

Ανάπτυξη Web εφαρμογών με HTML/CSS/JavaScript/jQuery/AJAX και Bootstrap

Ανάπτυξη Web Εφαρμογών

Εισαγωγή στο Git

Αθ. Ανδρούτσος

Συστήματα Διαχείρισης Εκδόσεων Λογισμικού

Ανάπτυξη Web Εφαρμογών

- VCS (Version Control Systems)

- Καταγράφει τις αλλαγές σε ένα αρχείο ή ομάδες αρχείων, ώστε να μπορούμε στο μέλλον να επαναφέρουμε τα αρχεία σε μία πρότερη κατάσταση, να συγκρίνουμε τις αλλαγές, να δούμε ποιος έκανε τις αλλαγές και πότε, κλπ.
- Επομένως, αν γίνει κάτι που δεν θέλουμε ή χάσουμε ένα αρχείο, μπορούμε εύκολα να το επαναφέρουμε

Τοπικό Version Control

Ανάπτυξη Web Εφαρμογών

- Ο εκφυλισμός της διαδικασίας του Version Control θα μπορούσε να είναι να κάνουμε copy/paste τα αρχεία που θέλουμε σε ένα νέο φάκελο (πιθανώς με timestamp στο όνομα του φακέλου, π.χ. photos-16-4-2019)
- Κάτι τέτοιο όμως δεν είναι ασφαλές ως στρατηγική διαχείρισης λαθών, δεδομένου ότι μπορεί να συμβούν λάθη που να μην μπορούμε να επαναφέρουμε (π.χ. να γράψουμε από πάνω
- Θα ήταν χρήσιμο αν είχαμε μια τοπική απλή ΒΔ που να αποθηκεύει όλες τις αλλαγές στα αρχεία που θέλουμε να εποπτεύουμε

Centralized Συστήματα (1)

Ανάπτυξη Web Εφαρμογών

- Για να μπορέσουν τα μέλη μιας ομάδας να συνεργαστούν σε ένα project δημιουργήθηκαν Centralized Version Control Systems (CVCSs)
- Ένας κεντρικός server περιείχε όλα τα αρχεία και τις αλλαγές και τα μέλη των ομάδων έπρεπε να δουλεύουν στον server

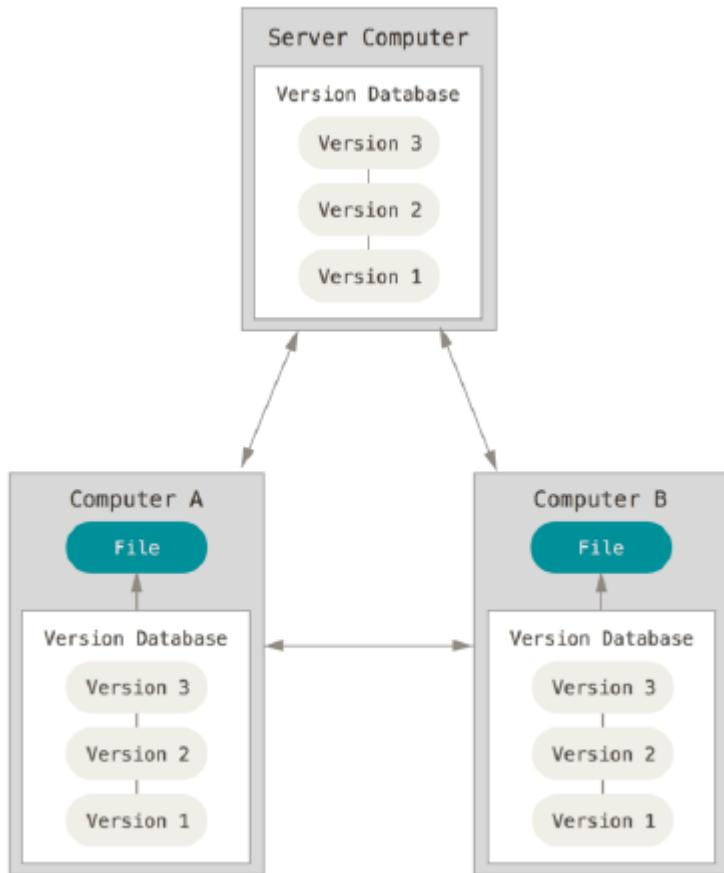
Centralized Συστήματα (2)

Ανάπτυξη Web Εφαρμογών

- **Οφέλη**
 - Sharing
 - Κεντρική διαχείριση δικαιωμάτων (ποιος κάνει τι)
- **Μειονεκτήματα**
 - Single Point of Failure
 - Το δίκτυο μπορεί να μην είναι διαθέσιμο
 - Όταν έχουμε όλο το history του project μας σε ένα μέρος και δεν παίρνουμε back-up κινδυνεύουμε να τα χάσουμε όλα

Κατανεμημένα (Distributed) Συστήματα

Ανάπτυξη Web Εφαρμογών



- Τα τοπικά PCs δεν έχουν απλώς την τελευταία έκδοση (latest snapshot) των αρχείων, αλλά έχουν (κάνουν mirror) όλο το repository και όλο το history
- Αν o server ‘χάσει’ τα αρχεία του Project κάθε μέλος της ομάδας μπορεί να τα επαναφέρει
- Άρα κάθε clone είναι ένα πλήρες backup του Project

Git (1)

Ανάπτυξη Web Εφαρμογών

- Σύστημα Διαχείρισης Εκδόσεων Λογισμικού (Version Management System)
- Μπορούμε να το εγκαταστήσουμε στον υπολογιστή μας και να το τρέχουμε από command line. Υπάρχει για όλα τα ΛΣ (Windows, Linux, Mac)
- Μπορούμε να το διαχειριστούμε και με δωρεάν account στο GitHub
- Απαραίτητο εργαλείο οργάνωσης και διαχείρισης του κώδικα

Git (2)

Ανάπτυξη Web Εφαρμογών

- Γιατί Git
 - Back-up των project μας
 - Portofolio-management
 - Version-Management
 - Συνεργασία ομάδων (Collaboration)
 - Ιστορικό (History)
 - Δυνατότητα αναίρεσης αλλαγών, μετάβαση σε παλαιότερες εκδόσεις
 - Εύκολη αναίρεση σε τυχόν λάθη

Git (3)

Ανάπτυξη Web Εφαρμογών

- Το Git είναι ένα κατανεμημένο VCS και εν αντιθέσει με τα υπόλοιπα κατανεμημένα συστήματα που κάνουν delta-based version control, το Git είναι ένα mini filesystem το οποίο αποθηκεύει references στα snapshots της κάθε έκδοσης (του κάθε commit) του κάθε αρχείου
- Δηλ. αποθηκεύει τα αρχεία και references προς τα αρχεία (και όχι τα delta changes)

Git (4)

Ανάπτυξη Web Εφαρμογών

- Κάθε άτομο της ομάδας μπορεί να δουλεύει σε διαφορετικό branch
- Μπορούμε να ενώσουμε όλα τα branches σχεδόν αυτόματα σε ένα ενιαίο branch (master branch)
- Τυπικά κάθε Project σε ένα Repository
- Σε ανοιχτά repositories ο καθένας μπορεί να βελτιώσει τον κώδικα (open source)

Git (5)

- Οι περισσότερες λειτουργίες του Git γίνονται σε ένα **τοπικό repository** (τοπικό φάκελο που περιέχει τα αρχεία που εποπτεύουμε)
- Όλο το history βρίσκεται τοπικά στο pc μας (στην local database) και όλες οι λειτουργίες γίνονται εξαιρετικά γρήγορα χωρίς τις καθυστερήσεις δικτύου που υπάρχουν στα centralized συστήματα
- Σε μεταγενέστερο χρόνο κάνουμε upload τις αλλαγές στον remote server (GitHub)

Ασφάλεια Δεδομένων

Ανάπτυξη Web Εφαρμογών

- Integrity - Secure Hash Algorithm 1 (**SHA-1**)
- Για κάθε αρχείο, με βάση τα περιεχόμενά του δημιουργείται ένα hash value -160-bit / 40 hex-digits- μοναδικό! Π.χ.
- 129fac190ea7e659129cd6190ebc1324feb66290
- .. αν και η google έδειξε πρόσφατα (2017) ότι ο SHA-1 δεν είναι 100% ασφαλής σε **collision attacks**, δηλ. διαφορετικά αρχεία με το ίδιο SHA-1 hash value
- To Git αποθηκεύει αυτό το hash value με κάθε snapshot των αρχείων

Τρεις βασικές καταστάσεις

Ανάπτυξη Web Εφαρμογών

- Τρεις καταστάσεις των αρχείων

1. **Committed** (stored στην DB – Git Directory / *.git directory -- repository*)
2. **Modified** (changed αλλά όχι committed / -- working directory, **working tree**)
3. **Staged** (tag ένα modified file για να γίνει commit στο επόμενο commit – **Staging Area**)

Μέρη ενός Git Project

Ανάπτυξη Web Εφαρμογών

- **Git Repository** (.git directory). Αυτό αντιγράφεται όταν κάνουμε clone
- **Working tree** – Μία version του project στην οποία δουλεύουμε
- **Staging area ή index** – Ένα αρχείο που στο .git directory που λειτουργεί ως buffer σχετικά με το τι πρόκειται να αποθηκευτεί στην DB στο επόμενο commit

Git Workflow

Ανάπτυξη Web Εφαρμογών

- Κάνουμε **modify** (add new files ή delete files) τα αρχεία μας στο working tree
- Κάνουμε **stage** εκείνα τα αρχεία που θέλουμε να πάνε με το επόμενο commit
- Κάνουμε **commit** – το git κοιτάει το staging area και παίρνει ένα snapshot των αρχείων και τα αποθηκεύει στο git directory

Ασφάλεια Git

Ανάπτυξη Web Εφαρμογών

- Αν τα αρχεία μας δεν έχουν γίνει commit, τότε μπορεί να χαθούν ή να αλλοιωθούν.
- Αν γίνουν commit, τότε δύσκολα χάνονται ιδιαίτερα αν έχουν γίνει push σε κάποιο εξωτερικό repository

Εγκατάσταση Git

Ανάπτυξη Web Εφαρμογών

- Το command line είναι **το μόνο μέρος που μπορούμε να τρέξουμε όλα τα git commands**
- Τα γραφικά περιβάλλοντα παρέχουν περιορισμένες δυνατότητες (δηλ. παρέχουν ένα υποσύνολο των git commands)

Εγκατάσταση Git (1)

Ανάπτυξη Web Εφαρμογών

- Linux
 - sudo apt-get update
 - sudo apt-get install git
 - git --version
- Windows
 - <https://git-scm.com/download>
 - <https://gitforwindows.org/>

Εγκατάσταση Git (2)

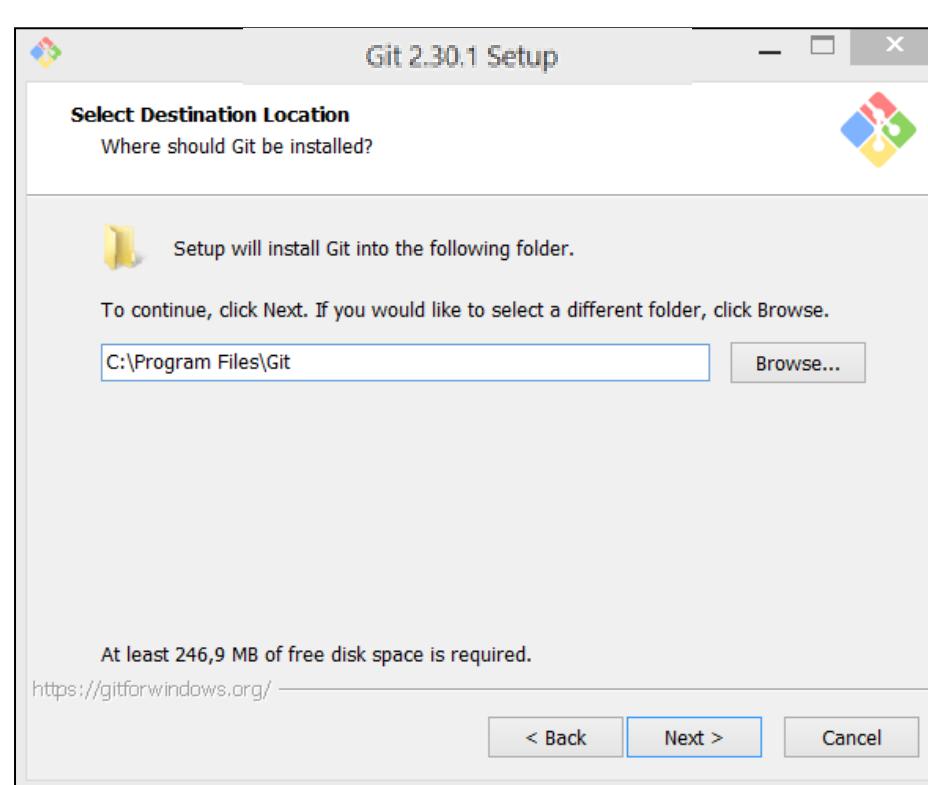
Ανάπτυξη Web Εφαρμογών

The screenshot shows the official Git download page at git-scm.com/download. The page features the Git logo and the tagline "local-branching-on-the-cheap". On the left, there's a sidebar with links for About, Documentation, Downloads (GUI Clients, Logos), and Community. The main content area has a "Downloads" section with icons for macOS, Windows, and Linux/Unix. Below this, a note says "Older releases are available and the Git source repository is on GitHub.". To the right, a large graphic of a computer monitor displays a teal-colored window titled "Latest source Release 2.30.1" with a "Release Notes (2021-02-08)" link and a prominent "Download 2.30.1 for Windows" button.

- Κατεβάζουμε την τελευταία έκδοση για το λειτουργικό σύστημα που έχουμε

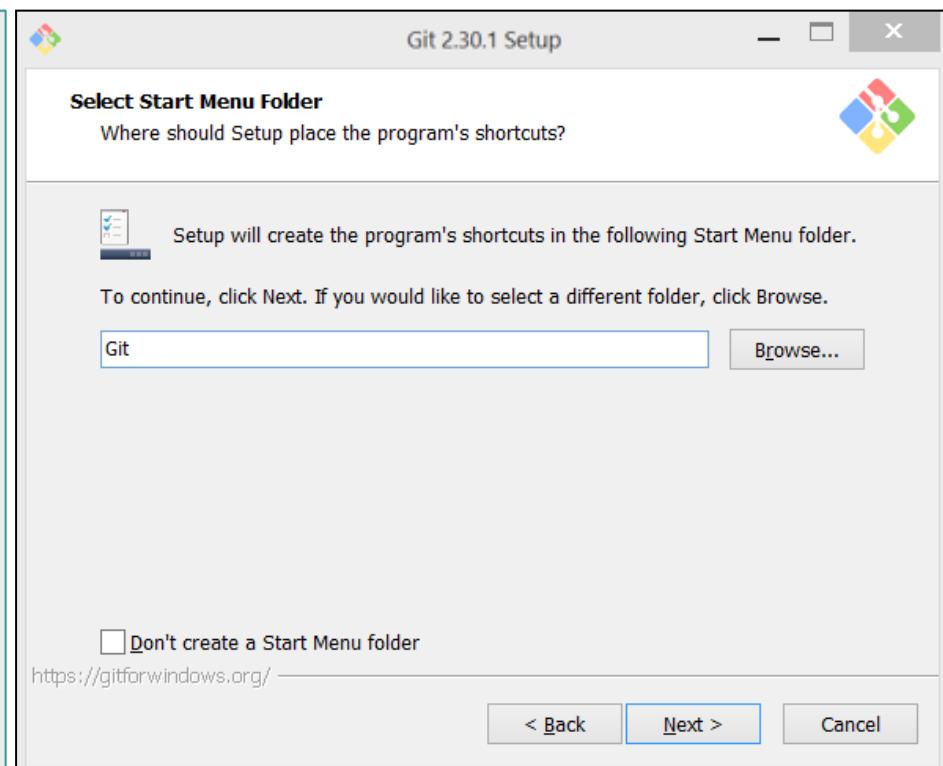
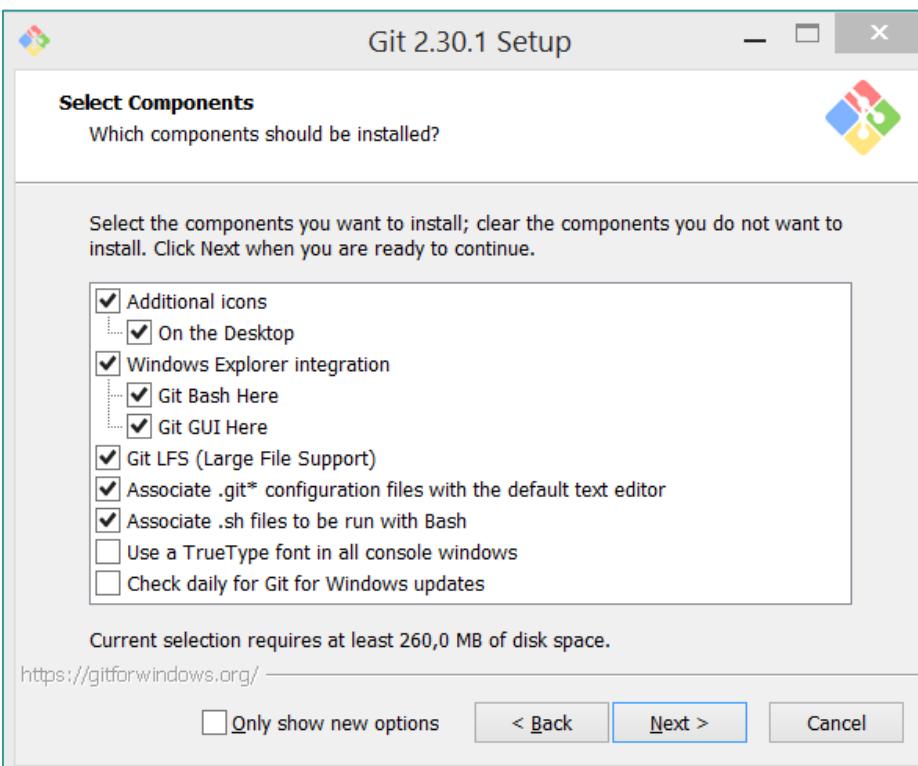
Εγκατάσταση Git Bash (1)

Ανάπτυξη Web Εφαρμογών



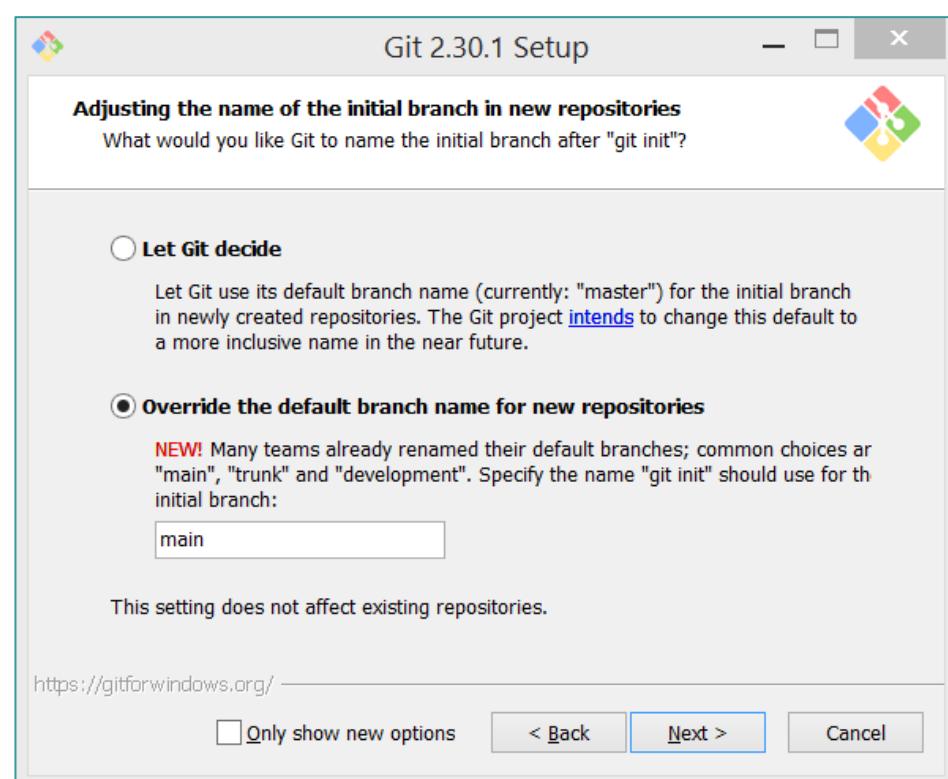
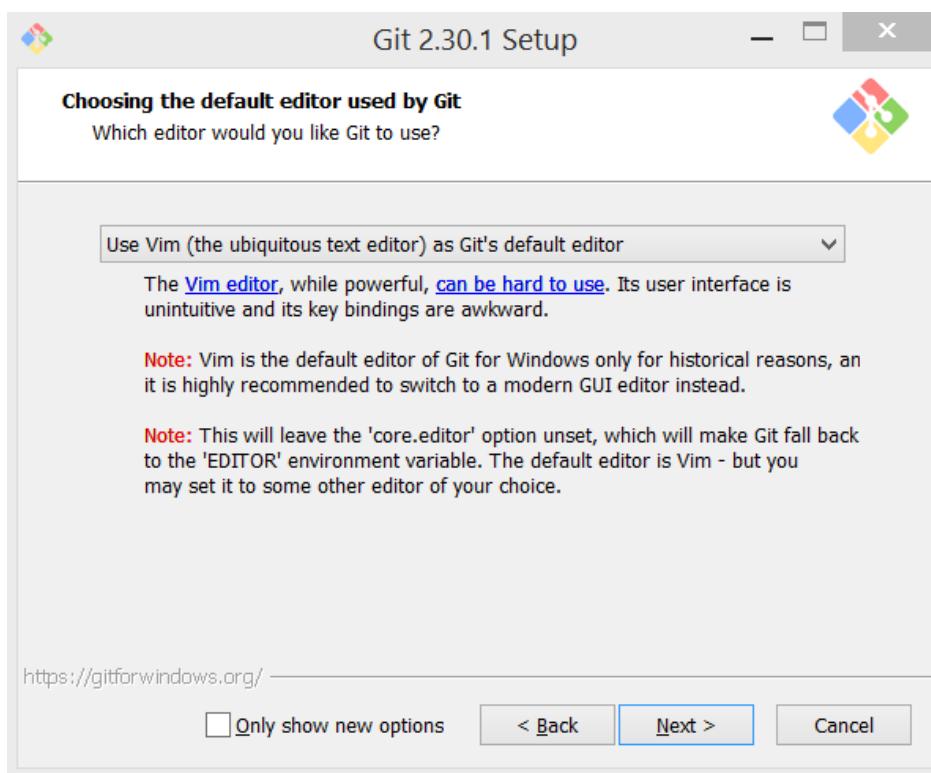
Εγκατάσταση Git Bash (2)

Ανάπτυξη Web Εφαρμογών



Εγκατάσταση Git Bash (3)

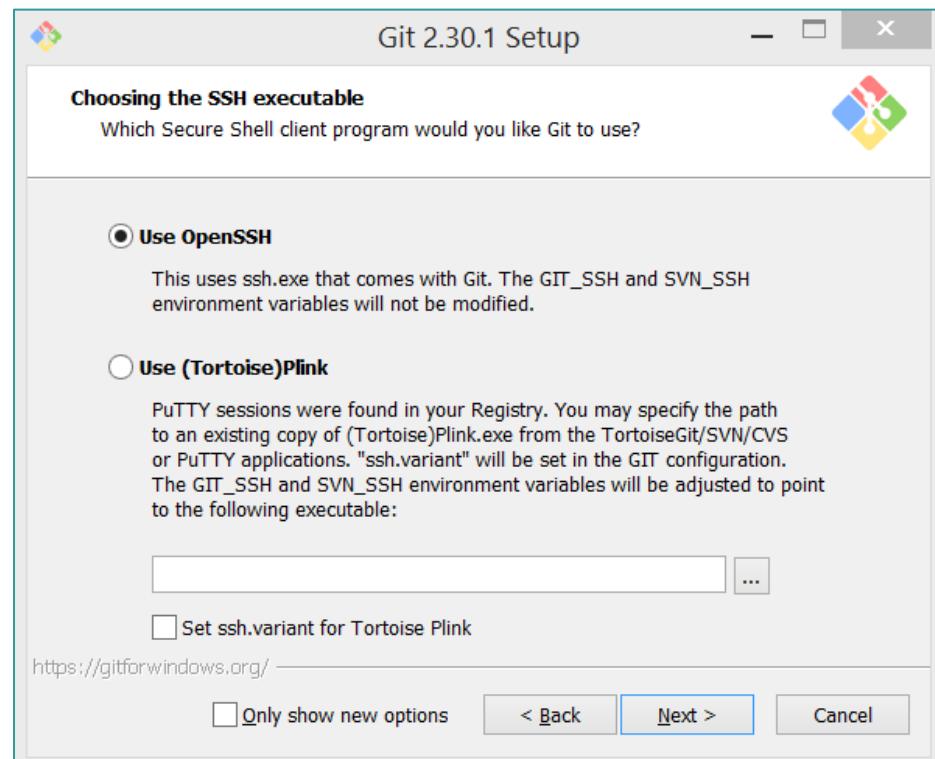
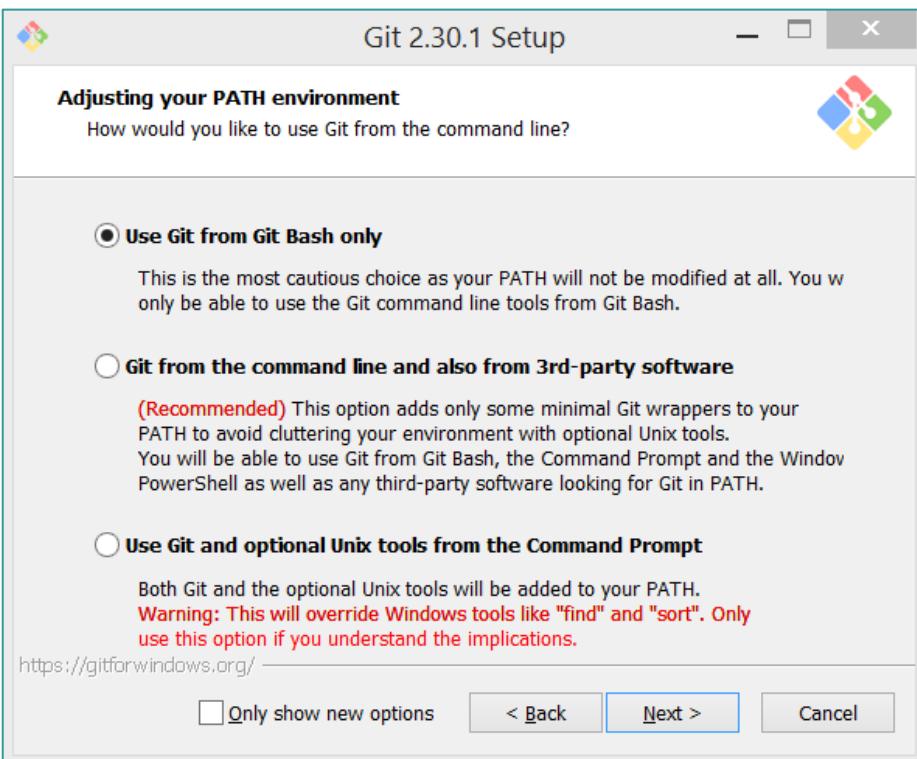
Ανάπτυξη Web Εφαρμογών



- Δεξιά επιλέξτε το **main** ως το default branch, δεδομένου ότι το GitHub από τον Οκτώβριο 2020 προτείνει όλα τα νέα repositories να χρησιμοποιούν το **main αντί του master branch** που ίσχυε τα προηγούμενα χρόνια

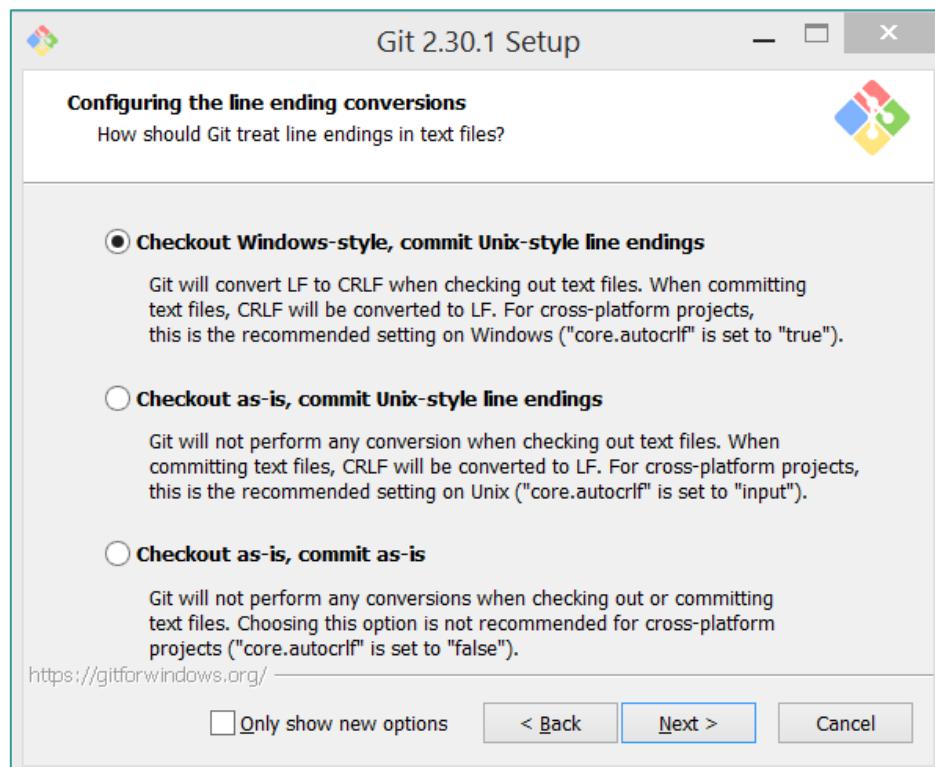
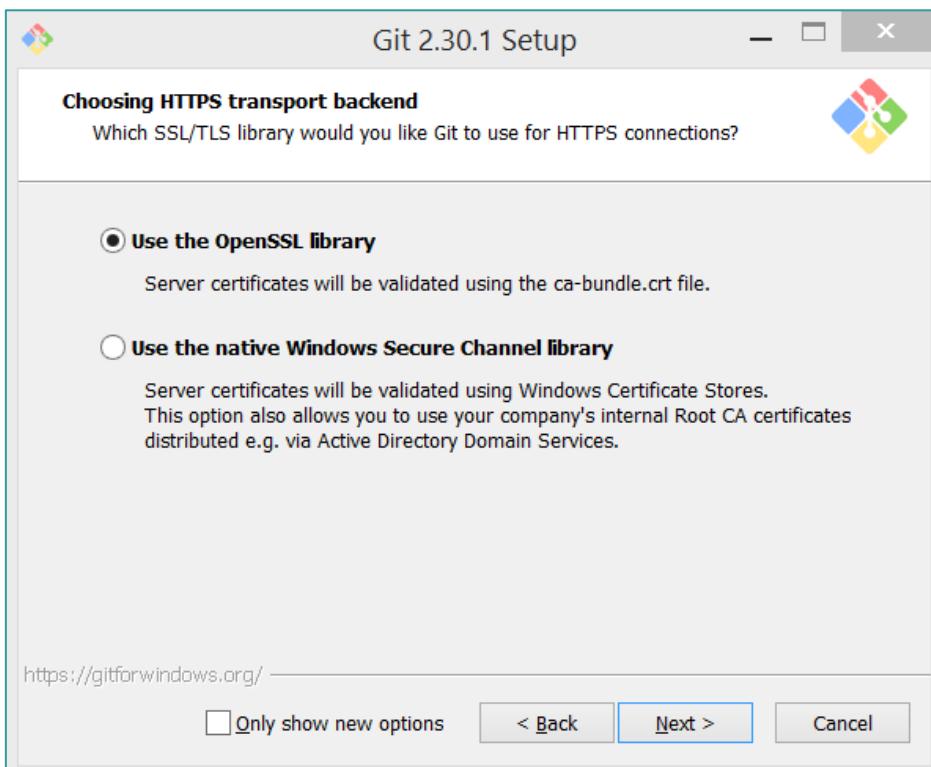
Εγκατάσταση Git Bash (4)

Ανάπτυξη Web Εφαρμογών



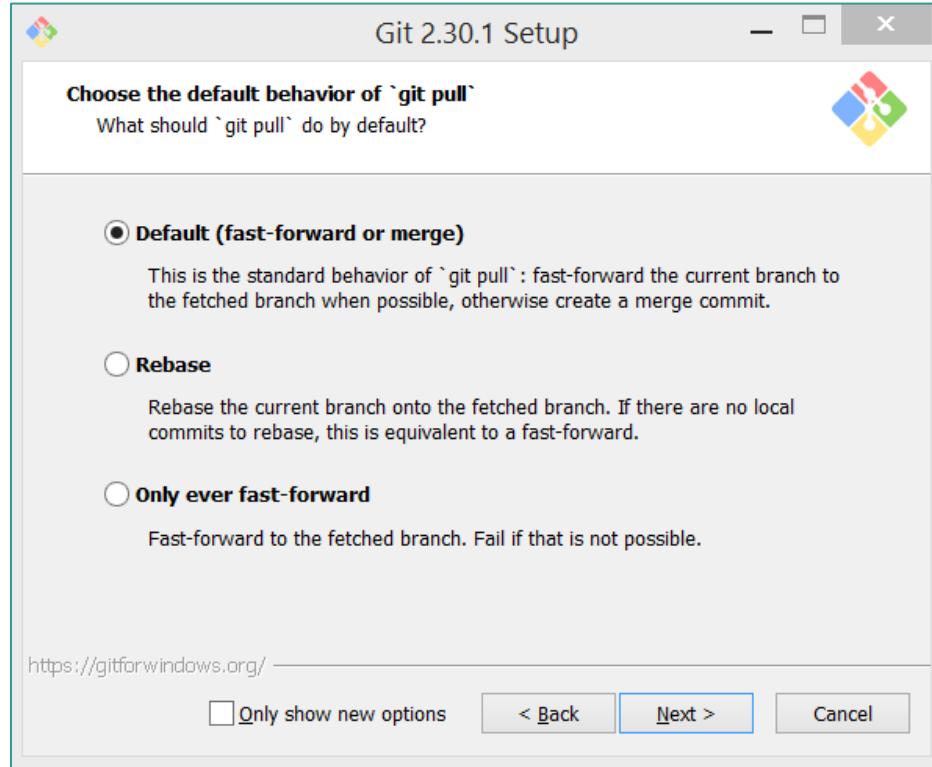
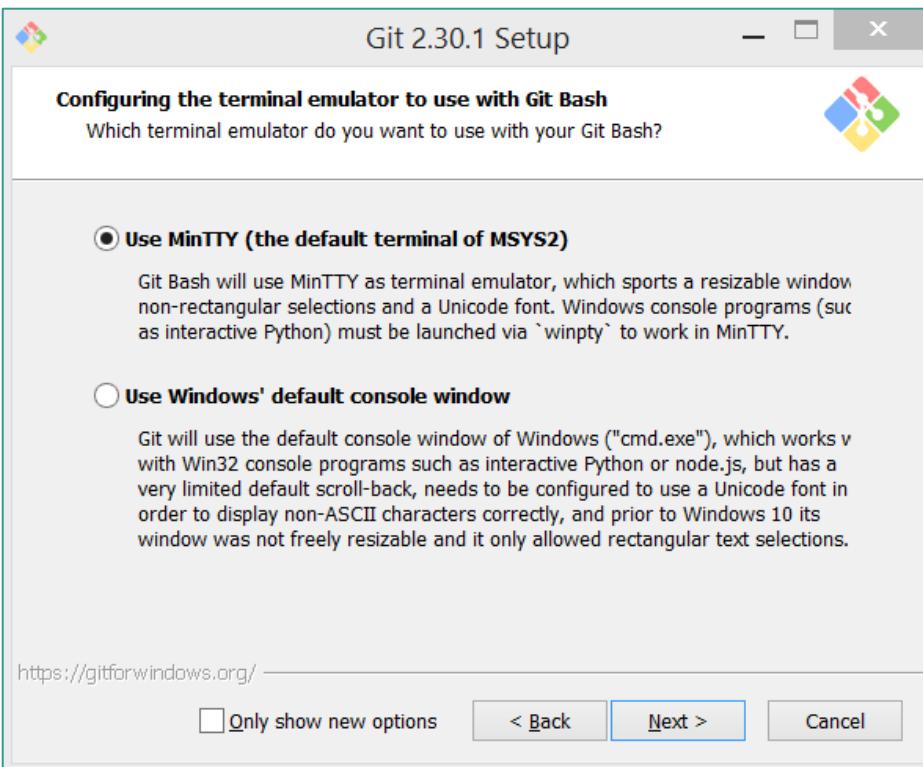
Εγκατάσταση Git Bash (5)

Ανάπτυξη Web Εφαρμογών



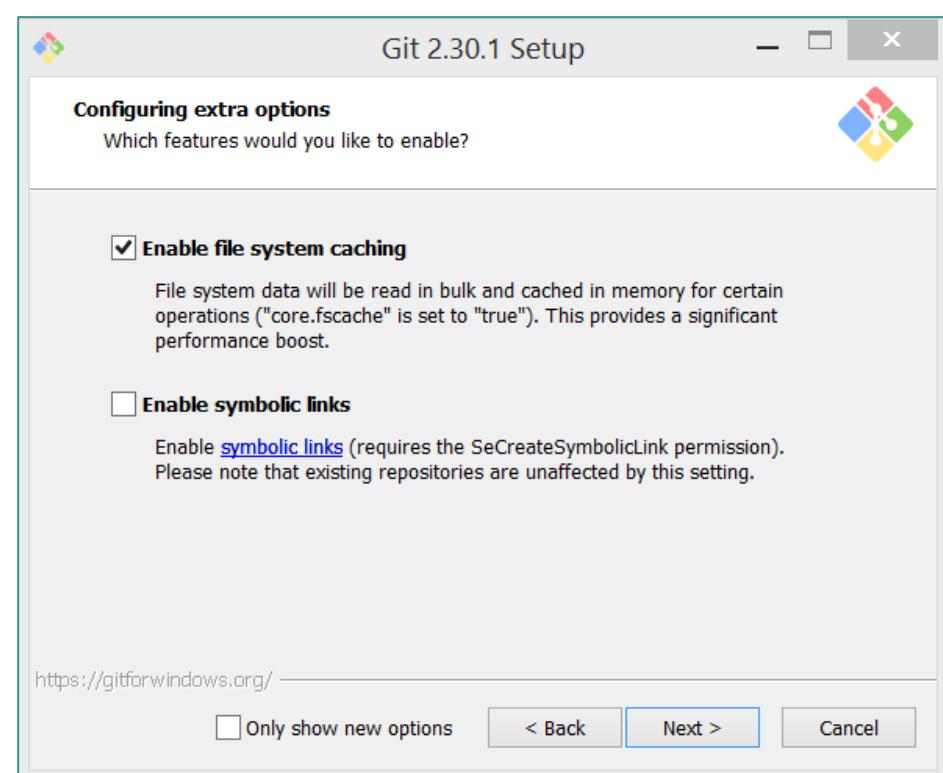
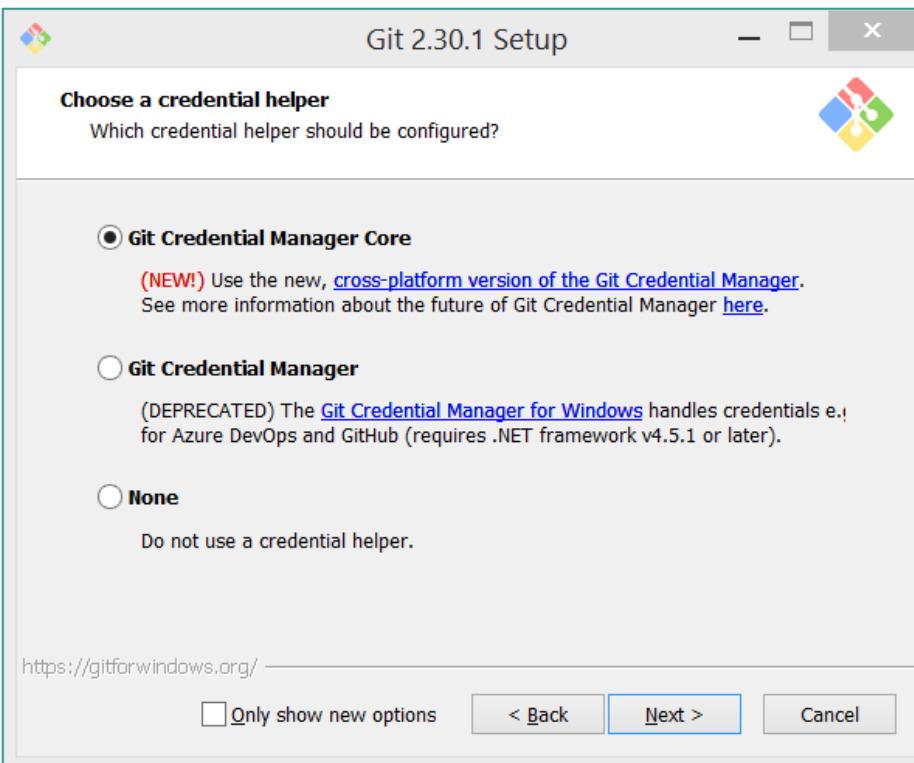
Εγκατάσταση Git Bash (6)

Ανάπτυξη Web Εφαρμογών



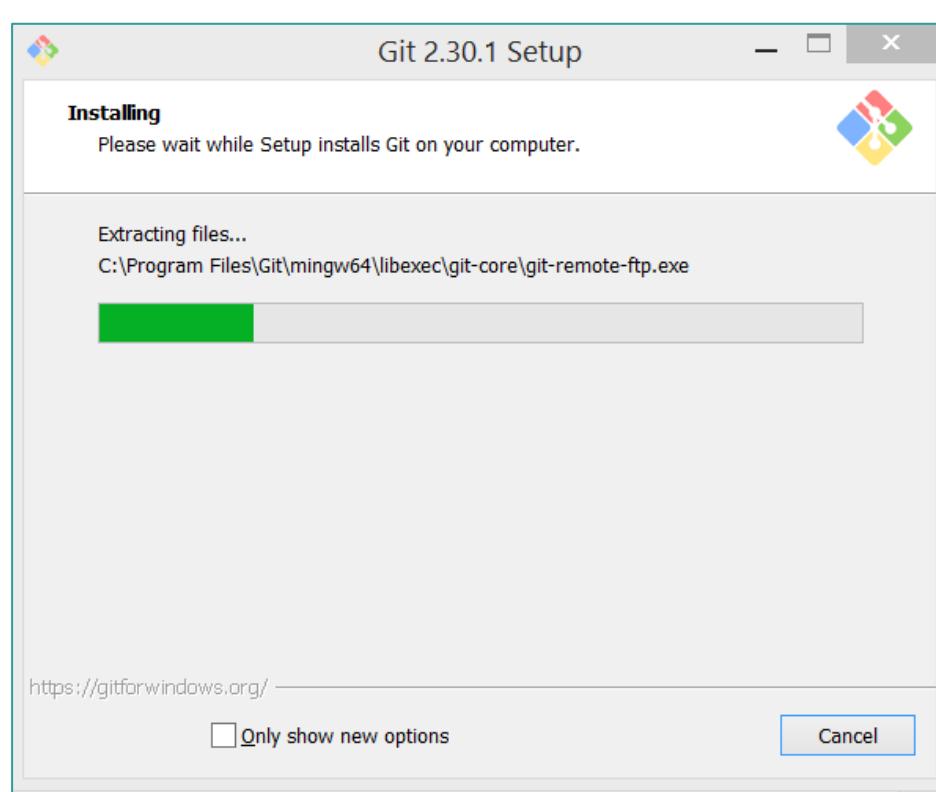
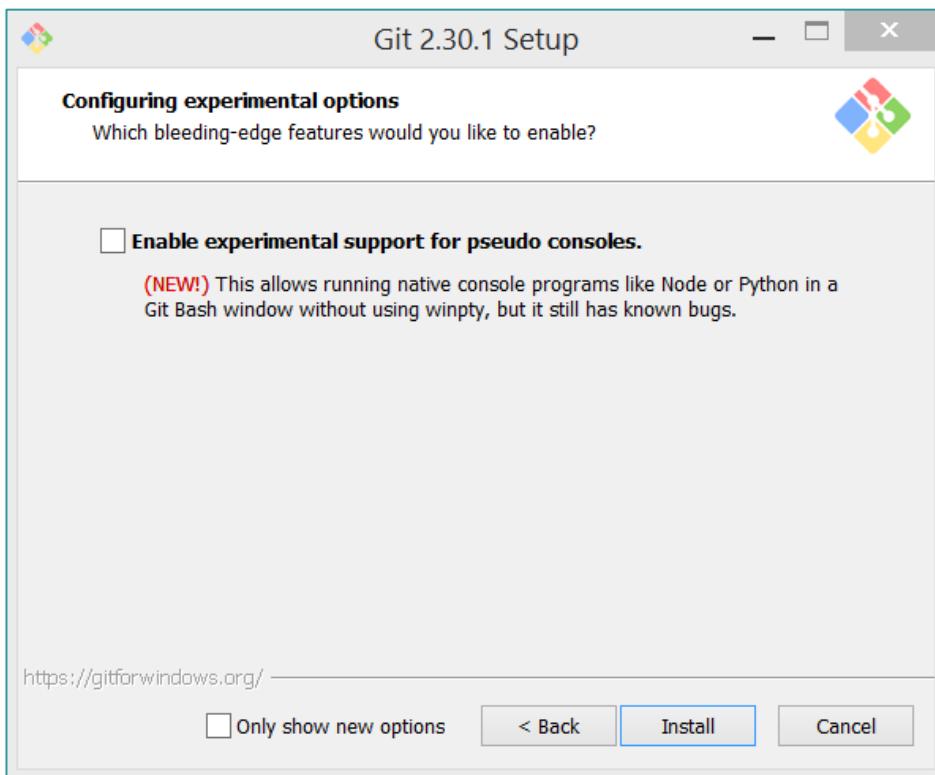
Εγκατάσταση Git Bash (7)

Ανάπτυξη Web Εφαρμογών



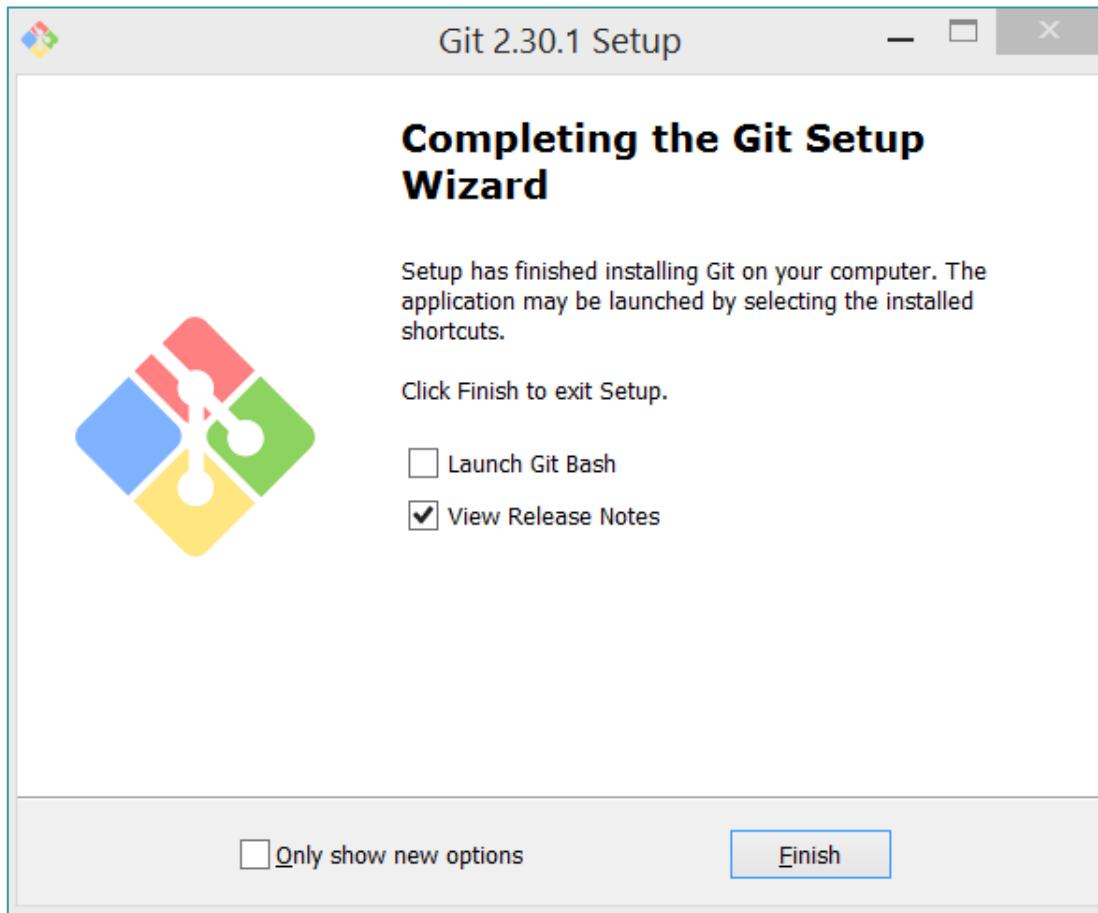
Εγκατάσταση Git Bash (8)

Ανάπτυξη Web Εφαρμογών



Εγκατάσταση Git Bash (9)

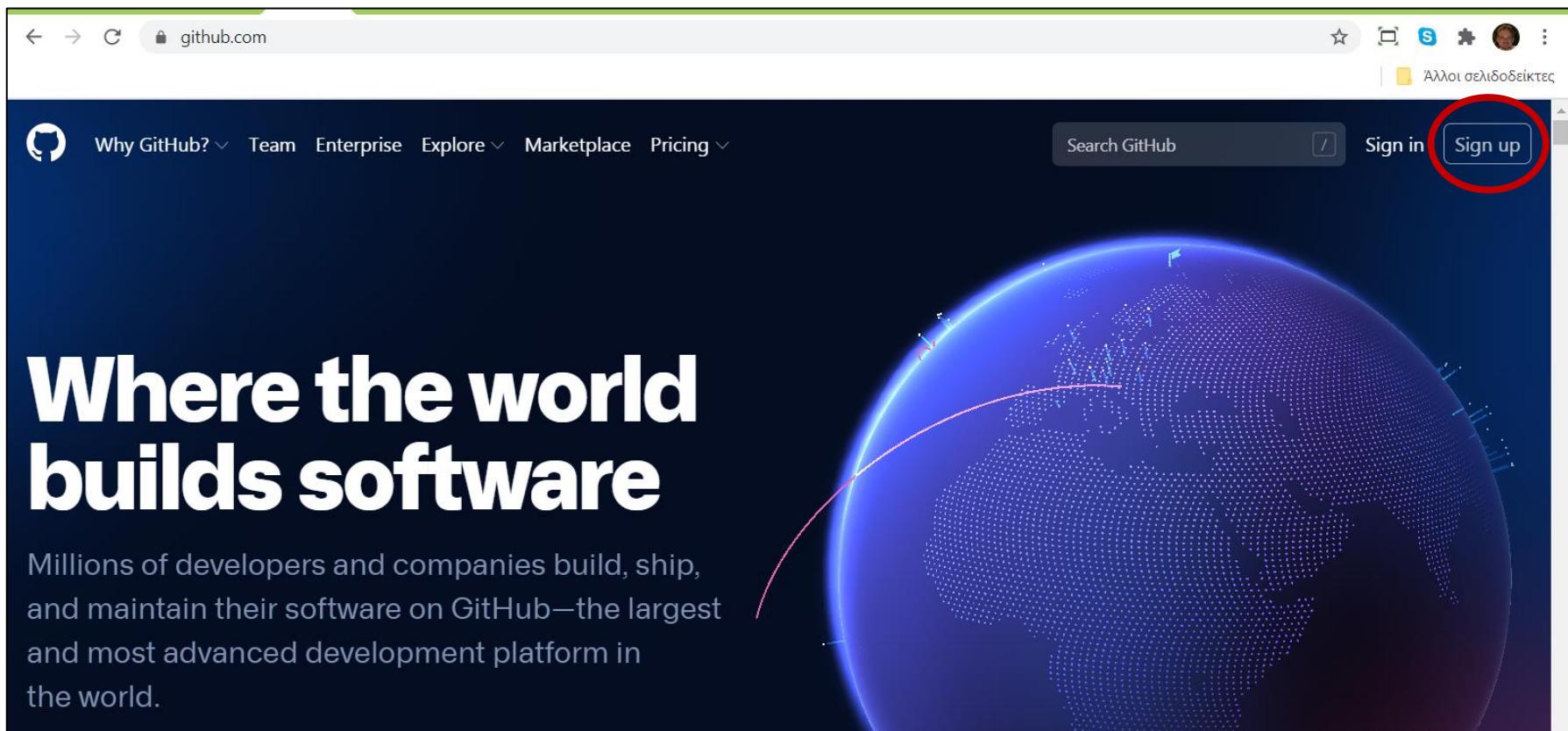
Ανάπτυξη Web Εφαρμογών



GitHub

Ανάπτυξη Web Εφαρμογών

- Τώρα ας πάμε στο GitHub (<https://github.com>) για να δημιουργήσουμε ένα λογαριασμό. Επιλέγουμε **Sign up**



Λογαριασμός χρήστη στο GitHub

Ανάπτυξη Web Εφαρμογών

Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences

Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we know you are a real person

[Verify](#)

[Create account](#)

- Στο sign up δίνετε τα στοιχεία που χρειάζεται, μετά Verify και Create Account

Δημιουργία προφίλ χρήστη

Ανάπτυξη Web Εφαρμογών

The screenshot shows the GitHub settings profile page for a user named Athanassios Androutsos. On the left, a sidebar lists account settings: Account settings (selected), Profile (highlighted with a red border), Account, Appearance (marked as New), Account security, Billing & plans, Security log, Security & analysis, Emails, Notifications, and SSH and GPG keys. The main content area is titled 'Public profile'. It includes fields for 'Name' (Athanassios Androutsos) and 'Profile picture' (a circular photo of a man speaking into a microphone). Below these are sections for 'Public email' (a8anassis@gmail.com) and 'Bio' (PhD in Informatics. Research fields: Software Engineering, Educational Technologies, Networking and Economics.). A note at the bottom says 'You can @mention other users and organizations to link to them.'

- Πάνω δεξιά επιλέγετε το avatar με το βελάκι και στο μενού **Settings**
- Εισάγετε στο **προφίλ** σας μία φωτογραφία, το ονοματεπώνυμό σας και το *e-mail* σας και πιθανώς και άλλα στοιχεία που προσδιορίζουν την επαγγελματική σας ταυτότητα

Προβολή Προφίλ

Ανάπτυξη Web Εφαρμογών



Athanassios
Androutsos
a8anassis

PhD in Informatics. Research fields:
Software Engineering, Educational
Technologies, Networking and
Economics.

- Athens University of Economics and Bu..
- Athens, Greece
- a8anassis@gmail.com
- <https://aueb.gr/>
- @a8anassis

- Πάλι από το avatar δεξιά επιλέγετε “Your profile”
 - Όνομα
 - Username
 - E-mail

Αρχικοποίηση Git (1)

Ανάπτυξη Web Εφαρμογών

- Αφού δημιουργήσουμε το λογαριασμό μας στο GitHub, ξαναπάμε στο Git Bash και εισάγουμε:
 - **username**, μπορεί να είναι το ίδιο ή διαφορετικό από αυτό που έχουμε στο GitHub, Μπορεί εδώ να είναι το όνομά μας, για παράδειγμα "Th. Androutsos"
 - **e-mail**, το ίδιο με αυτό που έχουμε στο GitHub, ώστε να μπορούμε να συνδέσουμε το GitHub με το τοπικό μας Git
- Για εισαγωγή username, δίνουμε:
 - **git config --global user.name "username"**
π.χ. **git config --global user.name "Th. Androutsos"**
- Για επιβεβαίωση, δίνουμε:
 - **git config --global user.name**

Αρχικοποίηση Git (2)

Ανάπτυξη Web Εφαρμογών

- Κάνουμε το ίδιο για το e-mail
 - `git config --global user.email "email@example.com"`
π.χ. `git config --global user.email "a8anassis@gmail.com"`
- Για επιβεβαίωση
 - `git config --global user.email`

Αρχικοποίηση Git (3)

Ανάπτυξη Web Εφαρμογών

Δίνουμε τις εντολές στο Git Bash:

```
MINGW64:/c/Users/thanos
thanos@thanasis MINGW64 ~
$ git config --global user.name "a8anassis"

thanos@thanasis MINGW64 ~
$ git config --global user.name
a8anassis

thanos@thanasis MINGW64 ~
$ git config --global user.email "a8anassis@gmail.com"

thanos@thanasis MINGW64 ~
$ git config --global user.name "Th. Androutsos"

thanos@thanasis MINGW64 ~
$ git config --global user.name
Th. Androutsos

thanos@thanasis MINGW64 ~
$ |
```

- Ως username μπορούμε να εισάγουμε είτε το username του GitHub, είτε το όνομά μας. Αυτό βασικά θα φαίνεται στα commits που κάνουμε στο Git και στο GitHub
- Ως e-mail πρέπει να δώσουμε το ίδιο που έχουμε δώσει στο GitHub

Repository

Ανάπτυξη Web Εφαρμογών

- Η βασική δομή στο Git/GitHub είναι το repository
- Ένα repository αντιστοιχεί σε ένα project
- Τεχνικά, το repository είναι σαν ένας φάκελος των Windows ή ένα directory στα Linux/Mac

Δημιουργία repository (1)

Ανάπτυξη Web Εφαρμογών

- Με δύο (2) τρόπους:

1. Αν θέλουμε ένα τοπικό μας φάκελο (local folder/directory) να τον μετατρέψουμε σε Git folder/directory κάνουμε **git init**
 - Στη συνέχεια δημιουργούμε ένα repository στο GitHub και **συνδέουμε το τοπικό μας repo με το repo στο GitHub**, ώστε να μεταφέρουμε (ανεβάζουμε/κατεβάζουμε – push/pull) αρχεία και φακέλους

Δημιουργία repository (2)

Ανάπτυξη Web Εφαρμογών

2. Ο δεύτερος τρόπος είναι να αντιγράψουμε (clone - σαν να κάνουμε copy/paste) ένα ήδη υπάρχον repository του GitHub (πιθανώς κάποιου άλλου χρήστη ή μέλους της ομάδας μας)
 - Αν θέλουμε να αντιγράψουμε ένα repository από το GitHub και να το κάνουμε cloning στον τοπικό μας δίσκο, τότε κάνουμε **git clone**

Μετατροπή φάκέλου σε repo -- git init

Ανάπτυξη Web Εφαρμογών

The screenshot shows a terminal window titled 'MINGW64:/c/Users/thanos/git/codingFactory'. The terminal output is as follows:

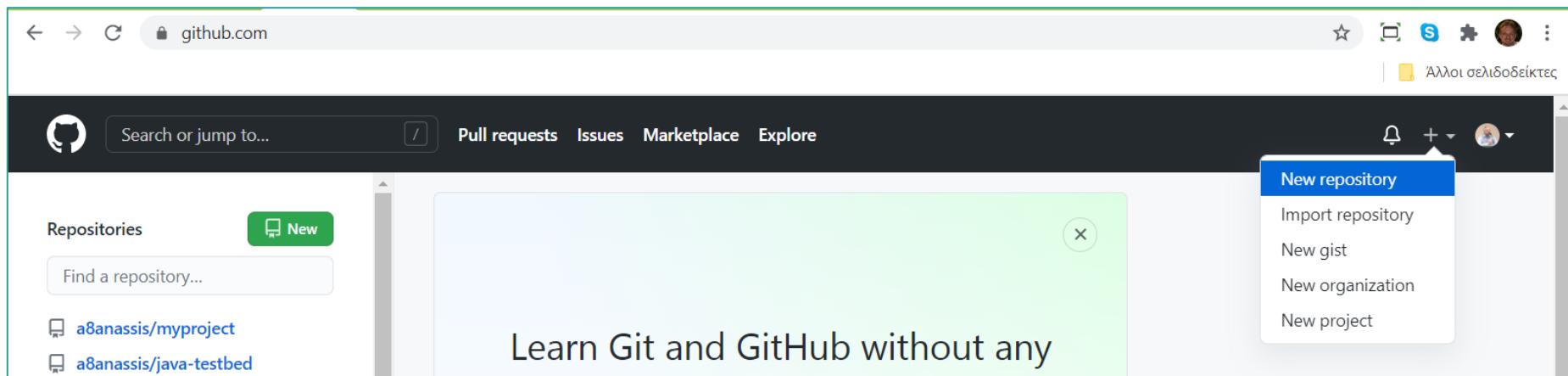
```
$ ls
codingFactory/  java-elearn/
thanos@thanasis MINGW64 ~/git
$ ls -la
total 68
drwxr-xr-x 1 thanos 197609 0 Απρ 17 12:03 .
drwxr-xr-x 1 thanos 197609 0 Απρ 14 15:25 ../
drwxr-xr-x 1 thanos 197609 0 Απρ 17 12:03 codingFactory/
drwxr-xr-x 1 thanos 197609 0 Μαρ 28 23:03 java-elearn/
thanos@thanasis MINGW64 ~/git
$ cd codingFactory/
thanos@thanasis MINGW64 ~/git/codingFactory
$ pwd
/c/Users/thanos/git/codingFactory
thanos@thanasis MINGW64 ~/git/codingFactory
$ git init
Initialized empty Git repository in C:/Users/thanos/git/codingFactory/.git/
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ |
```

The last command, `git init`, is highlighted with a red rectangle.

- Έχουμε δημιουργήσει ένα φάκελο git και μέσα στον git, δημιουργούμε υποφάκελο με όνομα έστω *codingFactory*
- Πάμε στον φάκελο με `cd`
- Μέσα στον *codingFactory*' κάνουμε: `git init`

GitHub – Νέο Repository (1)

Ανάπτυξη Web Εφαρμογών



- Στη συνέχεια, πάμε στο GitHub και δημιουργούμε νέο repository με όνομα `codingFactory` (δεν είναι όμως απαραίτητο τα ονόματα των repositories να είναι τα ίδια στο GitHub και στο τοπικό μας repo)

GitHub – Νέο Repository (2)

Ανάπτυξη Web Εφαρμογών

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * Repository name *

 a8anassis / 

Great repository names are **codingfactory** is available. Need inspiration? How about [redesigned-fortnight](#)?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository

- Δίνουμε το **όνομα** του repository, μία προαιρετική περιγραφή, αφήνουμε **public repository** εφόσον θέλουμε να έχουν πρόσβαση όλοι, αλλιώς **private** και επίσης επιλέγουμε Add a README file, ένα αρχείο κειμένου που συνηθίζουμε να περιλαμβάνουμε σε κάθε project με μία περιγραφή του project
- Στο τέλος αναφέρεται ως **default branch** το **main**. Δίνεται η δυνατότητα να αλλάξουμε το όνομα του **default branch**

GitHub Repository

Ανάπτυξη Web Εφαρμογών

The screenshot shows a GitHub repository page. At the top, there's a banner with the text "Learn Git and GitHub without any code!" and a "Read the guide" button. Below the banner, the repository name "a8anassis / codingFactory" is displayed, along with metrics: 1 unwatched, 0 stars, and 0 forks. The "Code" tab is selected. In the main area, a commit from "a8anassis" titled "Initial commit" is shown, dated "5bc6b52 on 17 Apr 2019". A file named "README.md" is listed with the commit message "Initial commit" and a timestamp of "2 years ago". On the right side, there's an "About" section with the note "No description, website, or topics provided".

- Έτσι δημιουργήσαμε το remote repository στο GitHub. Στο παράδειγμα έχω αλλάξει το όνομα του *main branch* σε *master*, για ιστορικούς λόγους. Ωστόσο οι εκπαιδευόμενοι δεν χρειάζεται να κάνουν κάτι τέτοιο, οπότε αντί για *master branch* θα εμφανίζεται το *main branch* (δείτε επόμενη διαφάνεια)

Main vs master branch

Ανάπτυξη Web Εφαρμογών

- Τα branches είναι ανεξάρτητα μονοπάτια, που περιέχουν τα αρχεία του repository αλλά όπου μπορούμε να κάνουμε αλλαγές που δεν επηρεάζουν τα άλλα branches
- Για πολλά χρόνια πριν τον Οκτώβριο 2020 το βασικό default branch του GitHub ήταν το master. Από τον Οκτώβριο 2020 το όνομα άλλαξε σε main και κάθε νέο repository που δημιουργούμε στο GitHub έχει ως default branch το main
- Εμείς μπορούμε να αλλάξουμε το όνομα του default branch από main σε master όπως ήταν παλιά αλλά κάτι τέτοιο δεν συνίσταται πλέον

Αλλαγή default branch name

Ανάπτυξη Web Εφαρμογών

The screenshot shows a browser window for GitHub's account settings. The URL is `github.com/settings/repositories`. The main content area is titled "Repository default branch". It contains a sub-section titled "Choose the default branch for your new personal repositories. You might want to change the default name due to different workflows, or because your integrations still require "master" as the default branch name. You can always change the default branch name on individual repositories. [Learn more about default branches.](#)" Below this is a text input field containing "main" and a blue "Update" button. To the left of the main content is a sidebar with the following items: "Account settings" (selected), "Profile", "Account", "Appearance" (with a "New" badge), "Account security", "Billing & plans", and "Security log". At the top of the page, there is a navigation bar with links for "Pull requests", "Issues", "Marketplace", and "Explore". The GitHub logo is at the top left, and the user profile "Athanassios Androutsos" is at the top right. A red box highlights the "main" text input field.

- Αν θέλαμε να αλλάξουμε το όνομα του default branch, στα **settings/repositories** μπορούμε να αλλάξουμε το main σε master. Αυτό μπορεί να το κάνουν όσοι έχουν συνηθίσει να δουλεύουν με το master branch αλλά σταδιακά όλοι θα πρέπει να μετακινηθούν όλοι στο main branch

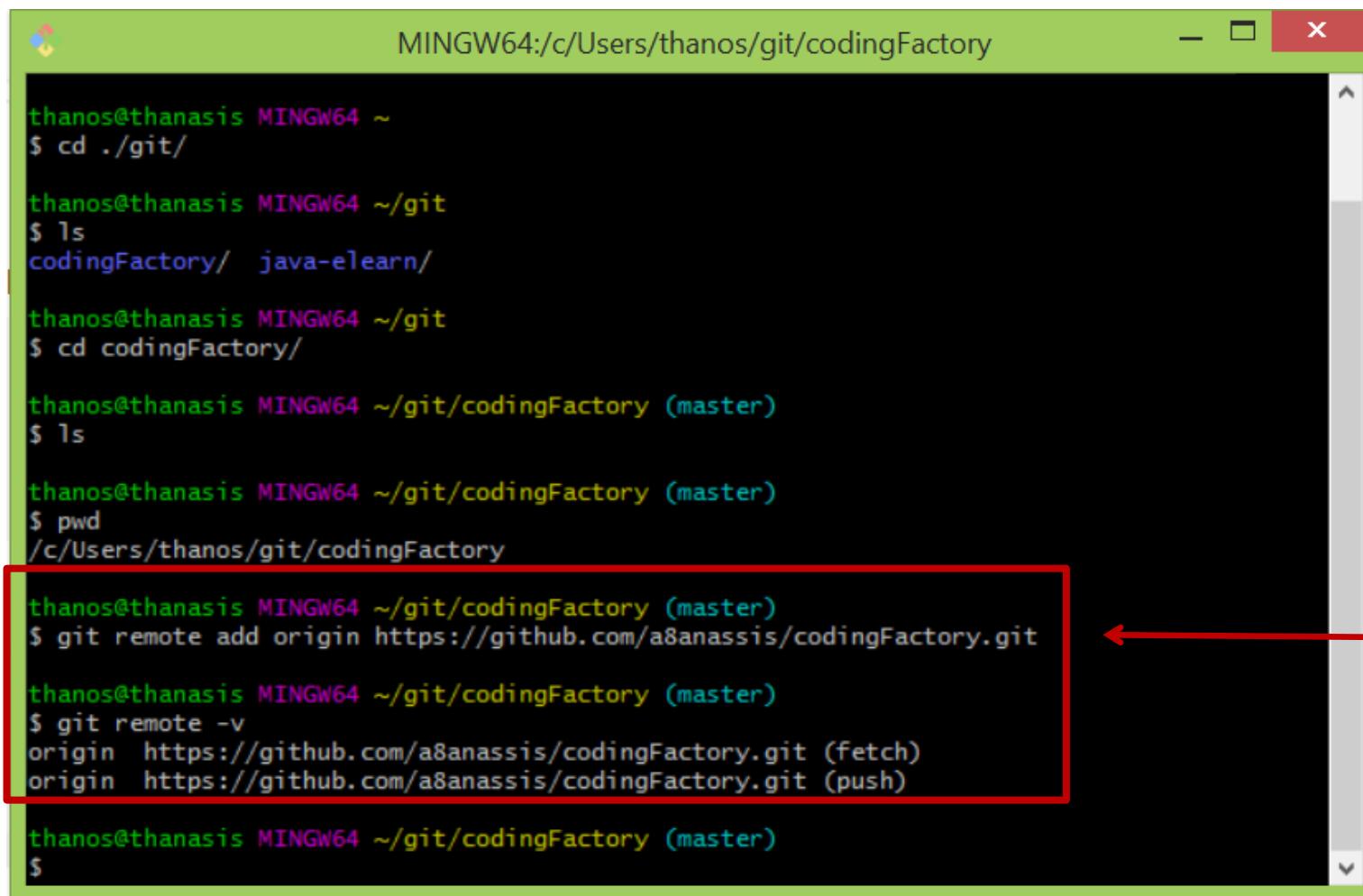
Παραδείγματα master vs main

Ανάπτυξη Web Εφαρμογών

- Στα παραδείγματα των σημειώσεων και των βίντεο χρησιμοποιούμε το master branch αλλά θα πρότεινα οι εκπαιδευόμενοι, αν δεν υπάρχει κάποιος ειδικός λόγος, να χρησιμοποιούν πλέον το main branch
- Σε αυτή την περίπτωση, στα παραδείγματα που ακολουθούν όπου βλέπετε τη λέξη master απλά την αντικαθιστάτε με τη λέξη main

Σύνδεση τοπικού με remote

Ανάπτυξη Web Εφαρμογών



```
MINGW64:/c/Users/thanos/git/codingFactory
thanos@thanasis MINGW64 ~
$ cd ./git/
thanos@thanasis MINGW64 ~/git
$ ls
codingFactory/  java-elearn/
thanos@thanasis MINGW64 ~/git
$ cd codingFactory/
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ ls
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ pwd
/c/Users/thanos/git/codingFactory
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git remote add origin https://github.com/a8anassis/codingFactory.git
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git remote -v
origin  https://github.com/a8anassis/codingFactory.git (fetch)
origin  https://github.com/a8anassis/codingFactory.git (push)
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$
```

Συνδέουμε το remote repository με το τοπικό με την git remote add δίνοντας ένα όνομα στη σύνδεση, συνήθως origin αλλά μπορεί να είναι και άλλο όνομα. Στη συνέχεια ακολουθεί το URL που αντιγράψαμε

- *git remote add <name> <URL>*

git pull origin master

Ανάπτυξη Web Εφαρμογών

- Η **git pull origin master** φέρνει τα περιεχόμενα του master branch από το GitHub. Εδώ τα περιεχόμενα είναι μόνο το README.md, το οποίο και αντιγράφεται από το origin/master στο τρέχον τοπικό branch. Μετά τον Οκτώβριο 2020 και εφόσον το default branch είναι το main, η εντολή γίνεται *git pull origin main*

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/a8anassis/codingFactory
 * branch            master      -> FETCH_HEAD
 * [new branch]      master      -> origin/master

thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ pwd
/c/Users/thanos/git/codingFactory

thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ ls
README.md

thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ |
```

cat <αρχείο>

Ανάπτυξη Web Εφαρμογών

- cat to README.md
- Με cat εμφανίζουμε τα περιεχόμενα του αρχείου

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ cat README.md
# codingFactory
thanos@thanasis MINGW64 ~/git/codingFactory (master)
```

Clone

Ανάπτυξη Web Εφαρμογών

- Ο δεύτερος τρόπος δημιουργίας repository είναι να αντιγράψουμε (Clone) ένα repository άλλου χρήστη από το GitHub
- Στο παράδειγμα που ακολουθεί έχουμε δημιουργήσει ένα φάκελο, έστω git και μέσα στον git θα κάνουμε τις αντιγραφές (git clone) όλων των repositories που θέλουμε

Δημιουργία φακέλου εργασίας

Ανάπτυξη Web Εφαρμογών

- Δημιουργούμε λοιπόν ένα γενικό φάκελο git μέσα στον οποίο θα κάνουμε clone repositories από το GitHub
 - *cd /users/thanos* σε Win ή *cd /home/username* σε Linux
- Δημιουργία φακέλου git μέσα στο home dir (*/users/username* σε Win ή */home/username* σε Linux)
 - *mkdir git*

```
thanosis@thanasis MINGW64 ~
$ pwd
/c/Users/thanos

thanosis@thanasis MINGW64 ~
$ pwd
/c/Users/thanos

thanosis@thanasis MINGW64 ~
$ mkdir git

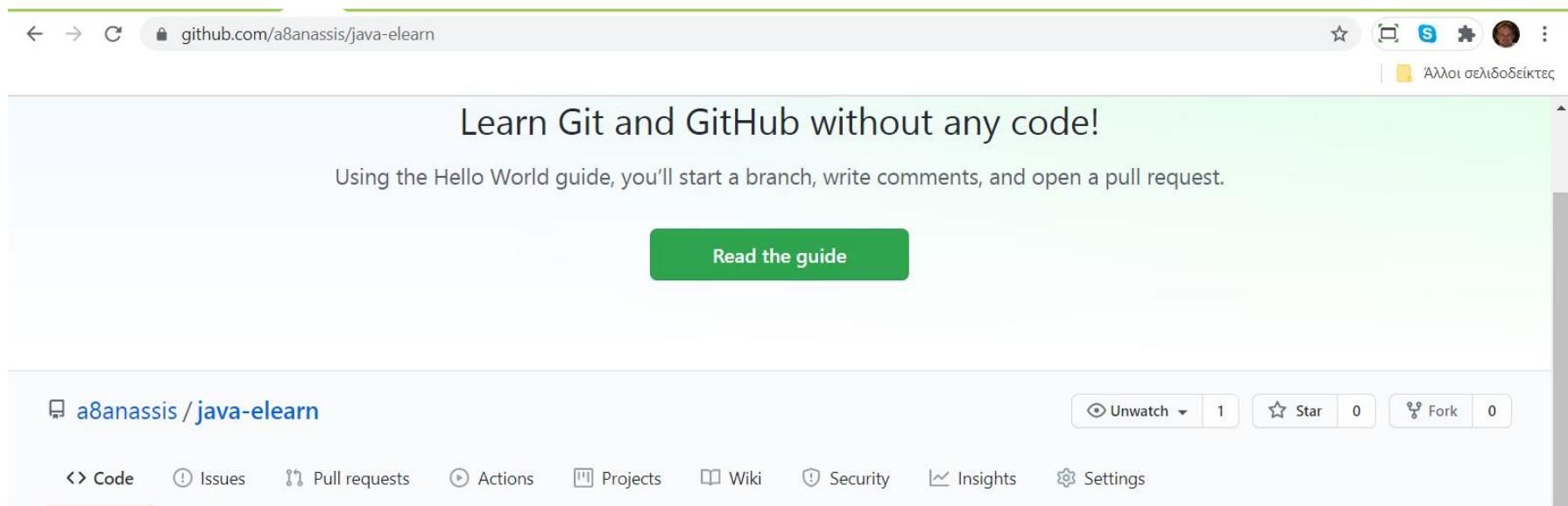
thanosis@thanasis MINGW64 ~
$ cd git/

thanosis@thanasis MINGW64 ~/git
$
```

Αντιγραφή URL (1)

Ανάπτυξη Web Εφαρμογών

- Όπως πριν, αφού πάμε στο repository που θέλουμε να αντιγράψουμε (εδώ έστω το java-elearn, αλλά μπορεί να είναι οποιοδήποτε repository) αντιγράφουμε το URL



Αντιγραφή URL (2)

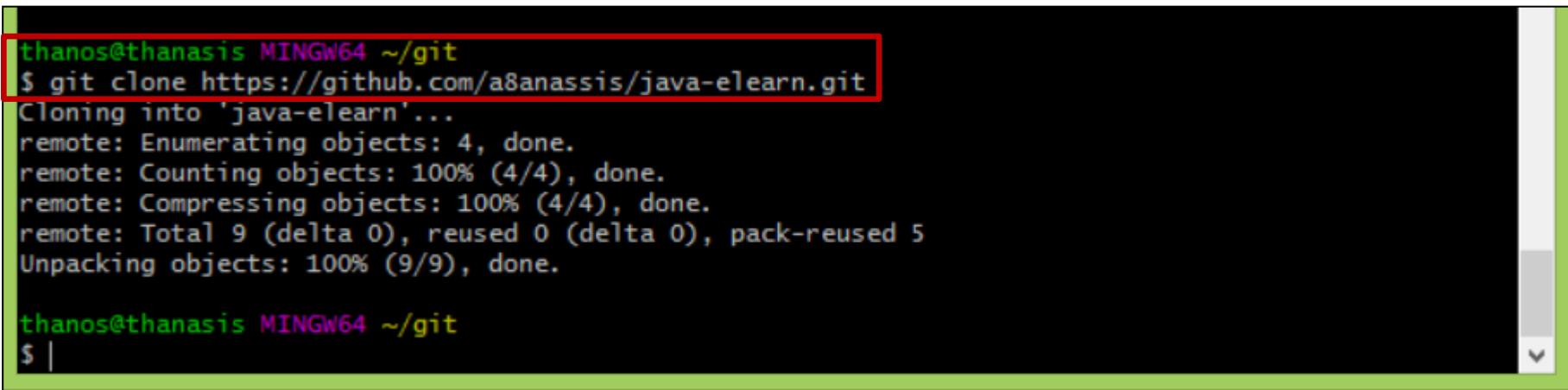
Ανάπτυξη Web Εφαρμογών

The screenshot shows a GitHub repository page for 'a8anassis/java-elearn'. The main heading says 'Learn Git and GitHub without any code!' with a 'Read the guide' button. Below the header, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is currently selected. On the left, there's a list of files: '.gitignore', 'ByteShortInt.java', and 'HelloWorld.java'. On the right, there's an 'About' section with details like 'Java e-learn', 'Readme', and 'GPL-3.0 License'. A modal window is open over the repository content, showing the 'Code' dropdown with options for 'Clone' via HTTPS, SSH, or GitHub CLI. The 'HTTPS' option is highlighted, and its URL, 'https://github.com/a8anassis/java-elearn', is copied to the clipboard. The modal also contains the text 'Use Git or checkout with SVN using the web URL.'

- Αντιγράφουμε τον σύνδεσμο

Δημιουργία local repository μέσω αντιγραφής

Ανάπτυξη Web Εφαρμογών

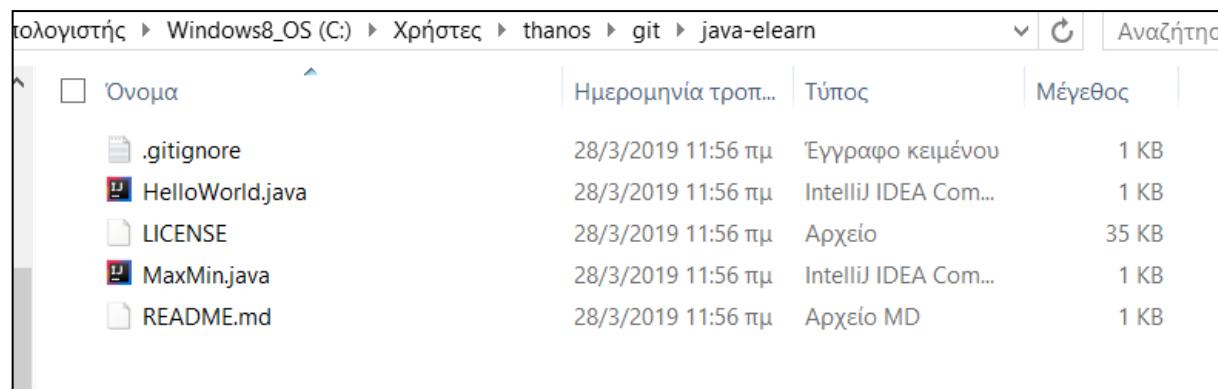


```
thanos@thanasis MINGW64 ~/git
$ git clone https://github.com/a8anassis/java-elearn.git
Cloning into 'java-elearn'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 5
Unpacking objects: 100% (9/9), done.

thanos@thanasis MINGW64 ~/git
$ |
```

Η εντολή είναι: **git clone <URL>**

Παρατηρούμε ότι δημιουργείται νέος git folder με όνομα java-elearn και περιεχόμενα τα περιεχόμενα του repo στο GitHub



Όνομα	Ημερομηνία τροπ... τροποποίησης	Τύπος	Μέγεθος
.gitignore	28/3/2019 11:56 πμ	Έγγραφο κειμένου	1 KB
HelloWorld.java	28/3/2019 11:56 πμ	IntelliJ IDEA Com...	1 KB
LICENSE	28/3/2019 11:56 πμ	Αρχείο	35 KB
MaxMin.java	28/3/2019 11:56 πμ	IntelliJ IDEA Com...	1 KB
README.md	28/3/2019 11:56 πμ	Αρχείο MD	1 KB

Αρχιτεκτονική Git

Ανάπτυξη Web Εφαρμογών

- Local working directory
- Local repository
- Tracked files
- Staged files
- Modified but not staged

Πληροφορίες για την staging area

Ανάπτυξη Web Εφαρμογών

- **git status**

```
thanos@thanasis MINGW64 ~/git/java-elearn (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

    ByteShortInt.java

nothing added to commit but untracked files present (use "git add" to track)

thanos@thanasis MINGW64 ~/git/java-elearn (master)
```

Προσθήκη στην staging area

Ανάπτυξη Web Εφαρμογών

- `git add ByteShortInt.java`
 - `git add .`
 - `git add -A (git add --all)`
- Προσθέτει μόνο το συγκεκριμένο αρχείο
 - Προσθέτει όλα τα καινούργια και modified αρχεία του τρέχοντος (.) φακέλου
 - Προσθέτει όλα τα αρχεία στο δένδρο των φακέλων-υποφακέλων

```
thanos@thanasis MINGW64 ~/git/java-elearn (master)
$ git add -A
```

Προσθέτει όλα τα νέα και modified τα αρχεία στο index file / staged files / cached files

```
thanos@thanasis MINGW64 ~/git/java-elearn (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)

    new file:   ByteShortInt.java
```

Εμφανίζει πληροφορίες για τα staging area

Αποστολή στη git DB – commit με message

Ανάπτυξη Web Εφαρμογών

- `git commit -m "..."`

```
thanos@thanasis MINGW64 ~/git/java-elearn (master)
$ git commit -m "Typecast example byteShortInt"
[master c3ff50d] Typecast example byteShortInt
 1 file changed, 15 insertions(+)
 create mode 100644 ByteShortInt.java
```

```
thanos@thanasis MINGW64 ~/git/java-elearn (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Modified files - Add & commit ταυτόχρονα

Ανάπτυξη Web Εφαρμογών

- Αν θέλουμε να κάνουμε add και commit μόνο τις αλλαγές που έχουμε κάνει σε ήδη υπάρχοντα files (ήδη tracked files) στο git repo, τότε μπορούμε με μια εντολή:
- **git commit -a -m “ ... ”**

Αφαίρεση file από staging area

Ανάπτυξη Web Εφαρμογών

- `git rm -cached <file_name>`
- `git reset <file_name>`

Σύνοψη

Ανάπτυξη Web Εφαρμογών

Git: configurations

```
$ git config --global user.name "FirstName LastName"  
$ git config --global user.email "your-email@email-provider.com"  
$ git config --global color.ui true  
$ git config --list
```

Git: starting a repository

```
$ git init  
$ git clone
```

Git: staging files

```
$ git add <file-name>  
$ git add <file-name> <another-file-name> <yet-another-file-name>  
$ git add .  
$ git add --all  
$ git add -A  
$ git rm --cached <file-name>  
$ git reset <file-name>
```

Ανεβάζουμε στο master branch

Ανάπτυξη Web Εφαρμογών

- Η εντολή είναι: **git push <remote> <branch>** π.χ.
git push origin master ή
git push origin main αν δουλεύουμε με main branch

```
thanos@thanasis MINGW64 ~/git/java-elearn (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 483 bytes | 483.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/a8anassis/java-elearn.git
  b623ce7..c3ff50d  master -> master
```

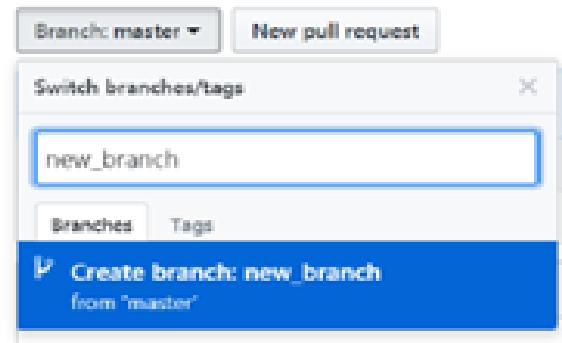
Αν θέλουμε τα default **git push** και **git pull** να αντιστοιχούν στο origin/master (ή origin/main) μπορούμε να ορίσουμε: **git branch --set-upstream-to origin/main** (ή **git branch --set-upstream-to origin/main**)

Branching

Ανάπτυξη Web Εφαρμογών

Στο Git μπορούμε να δουλεύουμε σε διαφορετικά “μονοπάτια” ή branches. Σε κάθε ένα από αυτά, μπορούμε να κάνουμε αλλαγές ανεξάρτητα, χωρίς να αλλάζει το αρχικό. Έτσι, έχουμε ένα κομμάτι κώδικα που δουλεύει και μένει απείραχτο, ενώ μπορούμε να δουλεύουμε σε ομάδες για διαφορετικό πράγμα ο καθένας, σε διαφορετικές εκδόσεις του κώδικα. Όταν οι αλλαγές από το κάθε ένα κομμάτι ολοκληρωθούν, μπορούμε να τις ενώσουμε στο κεντρικό “κορμό” του master branch.

Για να φτιάξουμε ένα νέο branch πηγαίνουμε στο GitHub και πατάμε branch και γράφουμε το όνομα του new branch, στη φόρμα:



Τώρα, επιστρέφοντας στην κονσόλα μπορούμε να δούμε το νέο branch γράφοντας:

Ανανεώνει την κατάσταση
του repository. → git pull

Αλλάζει σε ένα από
τα branches του
repository στο οποίο
βρισκόμαστε. → git checkout new_branch

Merging

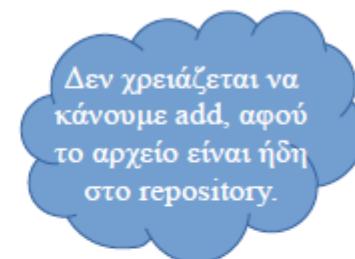
Ανάπτυξη Web Εφαρμογών

Κάνουμε οποιεσδήποτε αλλαγές στον κώδικά μας. Για να τις στείλουμε στο GitHub κάνουμε το ίδιο με πρίν:



```
git commit -a -m "new message"
```

```
git push
```



Η διαφορά με πρίν είναι ότι τώρα είμαστε σε διαφορετικό branch. Ας πούμε ότι τελειώσαμε τη δουλειά σε αυτό το branch και θέλουμε να την μεταφέρουμε στο master. Πρέπει να κάνουμε merge:

Ξαναπάμε στο master branch



```
git checkout master
```

```
git merge new_branch
```



Ενώνουμε τις αλλαγές του `new_branch` στο branch που βρισκόμαστε τώρα (το `master`)

Ξανασπρώχνουμε στο GitHub



```
git push
```

Δημιουργία νέου branch (2)

Ανάπτυξη Web Εφαρμογών

- Δημιουργήσαμε ένα νέο branch στο GitHub, έστω Part1-StructuralProgramming και ώρα κάνουμε **git pull** για να ‘φέρουμε’ το νέο branch
- Αν φτιάχναμε το branch τοπικά με `git checkout -b [name-of-new-branch]` μετά θα έπρεπε να κάνουμε `git push origin [name-of-new-branch]`

```
thanos@thanasis MINGW64 ~/git
$ cd java-elearn/

thanos@thanasis MINGW64 ~/git/java-elearn (master)
$ git pull
From https://github.com/a8anassis/java-elearn
 * [new branch]      Part1_StructuralProgramming -> origin/Part1_StructuralProgr
amming
Already up to date.

thanos@thanasis MINGW64 ~/git/java-elearn (master)
$
```

git branch -a

Ανάπτυξη Web Εφαρμογών

- Εμφανίζουμε τα branches

```
thanatos@thanasis MINGW64 ~/git/java-elearn (Part1_StructuralProgramming)
$ git branch -a
* Part1_StructuralProgramming
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/Part1_StructuralProgramming
  remotes/origin/master

thanatos@thanasis MINGW64 ~/git/java-elearn (Part1_StructuralProgramming)
```

Checkout και HEAD

Ανάπτυξη Web Εφαρμογών

- Checkout στο νέο branch
- Τώρα το HEAD δείχνει στο Part1_StructuralProgramming
- To HEAD είναι ένα δείκτης στο commit του τρέχοντος branch

```
thanatos@thanasis MINGW64 ~/git/java-elearn (master)
$ git checkout Part1_StructuralProgramming
Switched to a new branch 'Part1_StructuralProgramming'
Branch 'Part1_StructuralProgramming' set up to track remote branch 'Part1_StructuralProgramming' from 'origin'.
```

```
thanatos@thanasis MINGW64 ~/git/java-elearn (Part1_StructuralProgramming)
$
```

checkout στο master branch

Ανάπτυξη Web Εφαρμογών

- Ξανά checkout στο master branch

```
thanos@thanasis MINGW64 ~/git/java-elearn (Part1_StructuralProgramming)
$ git branch
* Part1_StructuralProgramming
  master

thanos@thanasis MINGW64 ~/git/java-elearn (Part1_StructuralProgramming)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

thanos@thanasis MINGW64 ~/git/java-elearn (master)
```

Όπως έχουμε πει αν δουλεύουμε με main branch (αντί για master) όπου βλέπετε master στα παραδείγματα, το αντικαθιστάτε με main (για παράδειγμα, εδώ, git checkout main)

Ανάκληση commit

Ανάπτυξη Web Εφαρμογών

Git: committing to a repository

```
$ git commit -m "Add three files"  
$ git reset --soft HEAD^  
$ git commit --amend -m <enter your message>
```

- Όπως εύπαμε το HEAD είναι ένα δείκτης στο τρέχον branch
- Το ^ σημαίνει το τελευταίο commit
- Το reset –soft HEAD^ σημαίνει: undo το last commit στο τρέχον branch και μετακίνησε τον HEAD πίσω κατά ένα commit

Update ενός commit

Ανάπτυξη Web Εφαρμογών

- Αντί να ανακαλέσουμε ένα commit μπορούμε να το κάνουμε update

```
1 $ git add file-i-forgot-to-add.html  
2 $ git commit --amend -m "Add the remaining file"
```

- Σύνοψη

Git: committing to a repository

```
$ git commit -m "Add three files"  
$ git reset --soft HEAD^  
$ git commit --amend -m <enter your message>
```

Καλές πρακτικές

Ανάπτυξη Web Εφαρμογών

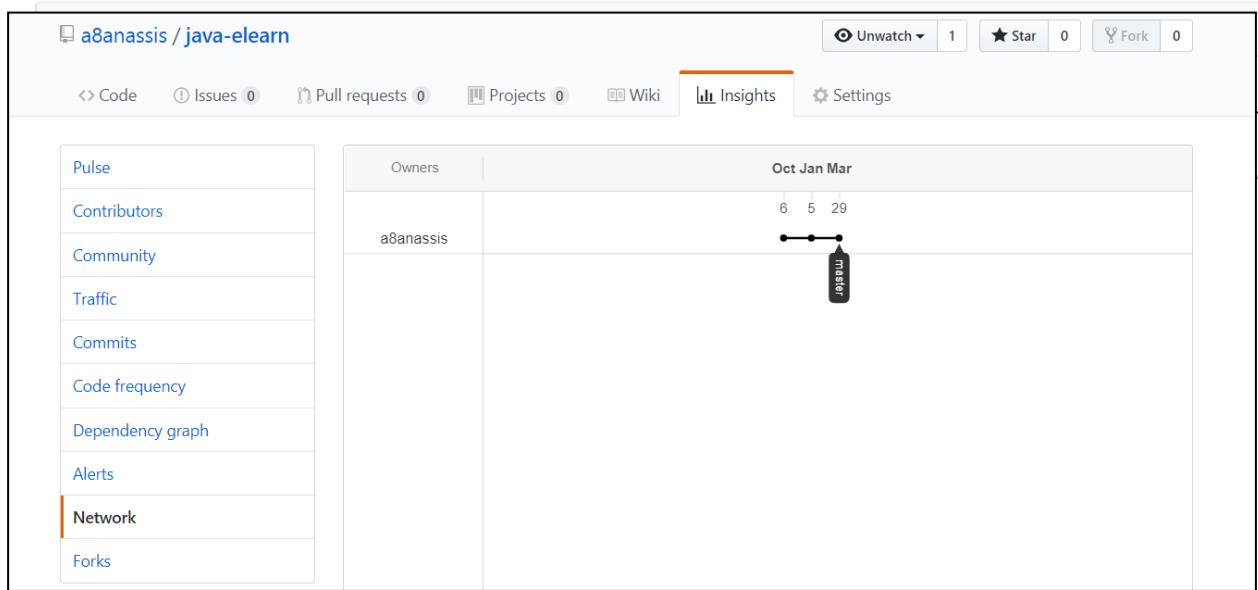
Αυτή ήταν η βασική χρήση του Git. Μερικές καλές πρακτικές:

- Φτιάχνουμε branches που αντιπροσωπεύουν διαφορετικές ενότητες εργασίας (περισσότερα σε λίγο).
- Δεν σπρώχνουμε ημιτελή κώδικα στο master branch. Το master πρέπει **πάντα** να περιέχει κώδικα που δουλέυει.
- Αν δουλεύουμε με άλλους, ρωτάμε από ποιό branch να αρχίσουμε να δουλεύουμε και τραβάμε και σπρώχνουμε εκεί. Συνήθως ο κάθε χρήστης έχει δικό του branch και σπρώχνει εκεί, και ο επιβλέπων κάνει την ένωση με τα develop και master (περισσότερα σε λίγο).
- Αποφεύγουμε να δουλεύουμε σε branches που δουλεύουν και άλλοι παράλληλα. Απλά κάνουμε νέο branch και κάνουμε τις αλλαγές μας εκεί, και μετά κάνουμε τα αντίστοιχα merge.

Επισκόπηση των branches – Git Network

Ανάπτυξη Web Εφαρμογών

Πηγαίνουμε στο GitHub και πατάμε insights -> network. Μας εμφανίζεται η εικόνα του δικτύου του repository μας:



Προφανώς αυτή η εικόνα μπορεί να γίνει πολύ πιο περίπλοκη.

Είδη branches – git flow

Ανάπτυξη Web Εφαρμογών

Για να αποφευχθεί το απόλυτο χάος, συνήθως ο κάθε χρήστης έχει το δικό του branch. Επίσης υπάρχουν 2 βασικά branches, το develop και το master, στα οποία έχουν πρόσβαση οι managers της ομάδας. Υπάρχει επίσης, ένα branch για κάθε μεγάλο κομμάτι εργασίας, από το οποίο κάνουν μικρότερα branches ο κάθε ένας που δουλεύει εκεί, ένα για κάθε μικρότερη δουλειά που κάνουν.

Αυτά τα μικρότερα branches ονομάζονται feature branches και το καθένα ξεκινά το όνομά του ανάλογα με το τι κάνει, πχ: feature-new_button, bug-wrong_dates, task-new_page.

Στην περίπτωση που θέλουμε μια άμεση αλλαγή στο master ή στο develop branch κάνουμε ένα hotfix branch με αντίστοιχο όνομα. Η ιδέα είναι ότι αυτά έχουν πολύ λίγες αλλαγές στον κώδικα και γενικά αποφεύγονται.

Όλες αυτές οι συμβάσεις που κάνουμε είναι μέρος του “git workflow”. Περισσότερα για αυτό στους συνδέσμους:

- <https://nvie.com/posts/a-successful-git-branching-model>
- <https://guides.github.com/introduction/flow/>
- <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

Conflicts (1)

Ανάπτυξη Web Εφαρμογών

Τα conflicts εμφανίζονται όταν δύο branches έχουν διαφορετικές εκδοχές του ίδιου αρχείου και των ίδιων γραμμών κώδικα και πάλι να τα κάνουμε merge.

Ας πούμε ότι αλλάζουμε το αρχείο “test.txt” στο master, και το αλλάζουμε διαφορετικά στο new_branch και κάνουμε και τις 2 αλλαγές commit και push. Όταν πάμε να κάνουμε merge το new_branch στο master, το git μας δίνει το ακόλουθο μήνυμα:

```
$ git merge new_branch
```

```
Auto-merging test.txt
```

```
CONFLICT (content): Merge conflict in test.txt
```

```
Automatic merge failed; fix conflicts and then commit the result.
```

Conflicts (2)

Ανάπτυξη Web Εφαρμογών

Βήμα 1: **ΔΕΝ** ανησυχούμε.

Το Git δημιουργήθηκε ακριβώς για αυτό το σκοπό. Κάθε αλλαγή, ακόμα και καταλάθος να σβήσουμε όλο τον κώδικα, μπορούμε εύκολα να την αναιρέσουμε (δεν θα το καλύψουμε εδώ πως γίνεται στη γενική περίπτωση).

Ας ανοίξουμε το αρχέιο:

<<<<<< HEAD

Μας δείχνει που ξεκινά το conflict

12 12 – allagi apo master

=====

Μας δείχνει που τελειώνει σε αυτό το version

13 13 – allagi apo new branch!

>>>>> new_branch

Μας δείχνει που τελειώνει το δεύτερο version

123

Conflicts (3)

Ανάπτυξη Web Εφαρμογών

Τι κάνουμε;

Απλά γράφουμε τι θέλουμε να μείνει στον κώδικα. Κρατάμε τις αλλαγές που μας αρέσουν και σβήνουμε τα HEAD, ===== και <<<, >>> και ξανακάνουμε commit και push!

Σα να μη συνέβη τίποτα. Η ιδέα είναι ότι δεν πρέπει να γίνονται conflicts, αλλά αυτό καμιά φορά είναι αναπόφευκτο. Οι καλές πρακτικές του git workflow φροντίζουν να ελαχιστοποιούνται τα conflicts.

Collaborators

Ανάπτυξη Web Εφαρμογών

- Στο GitHub Free υπάρχουν δύο μόνο ρόλοι:
 - **Owner** που δεν μπορεί να μεταβιβαστεί
 - **Collaborators (Settings/Collaborators)** που έχουν write permissions
- Στα private repos μπορούμε να προσθέσουμε **μέχρι 3 collaborators**
- Περισσότερα προϊόντα GitHub
 - <https://help.github.com/en/articles/githubs-products>

Ιστορικό commits – git log

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git log
commit 0346427d809dd4072f6cdef1199cef7289ac956 (HEAD -> master)
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 13:00:13 2019 +0300

    new commit

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:59:32 2019 +0300

    ByteShortInt

commit 850412abde3b468837b4e3d5dbb4d13e4e539758
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:51:18 2019 +0300

    First Commit

commit 5bc6b52dd3b0bbb2cacfeb7cce67b6b1c7efe424 (origin/master)
Author: Athanassios Androutsos <a8anassis@gmail.com>
Date:   Wed Apr 17 20:46:55 2019 +0300

    Initial commit
```

- Trace back the commit history
- **4 commits** εκ των οποίων το 1^o έγινε στο origin/master

git log -p -2

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git log -p -2
commit 0346427d809dd4072f6cdef1199cef7289ac956 (HEAD -> master)
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 13:00:13 2019 +0300

    new commit

diff --git a/ByteShortInt.java b/ByteShortInt.java
index 5323b56..cdbc479 100644
--- a/ByteShortInt.java
+++ b/ByteShortInt.java
@@ -6,7 +6,7 @@ package gr.thanassis.tmp;
 public class ByteShortInt {
     public static void main(String[] args) {
         int i = 15;
-        int j = 20;
+        int j = 320;
         System.out.println("i+j = " + (i + j));
         System.out.printf("i+j = %04d", i+j);
     }
 }

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:59:32 2019 +0300

    ByteShortInt

diff --git a/ByteShortInt.java b/ByteShortInt.java
new file mode 100644
index 0000000..5323b56
--- /dev/null
+++ b/ByteShortInt.java
@@ -0,0 +1,15 @@
+package gr.thanassis.tmp;
+
+/**
+ * Created by thanos on 10/4/2018.
+ */
+public class ByteShortInt {
+    public static void main(String[] args) {
+        int i = 15;
+        int j = 20;
+        System.out.println("i+j = " + (i + j));
+        System.out.printf("i+j = %04d", i+j);
+    }
+}
```

- -p ή --patch Δείχνει τις διαφορές σε κάθε commit

git log --stat

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git log --stat
commit 0346427d809dd4072f6cdef1199cefd7289ac956 (HEAD -> master)
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 13:00:13 2019 +0300

    new commit

ByteShortInt.java | 2 ++
1 file changed, 1 insertion(+), 1 deletion(-)

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:59:32 2019 +0300

ByteShortInt

ByteShortInt.java | 15 ++++++
1 file changed, 15 insertions(+)

commit 850412abde3b468837b4e3d5dbb4d13e4e539758
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:51:18 2019 +0300

First Commit

HelloWorld.java | 18 ++++++
1 file changed, 18 insertions(+)

commit 5bc6b52dd3b0bbb2cacfeb7cce67b6b1c7efe424 (origin/master)
Author: Athanassios Androutsos <a8anassis@gmail.com>
Date:   Wed Apr 17 20:46:55 2019 +0300

Initial commit

README.md | 1 +
1 file changed, 1 insertion(+)
```

Git log –author='name'

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git log --author='Th. Androutsos'
commit 0346427d809dd4072f6cdef1199cef7289ac956 (HEAD -> master)
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 13:00:13 2019 +0300

    new commit

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:59:32 2019 +0300

    ByteShortInt

commit 850412abde3b468837b4e3d5dbb4d13e4e539758
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:51:18 2019 +0300

    First Commit
```

History of a file

git log -- <filename>

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git log -- ByteShortInt.java
commit 0346427d809dd4072f6cdef1199cef7289ac956 (HEAD -> master)
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 13:00:13 2019 +0300

    new commit

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutsos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:59:32 2019 +0300

    ByteShortInt
```

- *git log – ByteShortInt.java*

Git log -p -- <filename>

Ανάπτυξη Web Εφαρμογών

```
$ git log -p -- ByteShortInt.java
commit 0346427d809dd4072f6cdef1199cefd7289ac956 (HEAD -> master)
Author: Th. Androutossos <a8anassis@gmail.com>
Date:   Thu Apr 18 13:00:13 2019 +0300

    new commit

diff --git a/ByteShortInt.java b/ByteShortInt.java
index 5323b56..cdcb479 100644
--- a/ByteShortInt.java
+++ b/ByteShortInt.java
@@ -6,7 +6,7 @@ package gr.thanassis.tmp;
 public class ByteShortInt {
     public static void main(String[] args) {
         int i = 15;
-        int j = 20;
+        int j = 320;
         System.out.println("i+j = " + (i + j));
         System.out.printf("i+j = %04d", i+j);
     }
 }

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutossos <a8anassis@gmail.com>
Date:   Thu Apr 18 12:59:32 2019 +0300

    ByteShortInt

diff --git a/ByteShortInt.java b/ByteShortInt.java
new file mode 100644
index 0000000..5323b56
--- /dev/null
+++ b/ByteShortInt.java
@@ -0,0 +1,15 @@
+package gr.thanassis.tmp;
+
+/**
+ * Created by thanos on 10/4/2018.
+ */
+public class ByteShortInt {
+    public static void main(String[] args) {
+        int i = 15;
+        int j = 20;
+        System.out.println("i+j = " + (i + j));
+        System.out.printf("i+j = %04d", i+j);
+    }
+}
```

- git log -p --
ByteShortInt.java

Επαναφορά παλιάς έκδοσης

Ανάπτυξη Web Εφαρμογών

Αφού βρούμε το hash value του commit, μπορούμε να το κάνουμε checkout

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git checkout 536d472c4b707dec639c38a1b88d87eb5216b7b6 -- ByteShortInt.java
```

Hash value

Ανάπτυξη Web Εφαρμογών

```
$ git log -p ByteShortInt.java
commit 0346427d809dd4072f6cdef1199cef7289ac956 (HEAD -> master)
Author: Th. Androutsos <a8anassis@gmail.com>
Date: Thu Apr 18 13:00:13 2019 +0300

    new commit

diff --git a/ByteShortInt.java b/ByteShortInt.java
index 5323b56..cdbc479 100644
--- a/ByteShortInt.java
+++ b/ByteShortInt.java
@@ -6,7 +6,7 @@ package gr.thanassis.tmp;
 public class ByteShortInt {
     public static void main(String[] args) {
         int i = 15;
-        int j = 20;
+        int j = 320;
         System.out.println("i+j = " + (i + j));
         System.out.printf("i+j = %04d", i+j);
     }
 }
```

```
commit 536d472c4b707dec639c38a1b88d87eb5216b7b6
Author: Th. Androutsos <a8anassis@gmail.com>
Date: Thu Apr 18 12:59:32 2019 +0300

ByteShortInt

diff --git a/ByteShortInt.java b/ByteShortInt.java
new file mode 100644
index 0000000..5323b56
--- /dev/null
+++ b/ByteShortInt.java
@@ -0,0 +1,15 @@
+package gr.thanassis.tmp;
+
+/*
+ * Created by thanos on 10/4/2018.
+ */
+public class ByteShortInt {
+    public static void main(String[] args) {
+        int i = 15;
+        int j = 20;
+        System.out.println("i+j = " + (i + j));
+        System.out.printf("i+j = %04d", i+j);
+    }
+}
```

- Hash του commit που θέλουμε

commit 536d472c4b707dec639c38a1b88d87eb5216b7b6

git diff <hash> <filename>

Ανάπτυξη Web Εφαρμογών

```
thanatos@thanasis MINGW64 ~/git/codingFactory (master)
$ git diff 536d472c4b707dec639c38a1b88d87eb5216b7b6 ByteShortInt.java
diff --git a/ByteShortInt.java b/ByteShortInt.java
index 5323b56..cdbc479 100644
--- a/ByteShortInt.java
+++ b/ByteShortInt.java
@@ -6,7 +6,7 @@ package gr.thanassis.tmp;
 public class ByteShortInt {
     public static void main(String[] args) {
         int i = 15;
-        int j = 20;
+        int j = 320;
         System.out.println("i+j = " + (i + j));
         System.out.printf("i+j = %04d", i+j);
     }
}

thanatos@thanasis MINGW64 ~/git/codingFactory (master)
```

Git checkout

Ανάπτυξη Web Εφαρμογών

Αν θέλουμε να δημιουργήσουμε ένα branch σε κάποιο πρότερο σημείο του χρόνου τότε, πάμε πίσω με

`git checkout <commit-ref>`

`git checkout -b <new branch name>`

Ανάκτηση παλιού version αρχείου

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git checkout 536d472c4b707dec639c38a1b88d87eb5216b7b6 -- ByteShortInt.java

thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ cat ByteShortInt.java
package gr.thanassis.tmp;

/**
 * Created by thanos on 10/4/2018.
 */
public class ByteShortInt {
    public static void main(String[] args) {
        int i = 15;
        int j = 20;
        System.out.println("i+j = " + (i + j));
        System.out.printf("i+j = %04d", i+j);
    }
}

thanos@thanasis MINGW64 ~/git/codingFactory (master)
$
```

- Κάναμε retrieve το παλιό version

Ανάκτηση αρχείου από το τελευταίο commit

Ανάπτυξη Web Εφαρμογών

```
thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ git checkout HEAD ByteShortInt.java
Updated 1 path from 527b284

thanos@thanasis MINGW64 ~/git/codingFactory (master)
$ cat ByteShortInt.java
package gr.thanassis.tmp;

/**
 * Created by thanos on 10/4/2018.
 */
public class ByteShortInt {
    public static void main(String[] args) {
        int i = 15;
        int j = 320;
        System.out.println("i+j = " + (i + j));
        System.out.printf("i+j = %04d", i+j);
    }
}
```

- **HEAD** είναι η αναφορά (reference) στο τελευταίο commit

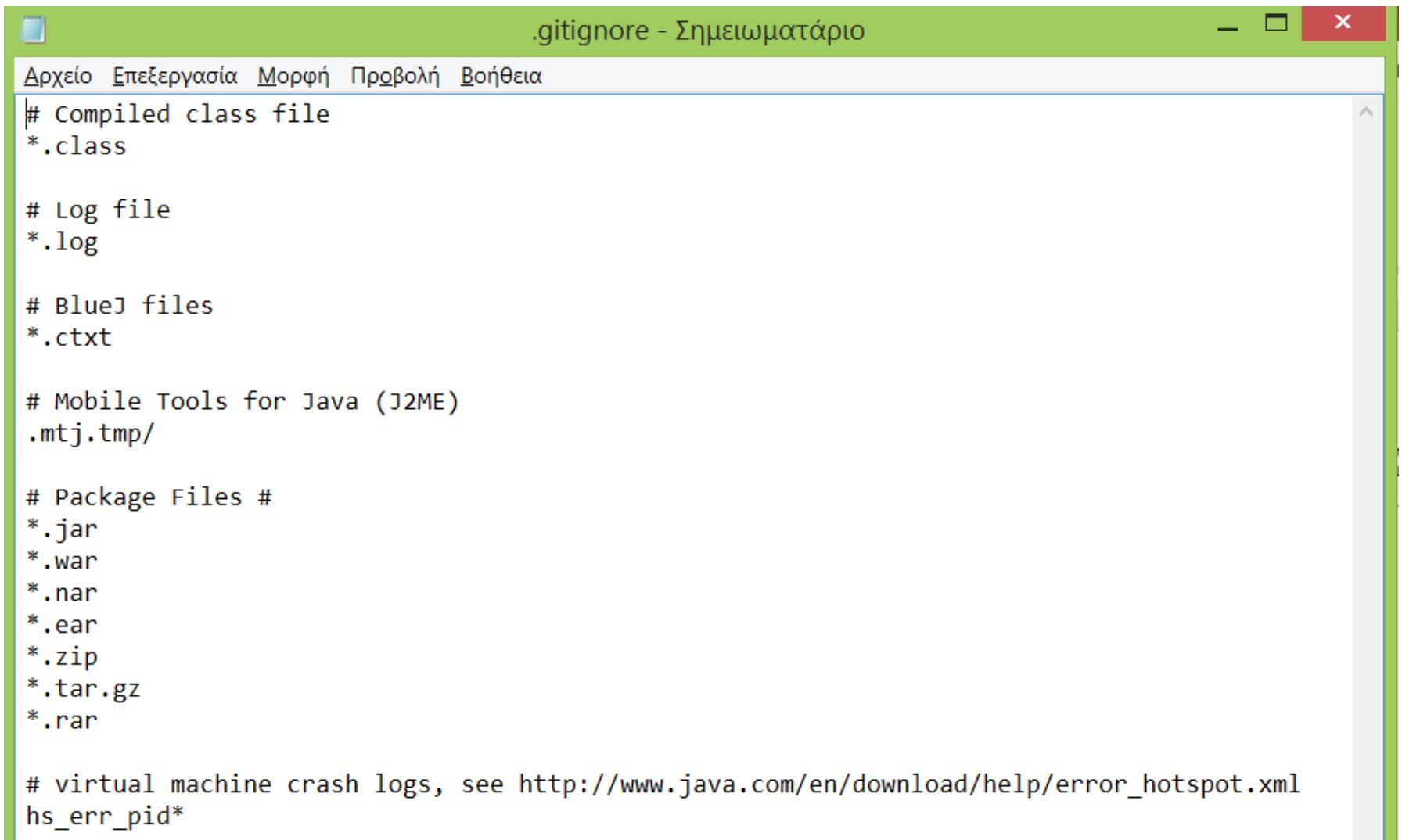
Αρχείο .gitignore (1)

Ανάπτυξη Web Εφαρμογών

- Αρχεία που αγνοούμε όταν κάνουμε όταν προσθέτουμε αρχεία στο staging area (π.χ. με git add –A)
- Το αρχείο .gitignore βρίκεται στο root φάκελο του repository
- Π.χ. με *.class αγνοούμε όλα τα class αρχεία της java που μπορεί να βρίσκονται μέσα στο φάκελο που εποπτεύουμε
- Δείτε περισσότερα στο <https://www.atlassian.com/git/tutorials/saving-changes/gitignore>

Παράδειγμα αρχείου .gitignore (2)

Ανάπτυξη Web Εφαρμογών



.gitignore - Σημειωματάριο

```
# Compiled class file
*.class

# Log file
*.log

# BlueJ files
*.ctxt

# Mobile Tools for Java (J2ME)
.mtj.tmp/

# Package Files #
*.jar
*.war
*.nar
*.ear
*.zip
*.tar.gz
*.rar

# virtual machine crash logs, see http://www.java.com/en/download/help/error_hotspot.xml
hs_err_pid*
```

Βιβλιογραφικές Αναφορές (1)

Ανάπτυξη Web Εφαρμογών

1 <https://git-scm.com/>

- Η ιστοσελίδα του Git.

2 <https://github.com/>

- Η ιστοσελίδα του GitHub.

3 <https://www.atlassian.com/git/tutorials/what-is-version-control>

- Περισσότερα για το version control.

4 <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

- Αναλυτικές οδηγίες εγκατάστασης.

5 <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>

- Αναλυτικές οδηγίες αρχικοποίησης.

Βιβλιογραφικές Αναφορές (2)

Ανάπτυξη Web Εφαρμογών

S. Chacon and B. Straub, *Pro Git*, 2nd ed. Apress 2014. [E-book] Available.

- Το βασικό ebook που αναλύει τα πάντα για το git για όποιον επιθυμεί να εμβαθύνει πλήρως. Μπορεί να βρεθεί στη διεύθυνση:

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

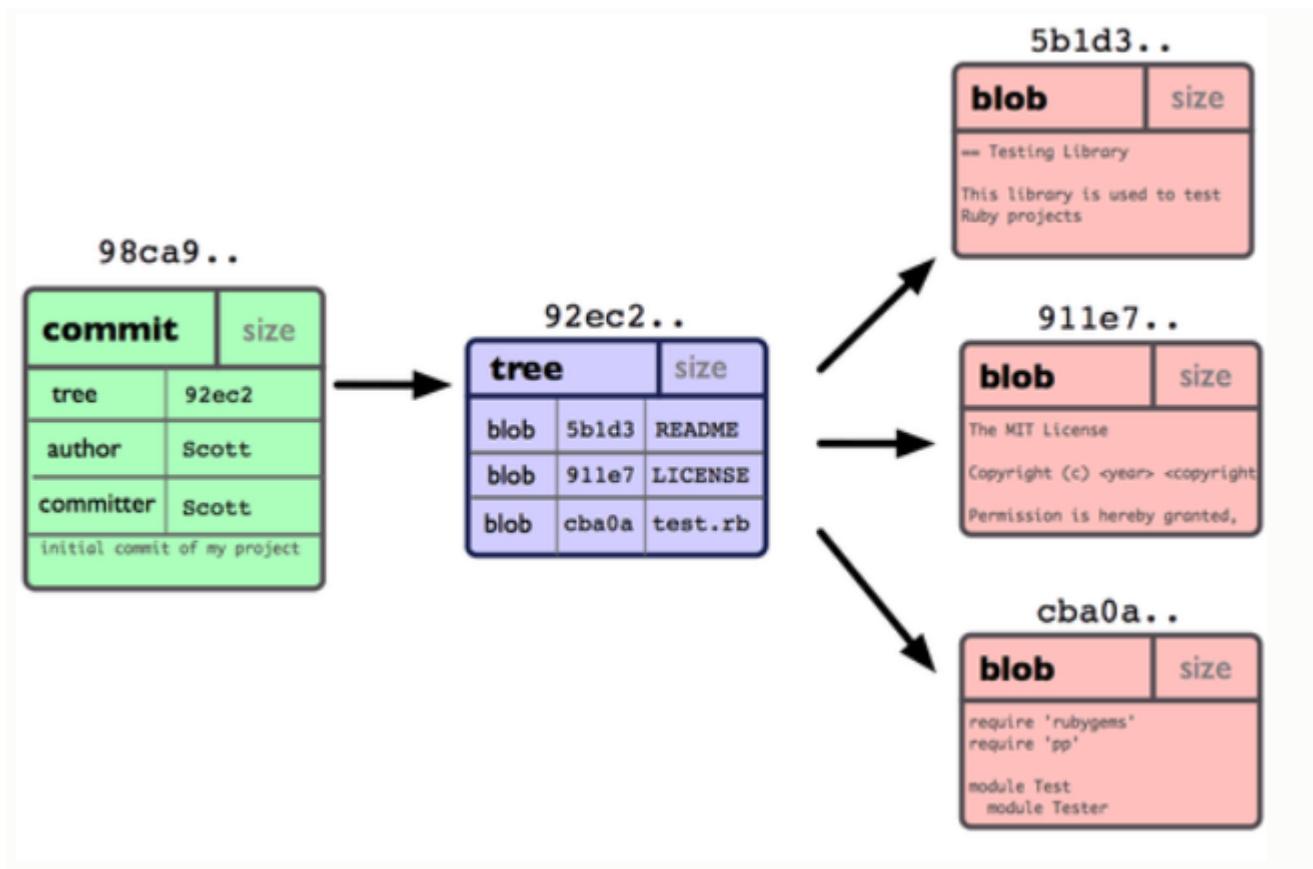
Ειδικά Θέματα στο Git

Ανάπτυξη Web Εφαρμογών

- Στις επόμενες διαφάνειες θα δούμε πως το git χειρίζεται τα branch / checkout και merge

Commit object

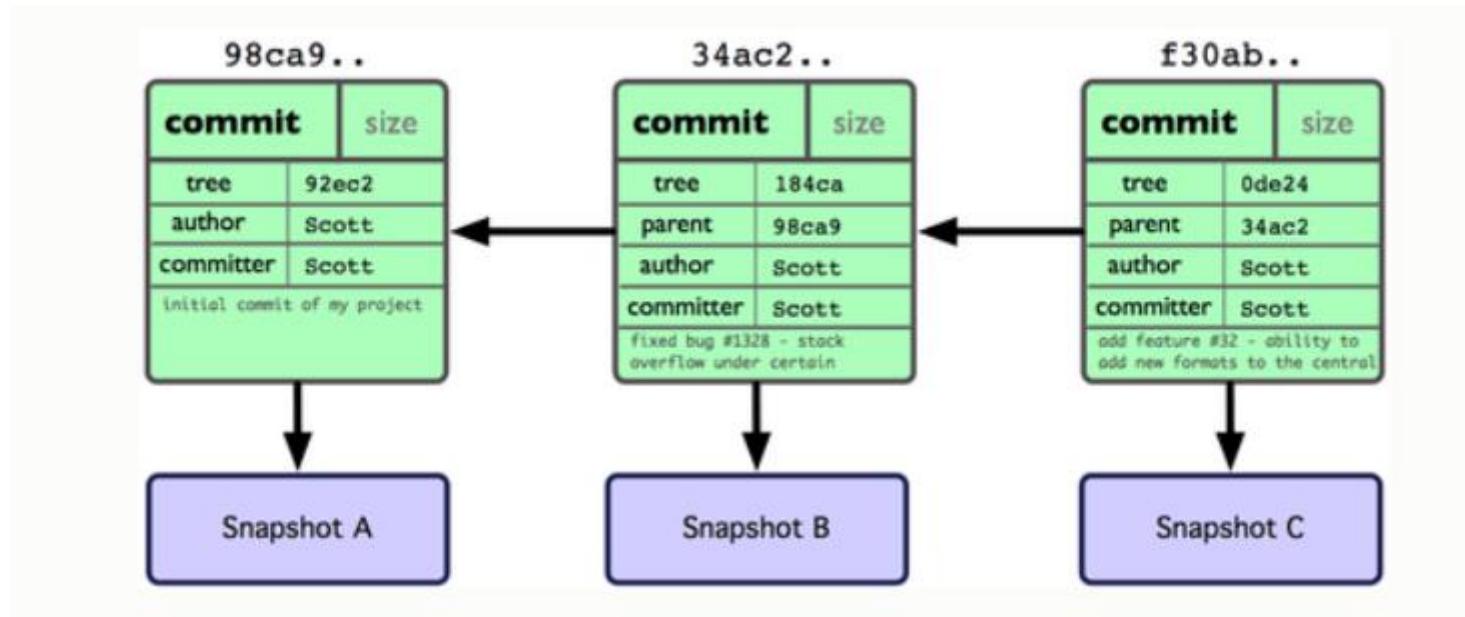
Ανάπτυξη Web Εφαρμογών



- Κάθε commit δημιουργεί tree objects και ένα blob για κάθε αρχείο που γίνεται commit

Commit objects

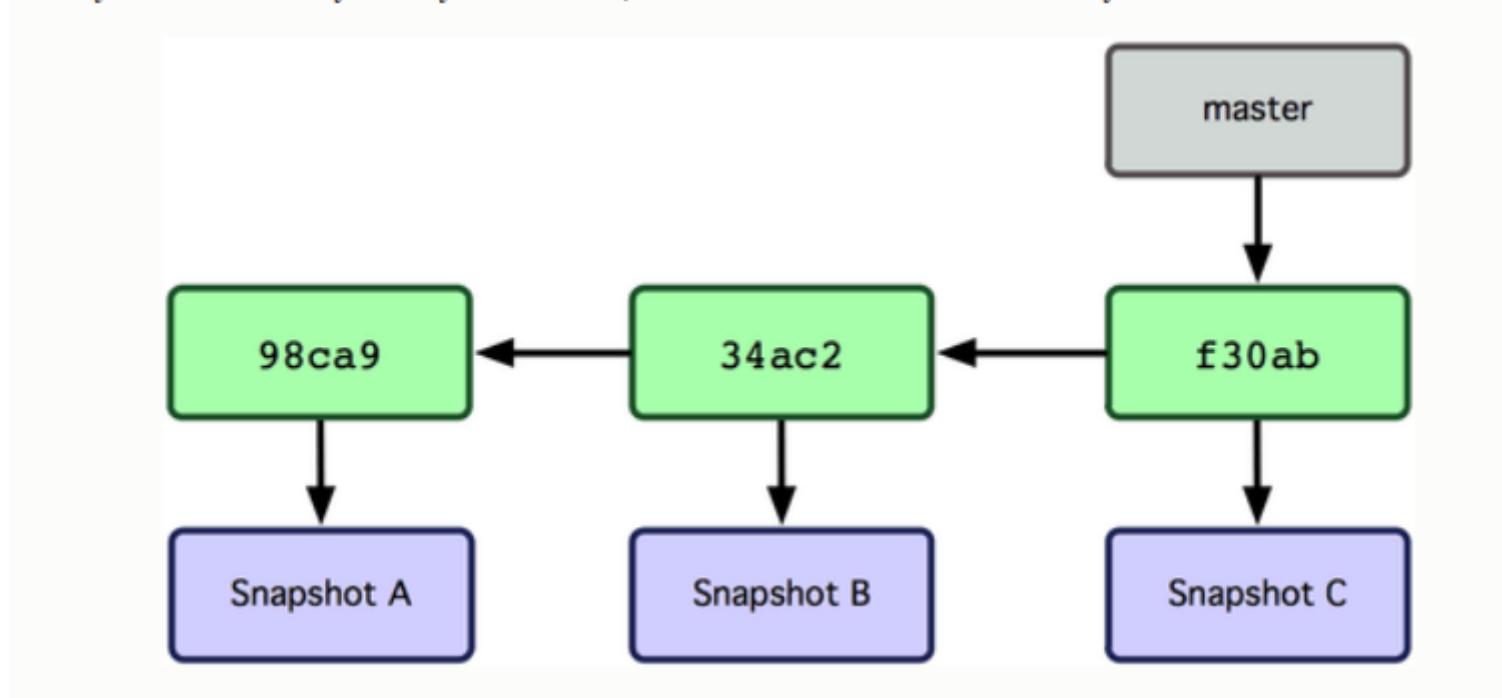
Ανάπτυξη Web Εφαρμογών



- Για κάθε καινούργιο commit δημιουργείται νέο commit object που δείχνει στο προηγούμενο

Branches – Master branch

Ανάπτυξη Web Εφαρμογών



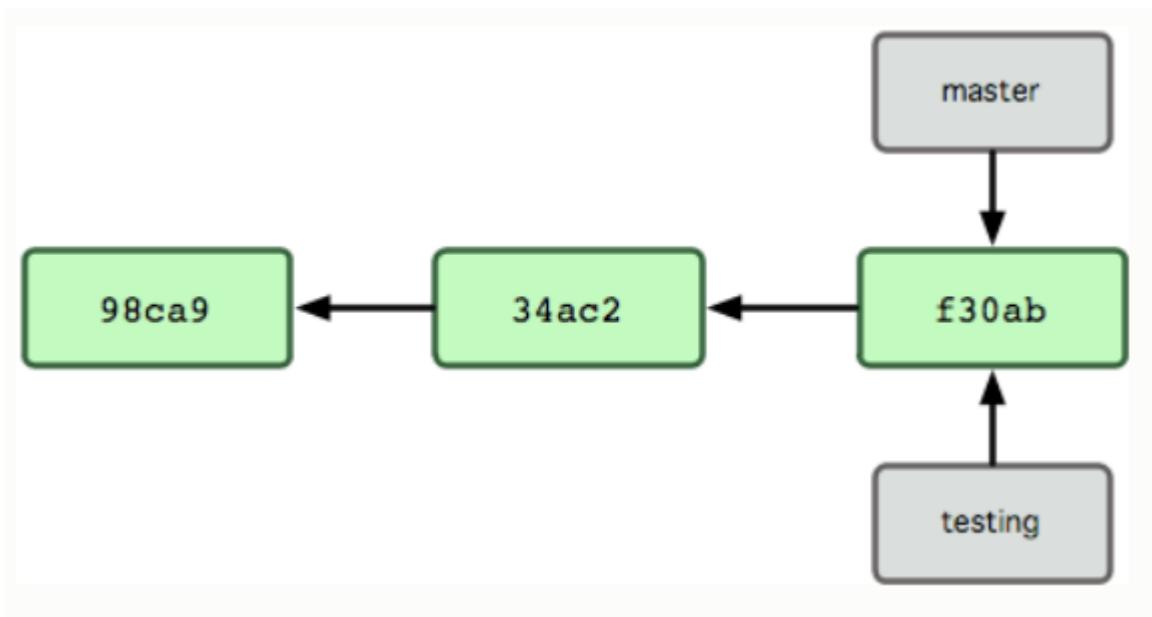
Κάθε branch στο git είναι ένα δείκτης που μπορεί να μετακινείται μεταξύ των commits. Δείχνει στο τελευταίο commit και μετακινείται κάθε φορά που κάνουμε commit

Δημιουργία νέου branch

Ανάπτυξη Web Εφαρμογών

- **git branch testing**

- Δημιουργείται νέος δέικτης στο τελευταίο commit



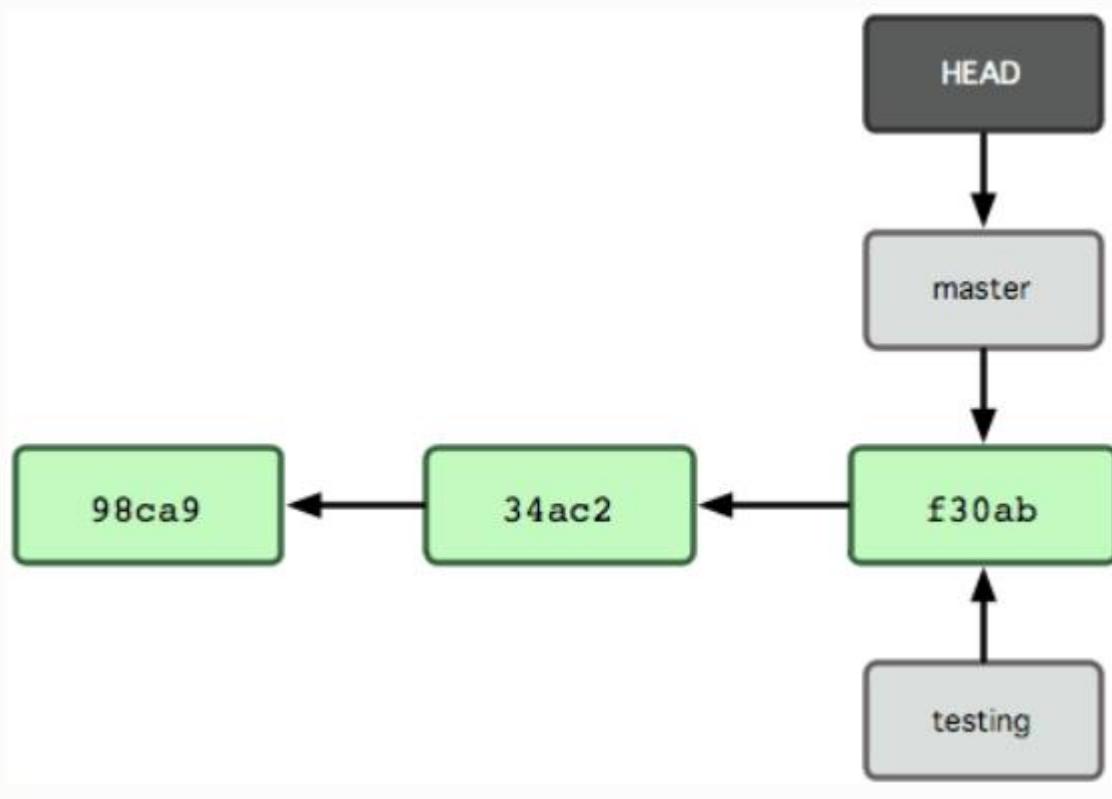
HEAD (1)

Ανάπτυξη Web Εφαρμογών

- Πως το git γνωρίζει σε ποιο branch είμαστε;
- Χρησιμοποιεί ένα άλλο δείκτη που δείχνει Special pointer που δείχνει στο current branch
- Δείτε επόμενη διαφάνεια

HEAD(2)

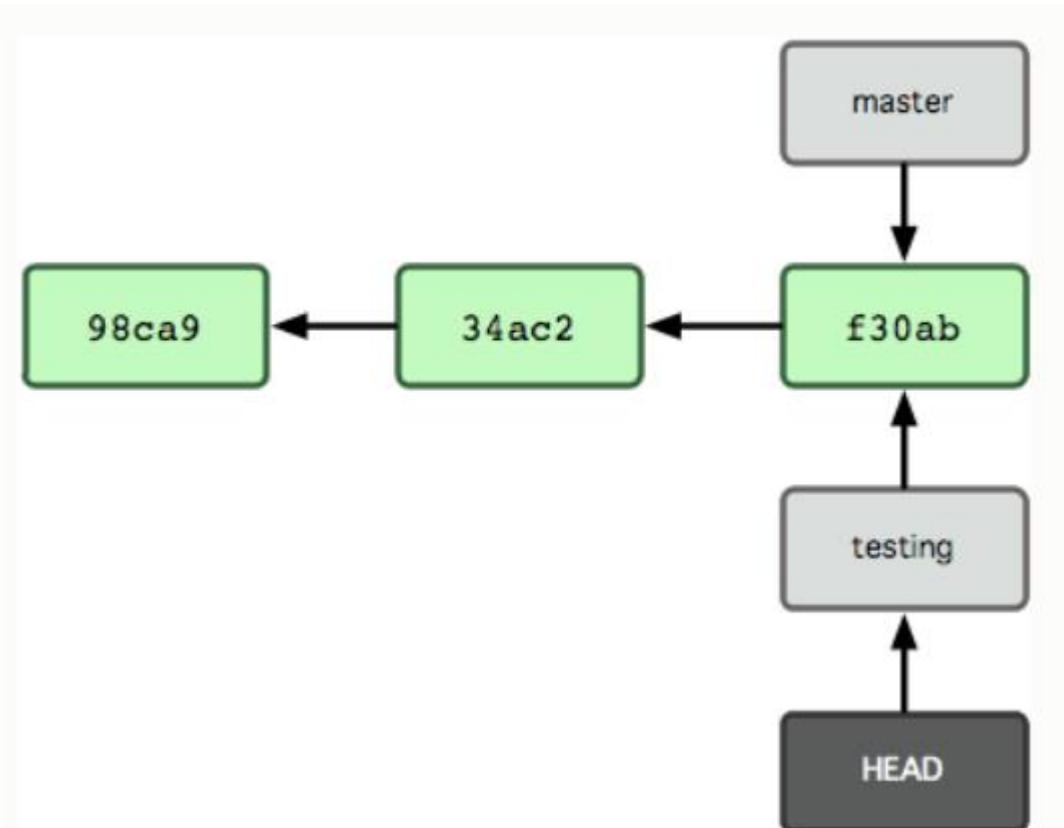
Ανάπτυξη Web Εφαρμογών



- Ο HEAD είναι ένας δείκτης που δείχνει στο current branch

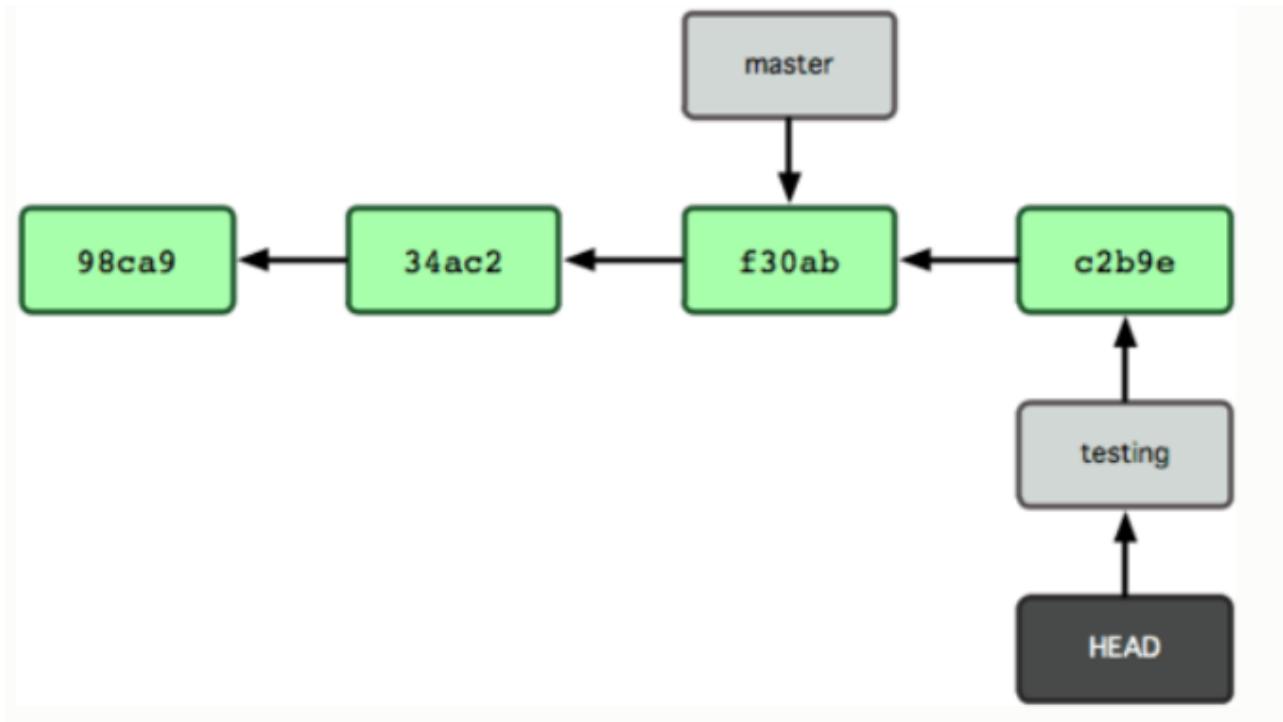
HEAD (3)

- **git checkout testing**



HEAD (4)

- **git commit –a –m “a change to a file”**

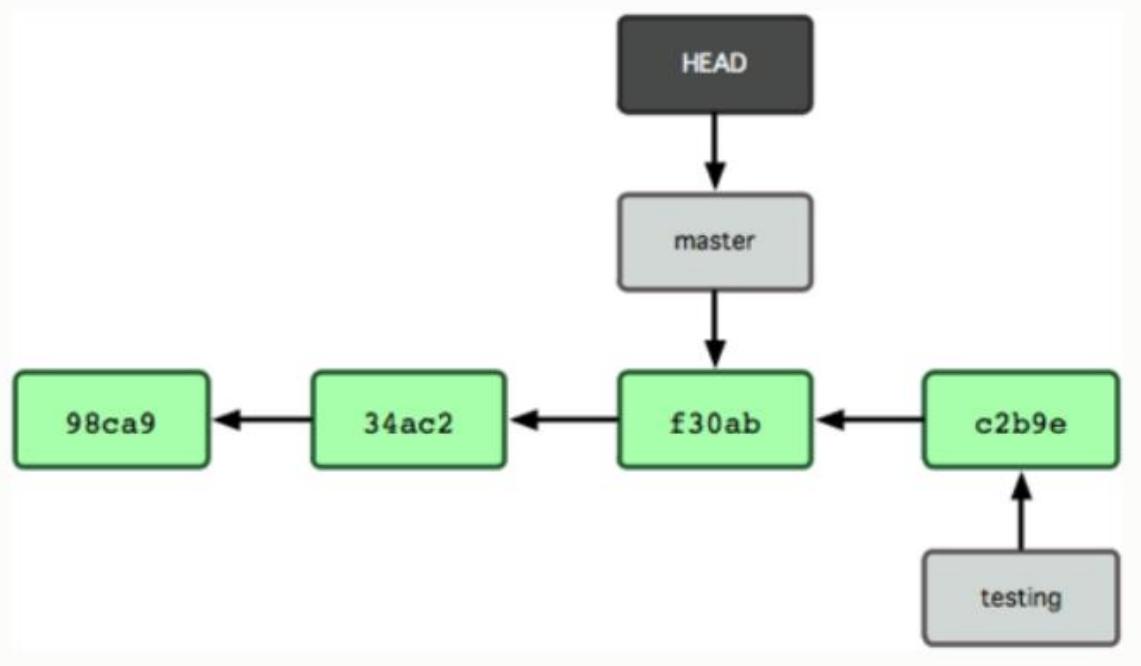


To HEAD προχωράει με κάθε commit

To ίδιο συνέβη και με το testing αφού το commit έγινε όταν είχαμε κάνει checkout στο branch testing

HEAD (5)

- **git checkout master**



Άρα το **HEAD** δείχνει στο *current branch*

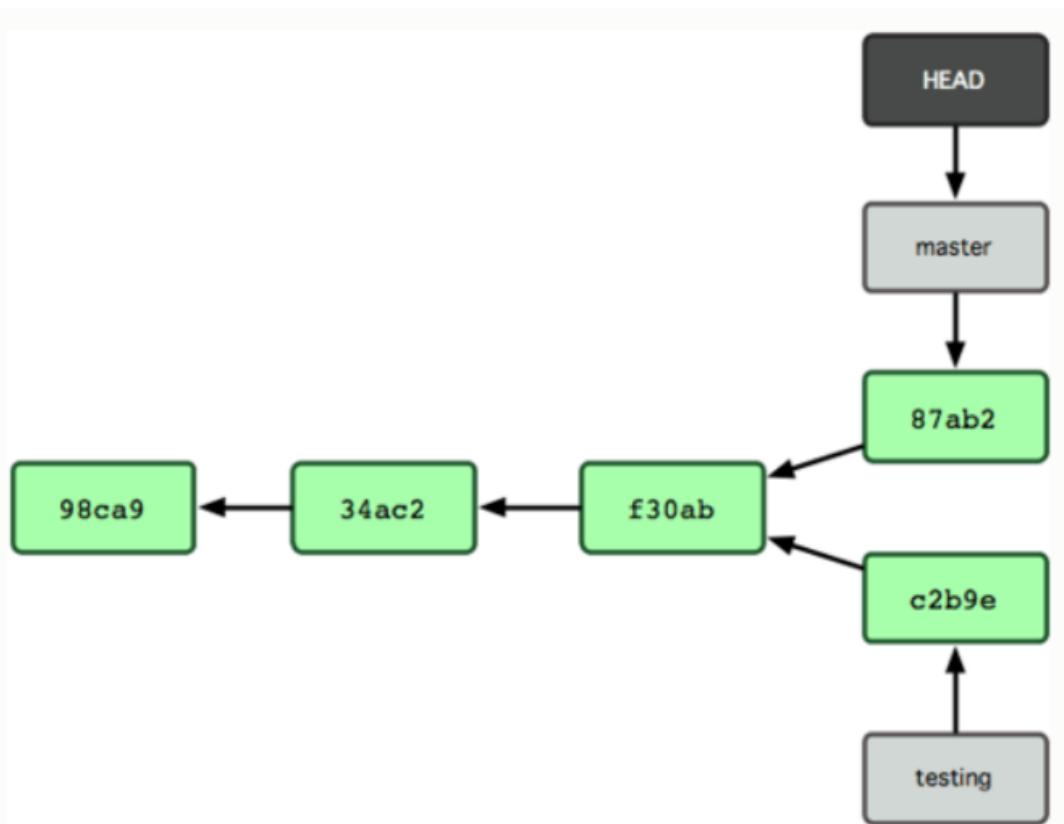
Η εντολή αυτή έκανε δύο πράγματα

1. **Μετακίνησε τον HEAD πίσω** στο master branch, και
2. **Τοποθέτησε στο working directory** τα αρχεία του *snapshot* που δείχνει ο *master*

Ότι αλλαγές γίνουν εδώ θα δημιουργήσουν ένα άλλο μονοπάτι που θα διαφέρει από την παλιά version του project

HEAD (6)

- git -a -m “new changes”



Οι αλλαγές μετά το 3^o commit βρίσκονται σε 2 διαφορετικά branches
Μπορούμε να δουλεύουμε στα 2 branches και όταν ολοκληρώσουμε να κάνουμε merge

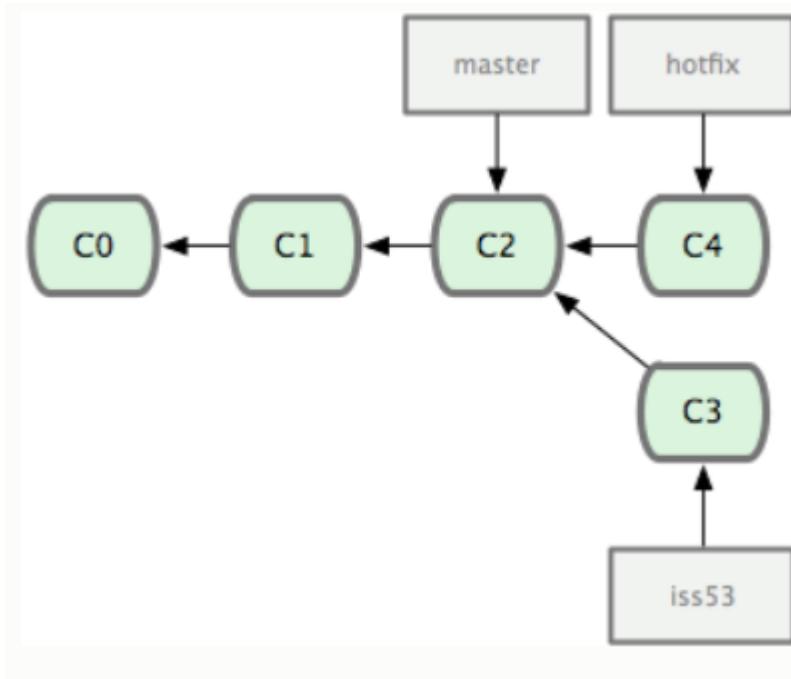
Branching

Ανάπτυξη Web Εφαρμογών

- Κάθε branch είναι ένα απλό αρχείο που περιέχει το SHA-1 checksum (hash value) του commit στο οποίο δείχνει, είναι εύκολο να δημιουργούμε και να διαγράφουμε branches

Merging

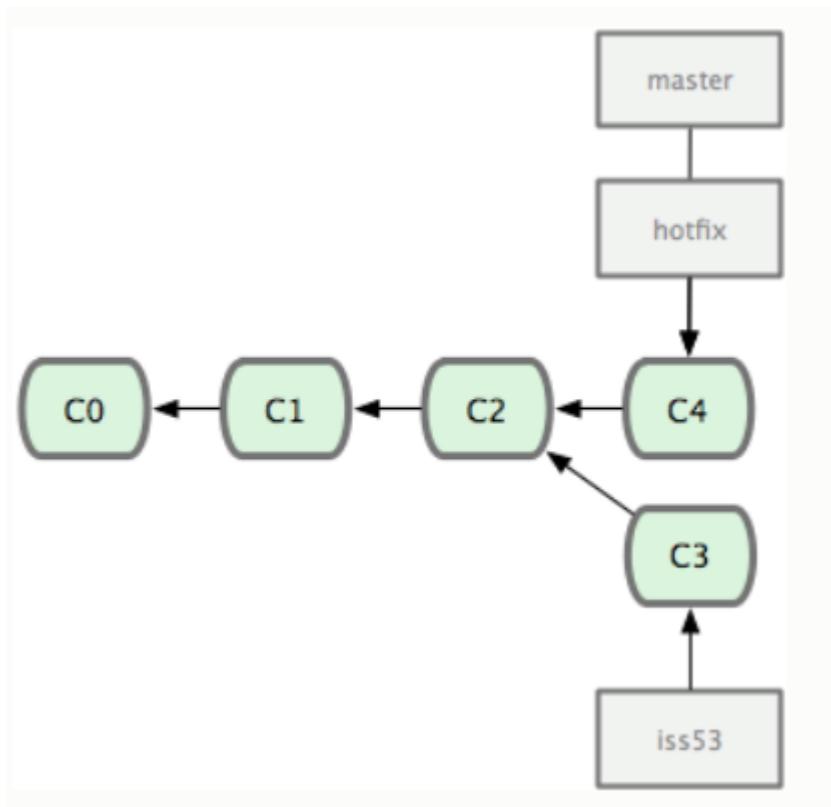
- `$ git checkout -b hotfix`
- `$ git -a -m "experiments"`



Κάνουμε πειράματα και αλλαγές στο hotfix και μετά κάνουμε merge στο master

`$ git checkout master`
`$ git merge hotfix`

Fast-forward

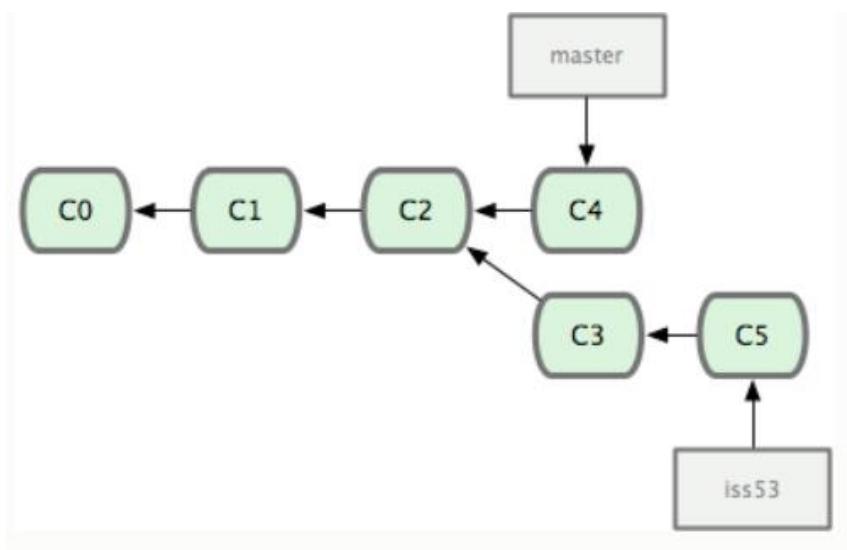


- Όταν το merge δύο commits που το ένα είναι μπροστά από το άλλο, το git απλά μετακινεί τον δείκτη του current branch μπροστά
- Αυτό ονομάζεται fast-forward

Διαγραφή branch

Ανάπτυξη Web Εφαρμογών

- Τώρα που κάναμε ένα merge με το hotfix μπορούμε να το διαγράψουμε και να συνεχίσουμε τη δουλειά μας
- `$ git branch -d hotfix`



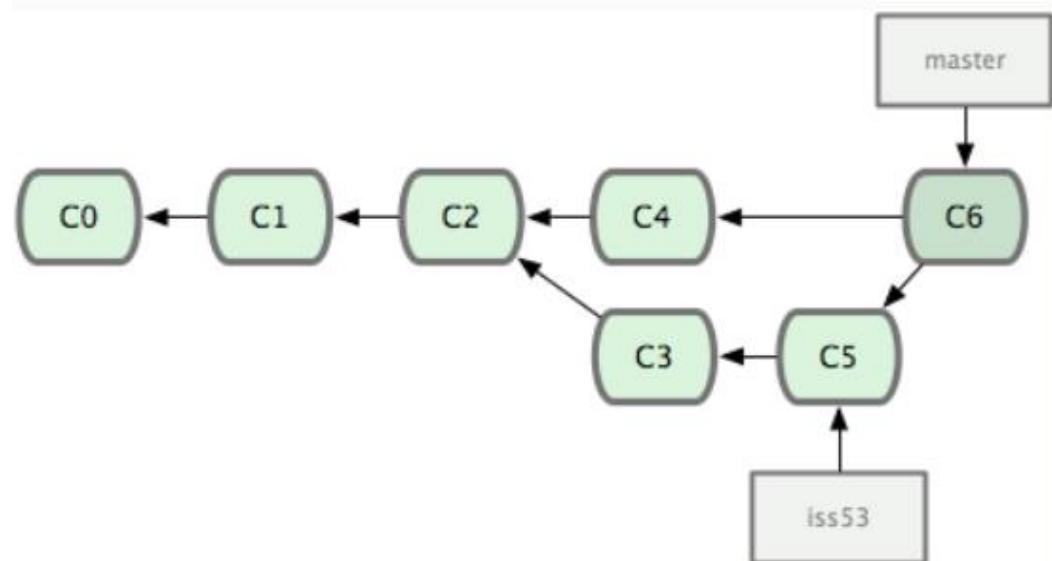
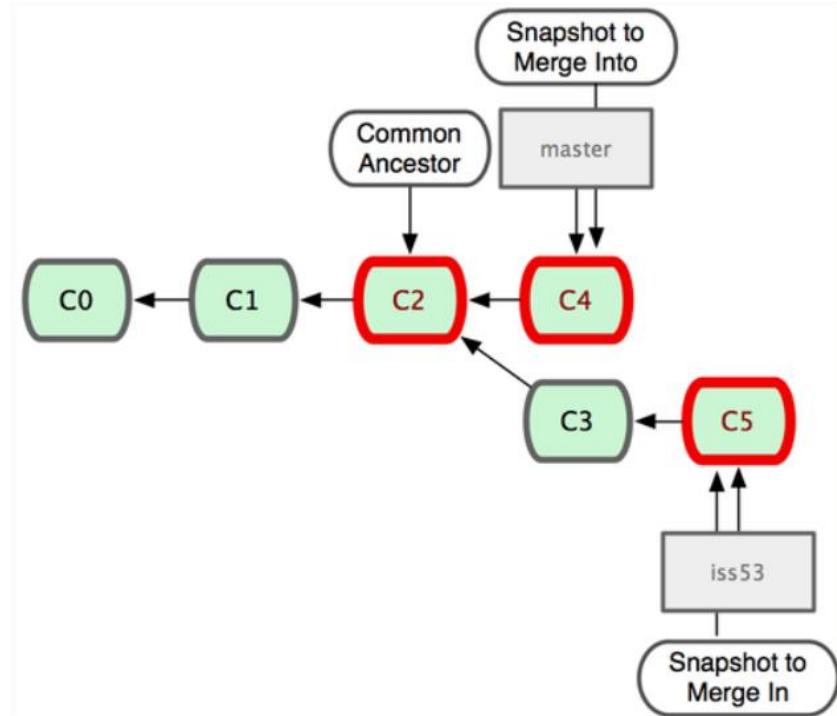
Three-way merge (1)

Ανάπτυξη Web Εφαρμογών

- Αν θέλουμε να κάνουμε merge branches που είναι σε διαφορετικά μονοπάτια, το git δημιουργεί ένα merge commit, ένα νέο δηλ. Commit βασισμένο στα δύο branches καθώς και στον κοινό τους πρόγονο

Three-way merge (2)

Ανάπτυξη Web Εφαρμογών



- Τώρα μπορούμε να διαγράψουμε το iss53