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 git



- Before Git there were:
  - CVS - 1990
  - Subversion (SVN) - 2000
- Created by Linus Torvalds, for Linux Development.
- *Git: unpleasant person* in British English slang
- “I'm an egotistical bastard, and I name all my projects after myself. First ‘Linux’, now ‘git’ ”

# Τι πάει να πει ‘Git’;

Ας το αναλύσουμε με μερικές λέξεις κλειδιά...

- Ιστορικό
- Οργάνωση
- Συνεργασία

# Αλλά πρώτα...

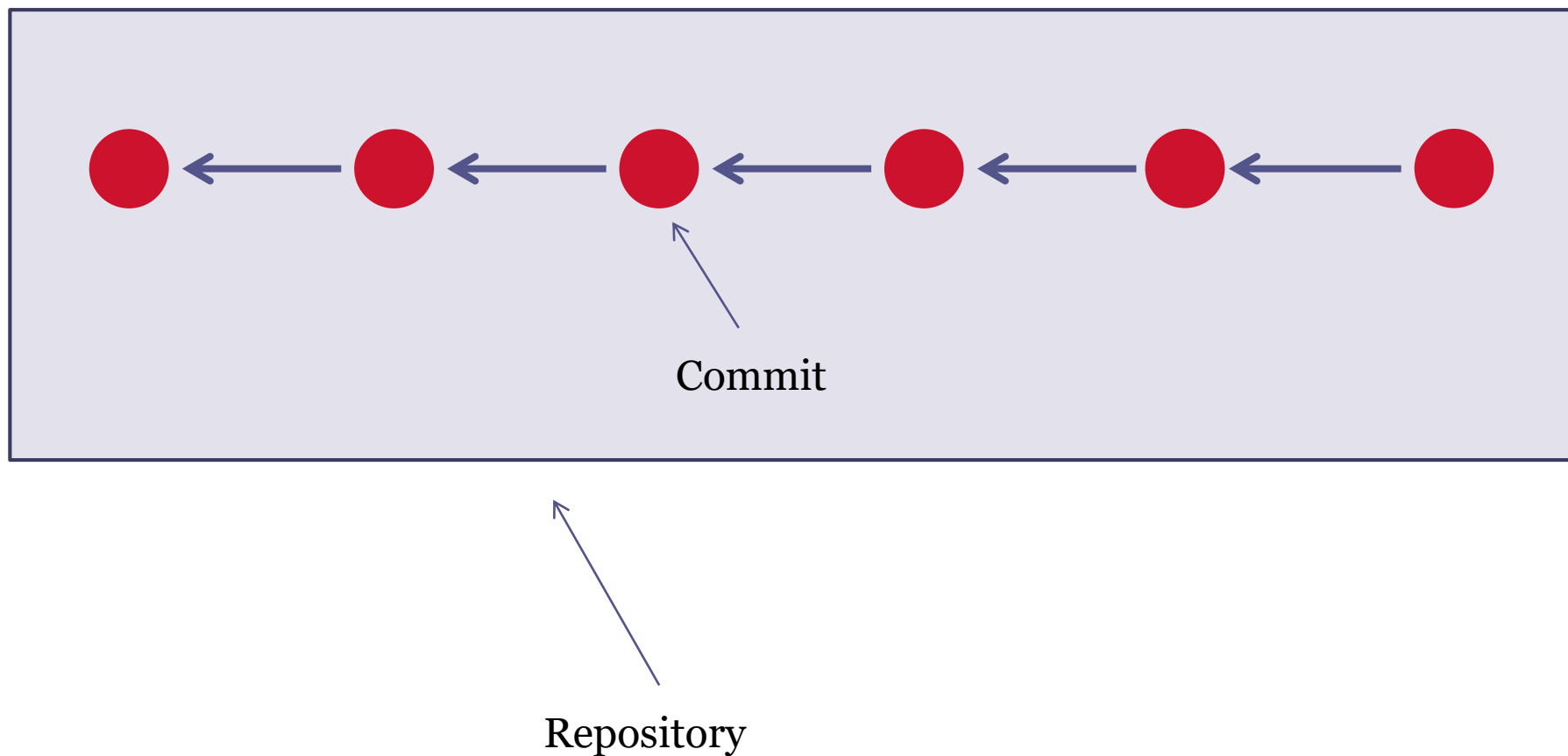
- Εγκατάσταση σε Linux:
  - `sudo apt-get install git` (Debian, Ubuntu)
  - `sudo dnf -y install git` (RedHat, Fedora)
  - `sudo pacman -Sy git` (Arch, Manjaro)
- Εγκατάσταση σε macOS:
  - Τρέχεις git και ακολουθείς τις οδηγίες εγκατάστασης
- Εγκατάσταση σε Windows:
  - <https://git-scm.com>



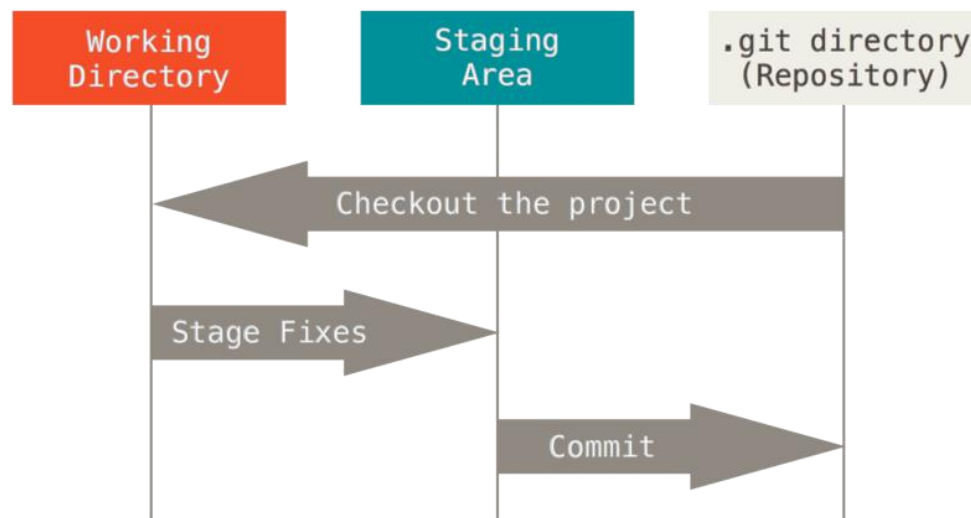
# Αλλά πρώτα...

- Ρύθμιση του git:
  - «Ποιος είσαι και τι κάνεις;;;»
    - `git config --global user.name "Kyriakos Giannakis"`
    - `git config --global user.email "kerk12gr@gmail.com"`
  - Αν είσαι σε Windows (X Compatibility with Unix):
    - `git config --global core.autocrlf true`

# Ιστορικό



# Staging



## ➤ Unstaged/Modified:

Έχουν γίνει **τροποποιήσεις** σε ένα αρχείο.

Ακόμα **ΔΕΝ ΠΑΡΑΚΟΛΟΥΘΕΙΤΑΙ** από το Git.

## ➤ Staged:

Οι αλλαγές που έχουν γίνει μέχρι στιγμής **ΠΑΡΑΚΟΛΟΥΘΟΥΝΤΑΙ**.

Οι αλλαγές **θα προστεθούν στο επόμενο Commit**.

## ➤ Committed:

Οι αλλαγές αποθηκεύτηκαν στην βάση δεδομένων του git.

# Οι 4 μαγικές εντολές!

- git init → Initialize Repository
- git status -> View the Status of the Repo
- git add -A → Stage ALL files
- git status -> View the Status of the Repo
- git commit -m "My message goes here" → Commit all staged files.

# 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
```

# Τι είναι ένα Commit;

- 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`
  - Αποτελεί Hash του περιεχομένων και των μεταδεδομένων του commit.

# git log

```
root@saturn:~/BackupSaturn# git log
commit 064a93ff7bedf20aca0b8b31ed068c28e0b69b86
Author: Kyriakos Giannakis <kerk12gr@gmail.com>
Date: Thu Dec 14 16:58:48 2017 +0200

    Added maintenance mode enablement.

commit 981ff152adc5560709d749e0c6b596cc9d7e5078
Author: Kyriakos Giannakis <kerk12gr@gmail.com>
Date: Thu Dec 7 20:23:15 2017 +0200

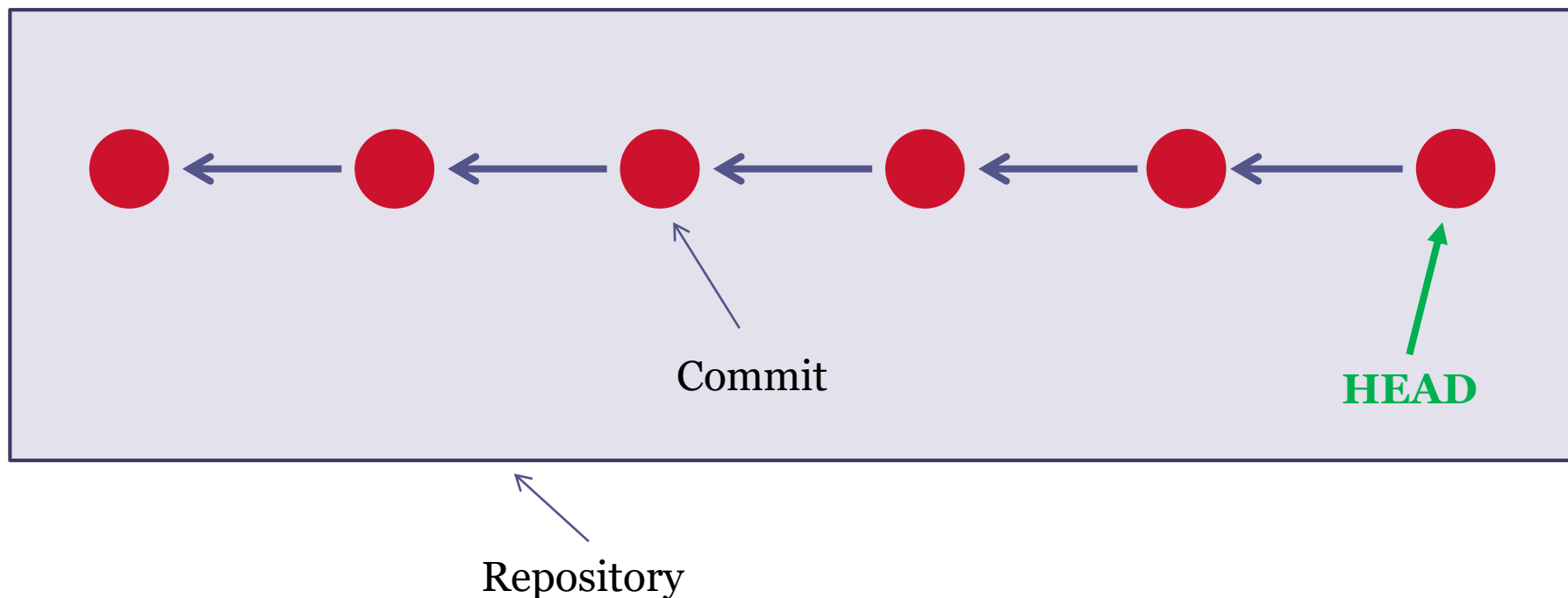
    Hello World!
```

Αναγνωριστικό

Δημιουργός/Ημερομηνία

Μήνυμα Commit

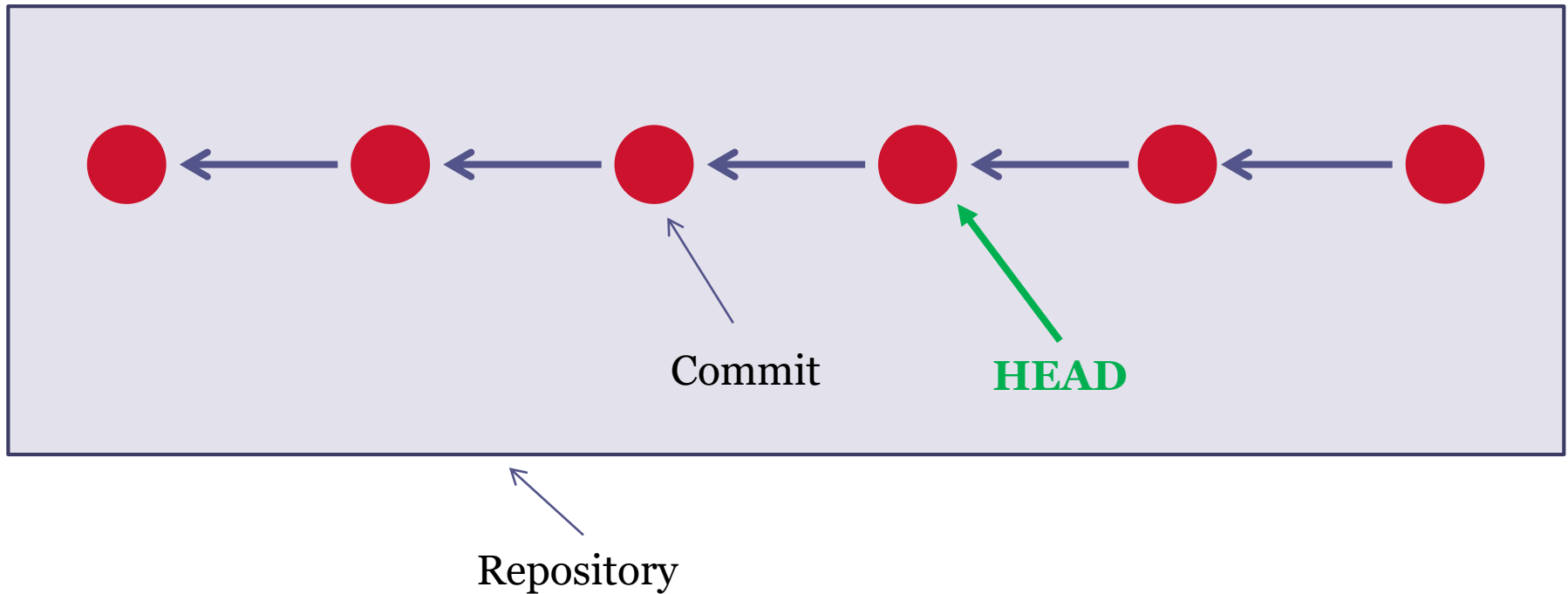
# Ιστορικό - Pt.2: Checkout



HEAD: Ένας δείκτης που δείχνει το commit το οποίο βλέπουμε τώρα.



# Ιστορικό - Pt.2: Checkout



Let's go to the past!

# How to Checkout

1. Αντιγράφουμε τους πρώτους, τουλάχιστον 4, χαρακτήρες του Αναγνωριστικού Commit

```
root@saturn:~/BackupSaturn# git log
commit 064a93ff7bedf20aca0b8b31ed068c28e0b69b86
Author: Kyriakos Giannakis <kerk12gr@gmail.com>
Date: Thu Dec 14 16:58:48 2017 +0200
```

```
    Added maintenance mode enablement.
```

```
commit 981ff152adc5560709d749e0c6b596cc9d7e5078
Author: Kyriakos Giannakis <kerk12gr@gmail.com>
Date: Thu Dec 7 20:23:15 2017 +0200
```

```
    Hello World!
```

2. `git checkout 981ff152ad`

# Άλλες χρήσιμες εντολές

- `git diff` → Δείχνει τι αλλαγές υπάρχουν από το τελευταίο commit μέχρι τώρα.
- `git diff -staged` → Δείχνει τι αλλαγές έχουν γίνει που θα καταγραφούν στο επόμενο commit.
- `git diff <commit_id1> <commit_id2>` → Δείχνει αλλαγές ανάμεσα σε 2 commits.
- `git show <commit_id>` → Δείχνει τι αλλαγές έχουν γίνει σε ένα συγκεκριμένο commit.

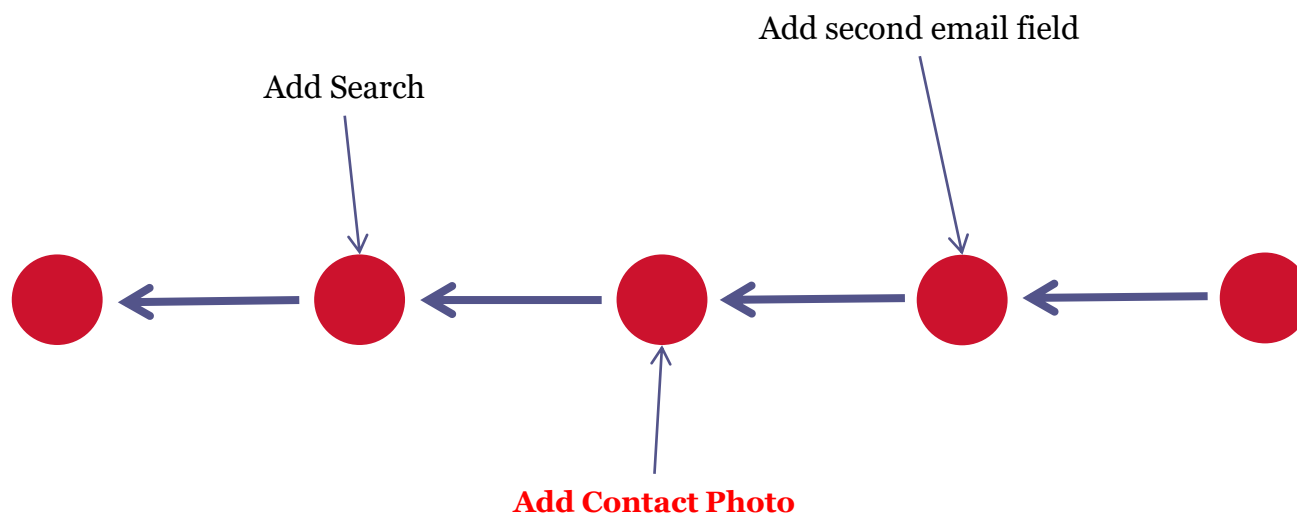
# Ενέργειες στο Staging Area:

- **git rm <file>**
  - Διαγράφει ένα αρχείο και ενημερώνει το git
    - Πρέπει να ακολουθείται από commit
  - Κατά μία έννοια το «αντίθετο» του add
- **git mv <source> <destination>**
  - Μεταφέρει ένα αρχείο και ενημερώνει το git
    - Πρέπει να ακολουθείται από commit
  - git mv:
    - mv <old path> <new path>
    - git rm <old path>
    - git add <new path>

# .gitignore

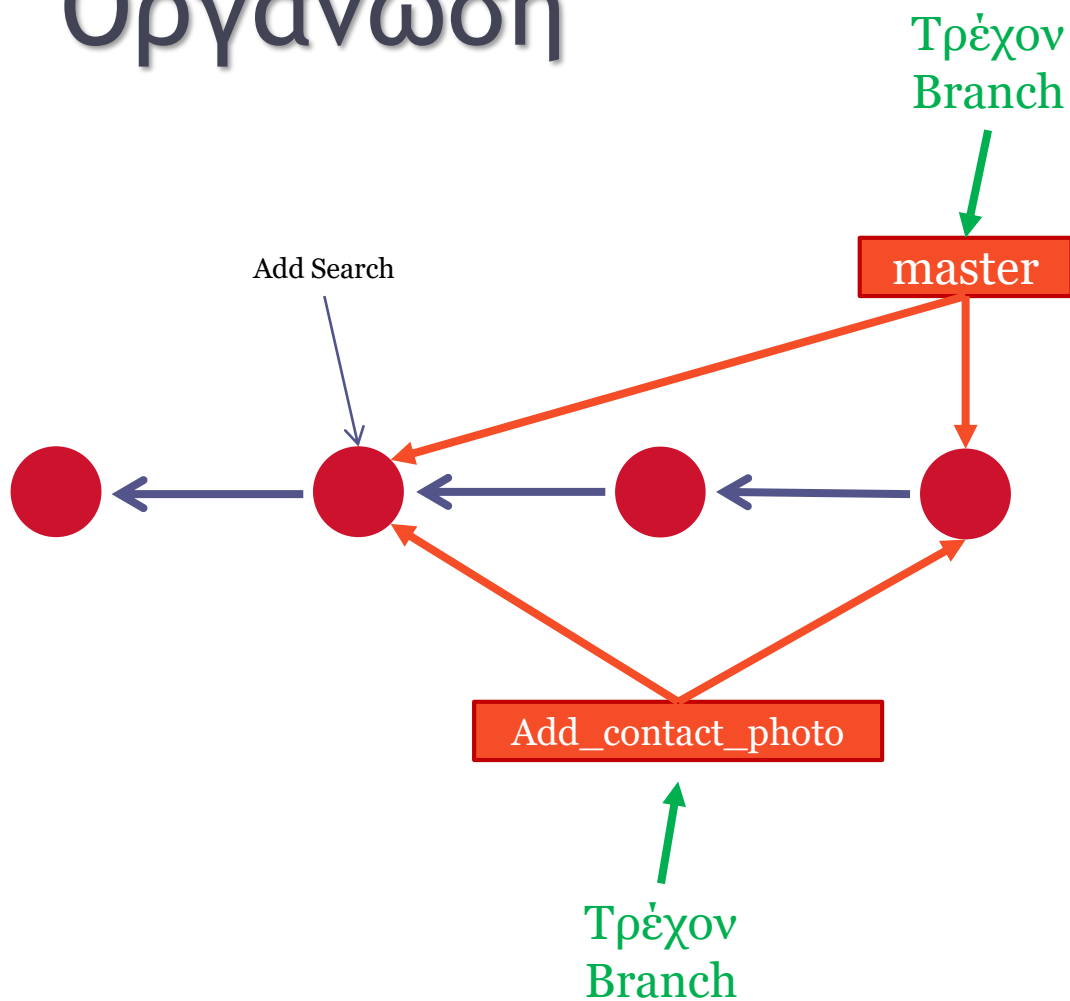
- Ένα αρχείο στην ρίζα (αρχικό κατάλογο) του Repository.
- Μέσα γράφουμε όσα αρχεία θέλουμε να αγνωεί το git, ένα ανά γραμμή.
  - Πχ:
    - `program.py`  
`program2.py`
- Μπορούμε να βάλουμε μοτίβα αρχείων.
  - Πχ. `*.pyc`
  - Το παραπάνω θα αγνωήσει όλα τα αρχεία που έχουν την κατάληξη `.pyc`

# Οργάνωση



# Οργάνωση

Master: Το Default Branch που έχουν (σχεδόν) όλα τα αποθετήρια.

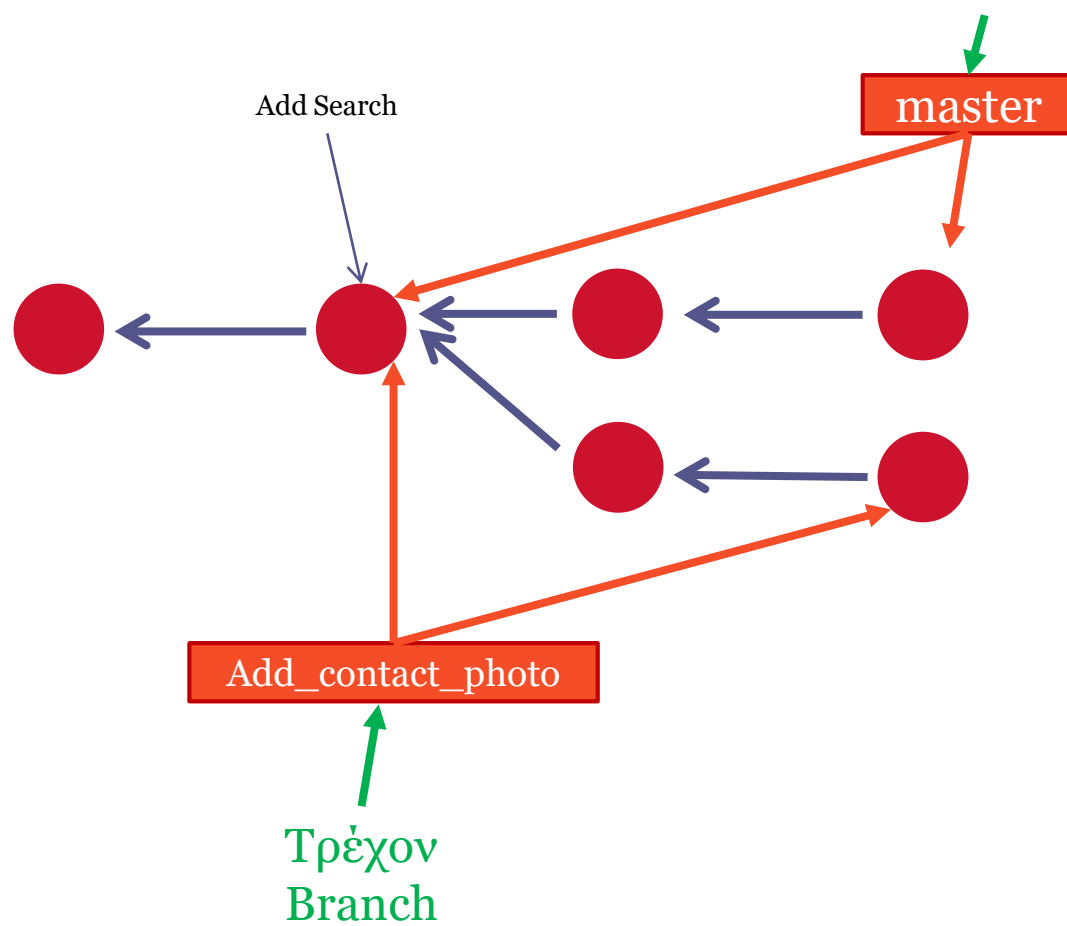


```
git checkout -b <branch_name>
=
git branch <branch_name>
git checkout <branch_name>
```

```
git checkout -b Add_contact_photo
...
git add -A
git commit -m "...
...
git add -A
git commit -m "...
git checkout master
git merge Add_contact_photo
git branch -d Add_contact_photo
```

Τρέχον  
Branch

master



```
git checkout -b Add_contact_photo
...
git add -A
git commit -m "..."/>
...
git add -A
git commit -m "..."/>
git checkout master
git merge Add_contact_photo
git branch -d Add_contact_photo
```



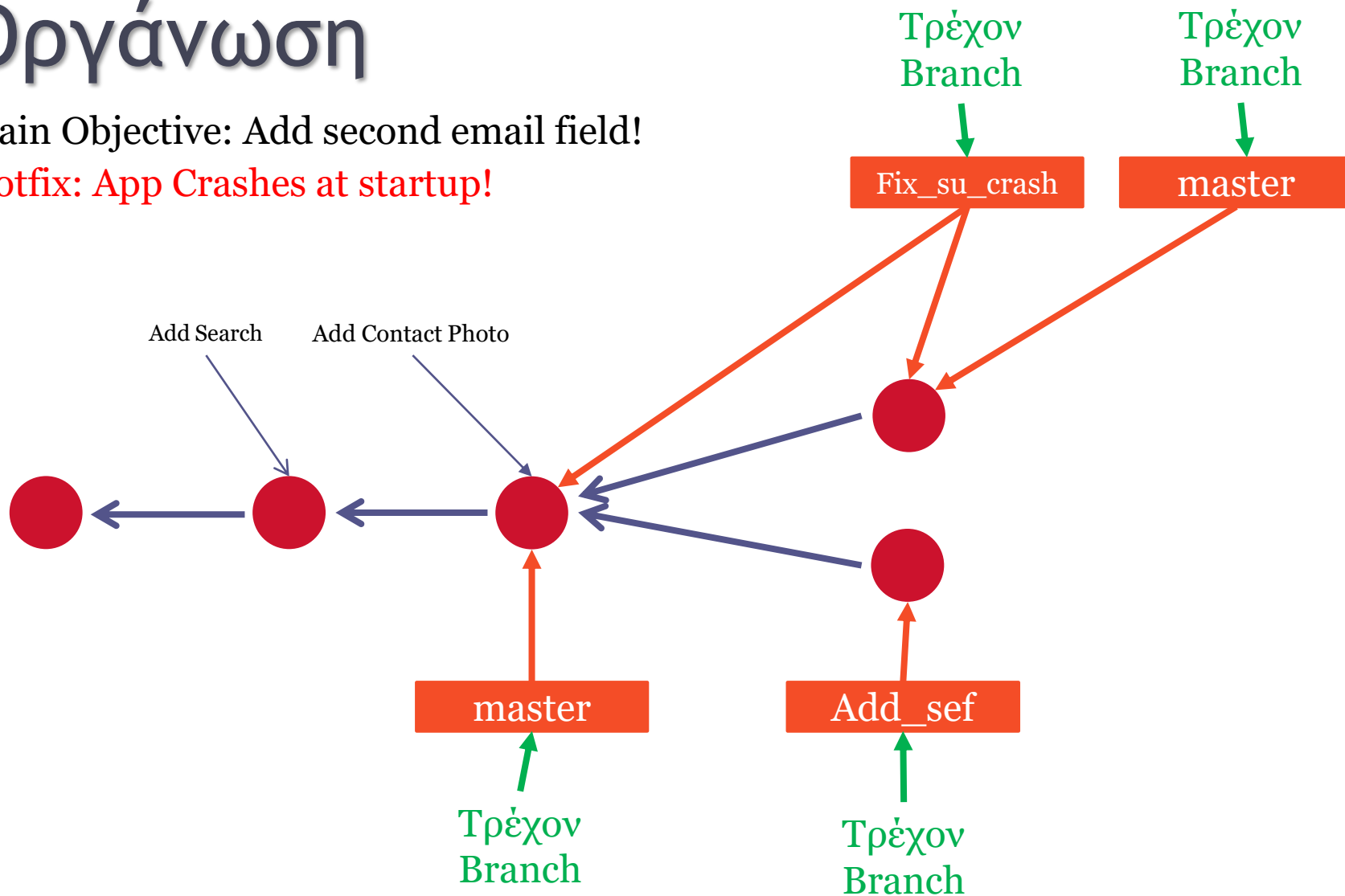
# Branches

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

# Οργάνωση

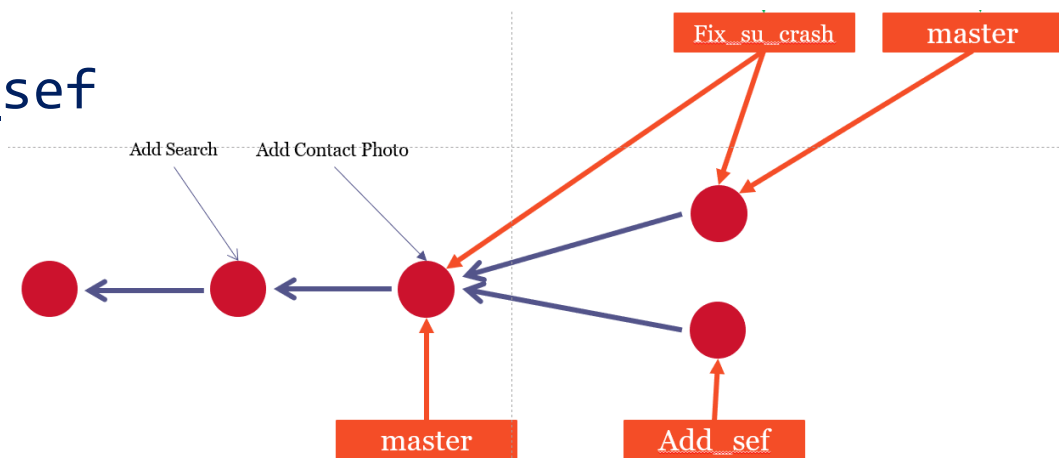
Main Objective: Add second email field!

Hotfix: App Crashes at startup!



# Πως δημιουργείται ο προηγούμενος γράφος;

- `git checkout -b Add_sef`
- ...
- `git add -A`
- `git commit -m "..."`
- \*Client calls\*
- `git checkout master`
- `git checkout -b fix_su_crash`
- ...
- `git add -A`
- `git commit -m "..."`
- `git checkout master`
- `git merge fix_su_crash`
- `git branch -d fix_su_crash`



```
git checkout -b <branch_name>
=
git branch <branch_name>
git checkout <branch_name>
```

# Workflow:

- Έστω οτι πρέπει να εισαχθεί ένα Feature (Feature A) στον κώδικα...
- Αποτελείται από 2 commits
- `git checkout -b feature_a` → Δημιουργία Branch
- ...
- `git add -A`
- `git commit -m "..."` } Πρώτο Commit
- ...
- `git add -A`
- `git commit -m "..."` } Δεύτερο Commit
- `git checkout master` → Αλλαγή ξανά στο master
- `git merge feature_a` → Ένωση του feature\_a με το master

# Stash:

- Έστω οτι κάνουμε δουλειά και ξαφνικά πρέπει να αλλάξουμε branch για κάτι έκτακτο (hotfix).
- Δεν είμαστε έτοιμοι ακόμα για commit (τσαπατσούλικο).
- `git stash`
- Κρατάει στην άκρη τις αλλαγές:
  - στο `working copy`
  - στο `staging area`
- Καθαρίζει το `working copy` και το `staging`

# Stash:

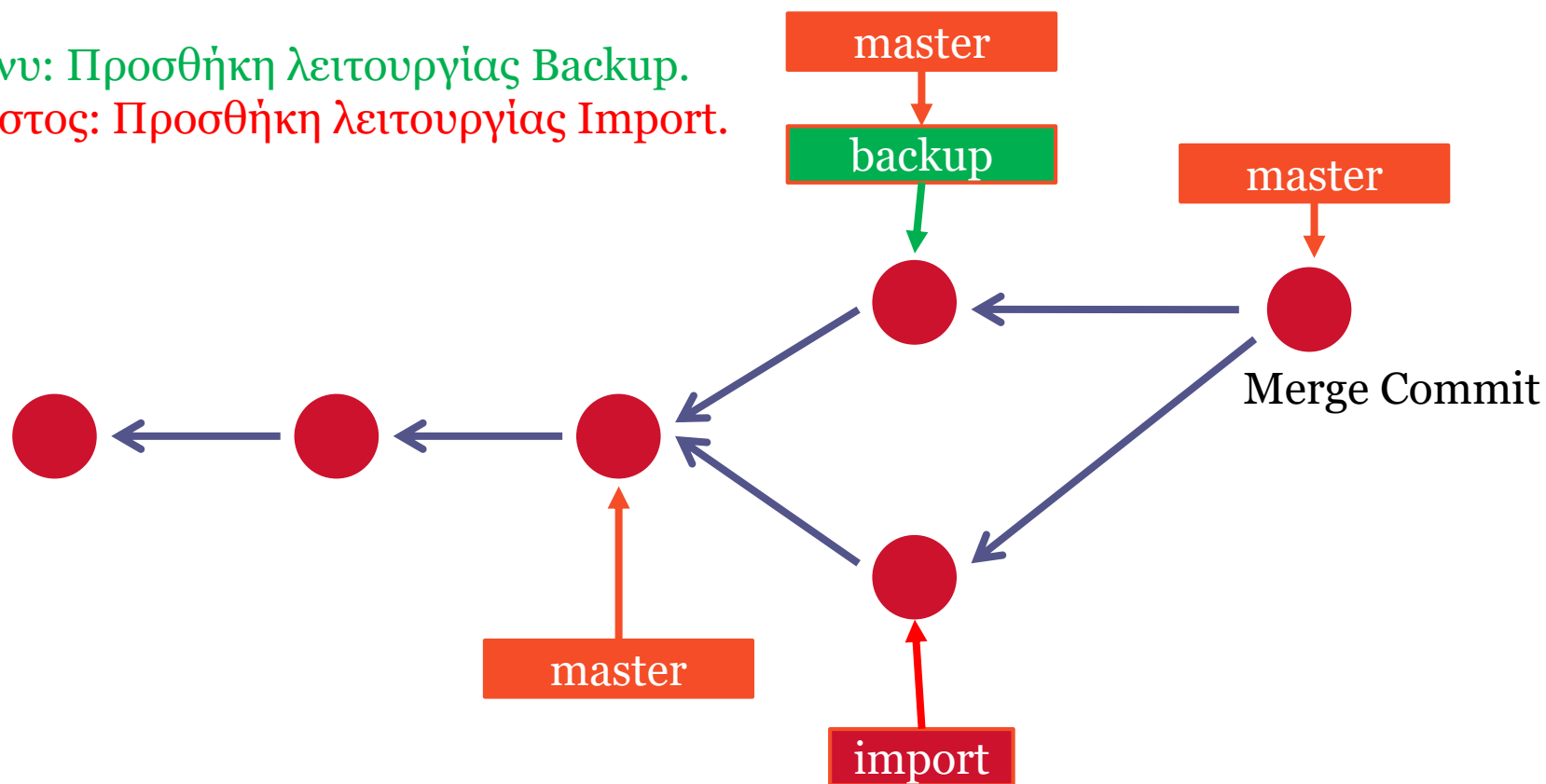
- Όταν είμαστε έτοιμοι να συνεχίσουμε, επιστρέφουμε στο branch που δουλεύαμε και...
- `git stash pop`
- Continue working...

# Το Git σε απομακρυσμένο επίπεδο

# Συνεργασία - Workflow:

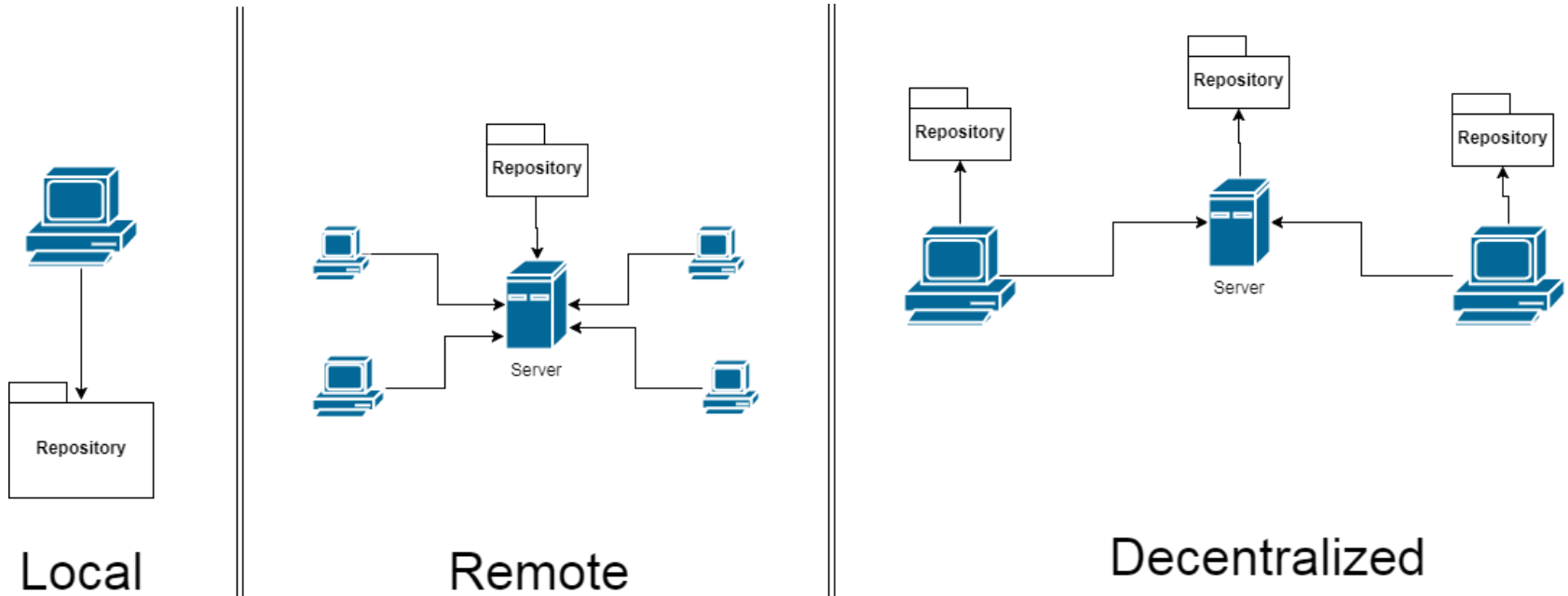
Πέννυ: Προσθήκη λειτουργίας Backup.

Χρήστος: Προσθήκη λειτουργίας Import.





# Version Control Decentralization



# Git Remote - Τι είναι;

Ένας απομακρυσμένο Repository που διατηρείται ένα αντίγραφο του τοπικού Repository.

# Cloning:

- Αντιγραφή του απομακρυσμένου Repo στον υπολογιστή/τοπικά.

```
git clone
```

```
https://github.com/cs-unipi-FOSS/InstallfestHelper
```

```
cd InstallfestHelper
```

- Πάντα, όταν γίνεται clone ένα Repo, το απομακρυσμένο Repo παίρνει το όνομα **origin**.

# Push:

- Έχουν γίνει κάποιες αλλαγές, θέλουμε να τις στείλουμε στο απομακρυσμένο Repo.
- `git commit`
- `git push <remote> <branch>`
- Πχ. `git push origin master`
- **ΔΕΝ κάνουμε push απευθείας στο Master.**  
Δημιουργούμε δικό μας branch και δουλεύουμε εκεί.
- **ΔΕΝ κάνουμε Push αν δεν είμαστε σίγουροι για τα commits που έγιναν.**

Σοβαρά! Μην κάνετε Push στο Master!



# Pull

- Φέρνει στο τοπικό ενεργό branch τα commits που υπάρχουν στο branch του remote που επιλέξαμε.
- Έτσι κατεβάζουμε τις αλλαγές των άλλων
- πχ: `git pull origin master`

# Remote Management:

## `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

# GitHub

- Μια υπηρεσία φιλοξενίας αποθετηρίων Git.
- Προσφέρει εργαλεία συνεργασίας και οργάνωσης.

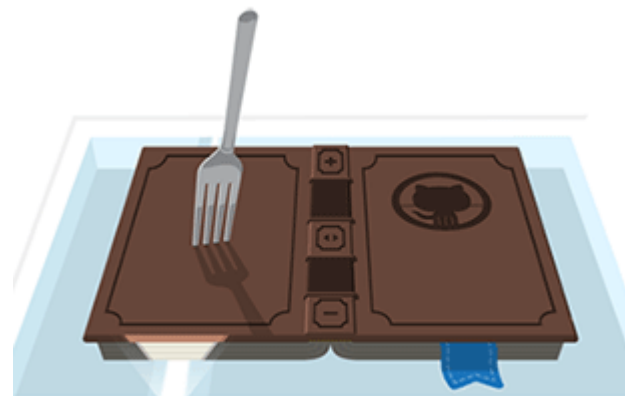




# Forks:

- Το Github παρέχει την δυνατότητα πλήρους αντιγραφής ενός αποθετηρίου στο δικό μας προφίλ.
- Παράγεται ένα ακριβές αντίγραφο του αποθετηρίου στο οποίο έχουμε πλήρη δικαιώματα τροποποίησης.

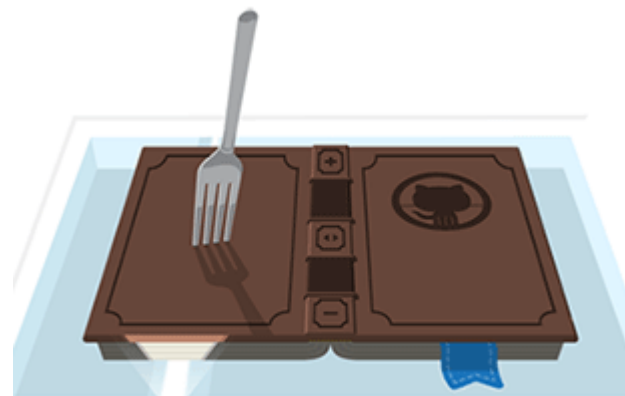
 **hpanago / VMInit**  
forked from [kerk12/VMInit](#)



# Forks:

- Μέσω αυτών κάνουμε βελτιώσεις σε ένα υπάρχον Repository, ή ξεκινάμε το δικό μας Project βασιζόμενοι σε κάποιο άλλο Repository.

 **hpanago / VMInit**  
forked from [kerk12/VMInit](#)



# Pull Requests:

- Το αίτημα για να κάνει ένας χρήστης Pull ένα δικό μας branch σε κάποιο δικό του (συνήθως το Master)
- Χρησιμοποιούνται προκειμένου να:
  - Εισαχθούν νέα Features
  - Διορθωθούν Bugs
  - Προστεθεί Documentation

# Pull Requests:

- 1. Μέχρι τώρα πρέπει να έχουν γίνει:
  - `git clone ...`
  - `git checkout -b new_feature1`
  - \*αλλαγές\*
  - `git add && git commit`
  - `git push -u origin new_feature1`
- Αυτό ανεβάζει το branch feature στο Forked Repo.
- 2. Δημιουργούμε από το Github ένα PR.



This repository Search

[Pull requests](#) [Issues](#) [Gist](#)Oblynx / **git-class**[Watch](#) ▾

2

A 4h class to teach git and GitHub

42 commits

2 branches

0 releases

3 contributors

Your recently pushed branches:

**themicp:feature** (5 minutes ago)[Compare & pull request](#)Branch: **master** ▾**git-class** / +

This branch is 42 commits ahead, 1 commit behind dionyziz:master.

[Pull request](#) [Compare](#)[Merge pull request #3](#) from themicp/gh-pages ... **Oblynx** authored 5 hours agolatest commit **ba3cb3c182** [images](#) Add NTUA logo, remove shadows from logos

6 days ago

[javascripts](#) Create gh-pages branch via GitHub

12 days ago

# Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: **Oblynx/git-class** ▾base: **master** ▾

...

head fork: **themicp/git-class** ▾compare: **feature** ▾

✓ **Able to merge.** These branches can be automatically merged.



Added a link to the slides of Day 1

Write

Preview

Markdown supported

Uploaded the pptx file and added a link in the HTML page.

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

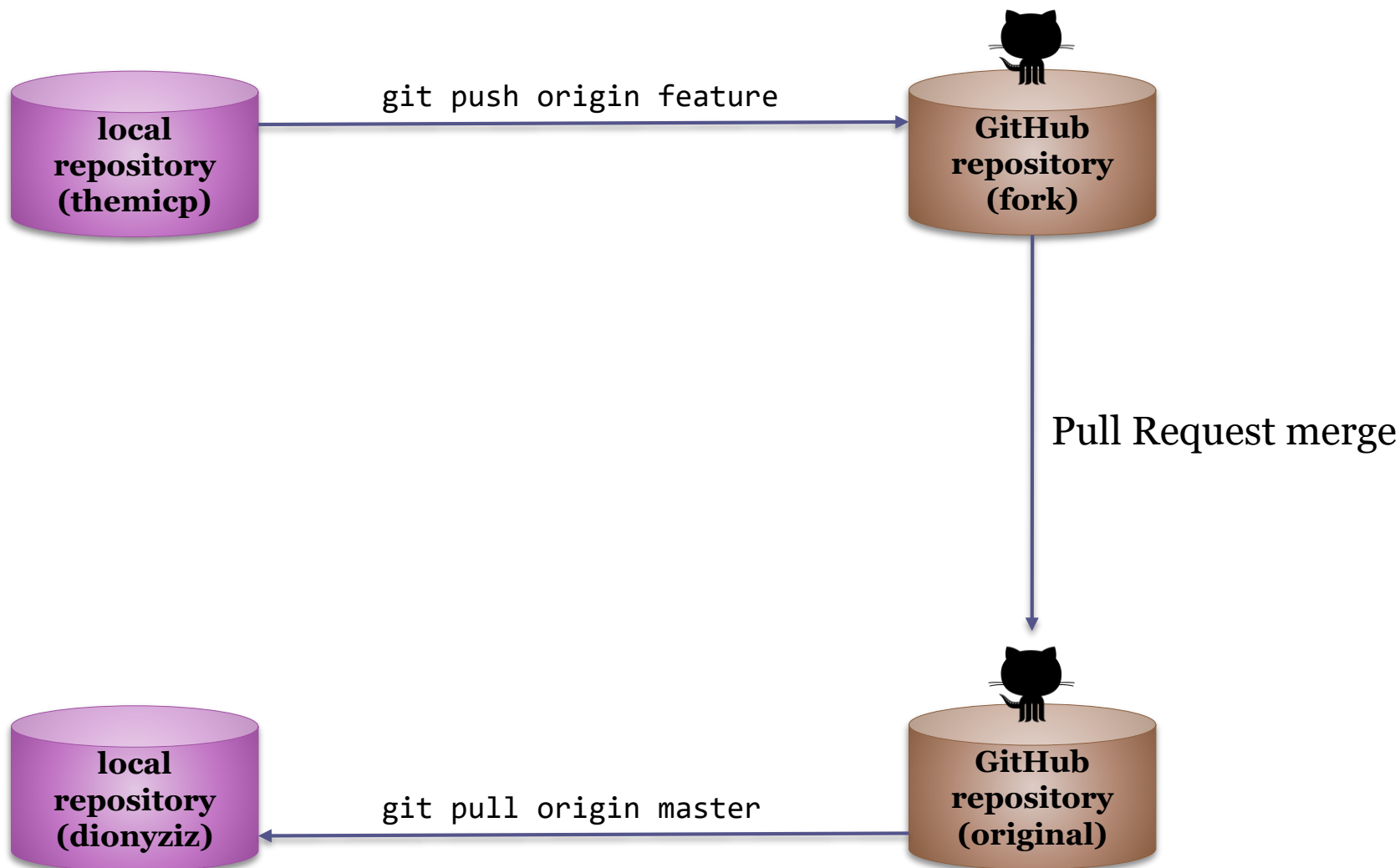
[Create pull request](#)

1 commit

1 file changed

0 commit comments

1 contributor



# Pull Requests - Reviews

- Το pull request ελέγχεται για ποιότητα κώδικα πριν γίνει merge.
- Αν υπάρχουν προβλήματα, γράφονται σχόλια που αφορούν την ποιότητα του κώδικα.
- Ο δημιουργός του pull request διορθώνει τα προβλήματα.
  - Κάνοντας push νέα commits στο υπάρχον feature branch του fork του.



```
53 +     if (username == '') {
54 +         winston.info('empty');
```



**dionyziz** added a note 25 days ago

Owner

Use more descriptive error messages.

Add a line note



**dionyziz** commented on an outdated diff 25 days ago

Hide outdated diff

realtime/tests/server.js

```
@@ -37,8 +37,39 @@ describe('Node Server', function() {
37 37         done();
38 38     });
39 39 });
40 - });
40 +
41 +     it('rejects an empty username', function(done) {
42 +         socket.emit('login', '');
43 +
44 +         socket.on('login-response', function(resp) {
45 +             expect(resp.success).toBe(false);
46 +             expect(resp.error).toBe('empty');
47 +             done();
48 +         });
49 +     });
50 +
51 +
```



**dionyziz** added a note 25 days ago

Owner

Avoid double blank lines

Add a line note

# GitHub Issues:

- Bug tracking σύστημα
- Κάθε issue
  - Είναι ένα bug ή ένα task
  - Είναι ανοιχτό (open) ή κλειστό (closed-resolved)
  - Έχει ένα description
  - Έχει comments
  - Μπορούμε να του βάλουμε 1 ή παραπάνω Label(s), για να το κατατάξουμε σε μια κατηγορία (πχ Bug).
- Issues μπορούν να ανοίξουν όλοι (και μη ιδιοκτήτες του repo)



dionyziz / endofcodes

Watch 3

★ Star 10

Fork 10

Issues

Pull requests

Labels

Milestones

Filters

is:issue is:open

New issue

96 Open ✓ 66 Closed

Author

Labels

Milestones

Assignee

Sort

! Stop migration procedure in case that .history has wrong permissions

#385 opened on Aug 12 by pkakelas

0

! Write custom error on permissions error in migration UI.

#384 opened on Aug 12 by pkakelas

0

! [!] Memory limit reached. bug

#383 opened on May 17 by nikatlas



1

! Unauthorized after Register. String userID bug

#381 opened on May 17 by nikatlas

5

! Coverage report

#378 opened on Aug 27, 2014 by dionyziz

0

! Attribute generation in HTML helper DRY

#376 opened on Aug 16, 2014 by dionyziz



0

! Beautify forgot password page

#370 opened on Aug 12, 2014 by dionyziz



0

! Remember me checkbox must be clickable through its label on login

#369 opened on Aug 12, 2014 by pkakelas



0

! Username field must be focused by default on login page

#368 opened on Aug 12, 2014 by pkakelas



0

! Make boturl work without http

#367 opened on Aug 12, 2014 by pkakelas

0

&lt;&gt;

!





dionyziz / endofcodes

Watch 3

★ Star 10

Fork 10

## [!] Memory limit reached. #383

New Issue



Open nikatlas opened this issue on May 17 · 1 comment



nikatlas commented on May 17

I created 2 default nodejs bots and ran gamescript.sh.

First of all the script timeout after a minute.

Secondly and most important, from now on there is this error on my home page:

Fatal error: Allowed memory size of 134217728 bytes exhausted (tried to allocate 65552 bytes) in /var/www/html/endofcodes/models/db.php on line 39

Debugging showed that this memory allocations is because it tries to load the last game.

The last game had 131671 rounds (even after timeout)

On Game Model constructor if id is passed the whole game is loading on memory including data for every round and this caused me the memory limit reached exception.

Actually the model load exactly 4439 rounds before crashing and that's only with two bots.

Maybe the game object shouldnt load all the states on memory but only current.

```
endofcodes/models/game.php - Sublime Text (UNREGISTERED)
protected static $attributes = [ 'width', 'height', 'created' ];

public static function getGame($id) {
    if ( $game = dbSelect( 'games', [ 'id' ], [ ], 'created DESC', 1 ) ) {
        throw new ModelNotFoundException();
    }
    return new Game( $game[0][ 'id' ] );
}

public function __construct($id = false, $roundid = false) {
    require_once 'models/round.php';
    if ( $id == false ) {
        $this->exists = true;
        try {
            $game_info = dbSelectOne( 'games', [ 'created', 'width', 'height' ], compact( 'id' ) );
        } catch ( DBException $e ) {
            throw new ModelNotFoundException();
        }
    }
}
```

Labels

bug

Milestone

No milestone

Assignee



VitSalis

Notifications

Subscribe

You're not receiving notifications from this thread.

3 participants



# Αναγνωριστικά GitHub Issues

- Μοναδικό αναγνωριστικό για κάθε issue μέσα στο repo
- Μπορούμε να αναφερθούμε σε αυτό
  - από το commit message:
    - `git commit -m "Refactor code for issue #765"`
    - `git commit -m "Closes #765"` (θα κλείσει το issue 765)
  - από PR descriptions παρομοίως
- Το GitHub μετατρέπει τις αναφορές σε links

# Remote Workflow

- Ας ανακεφαλαιώσουμε:  
Έστω οτι θέλουμε να βάλουμε ένα νέο feature σε ένα ομαδικό project.
  - `git pull` → «Τραβάμε» τις τελευταίες αλλαγές
  - `git checkout -b feature_a` → Δημιουργία νέου Branch
  - ...
  - `git add`
  - `git commit -m "..."`
  - `git push origin feature_a`
  - \*Δημιουργία Pull Request στο GitHub\*

# Εναλλακτικές υπηρεσίες Git

- GitLab
  - Free and Open Source Repository Management System
  - Host your private Git Server!
  - Uses its own CI Mechanism.
- BitBucket
  - Made by Atlassian
  - Also supports Mercurial Repos.
  - Integration with JIRA, Crucible, Jenkins, Bamboo



GitLab



# Ευχαριστούμε

<https://www.facebook.com/unipi.sls/>

<https://github.com/SLSUnipi>

<https://linkedin.com/in/kgiannakis>