



ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ &  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

1

University of West Attica - Dr. Grigorios Koulouras

Προπτυχιακό Πρόγραμμα Σπουδών  
«Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών»

# Μάθημα «Νεφροϋπολογιστική»

Hands-on LAB05  
«Git Cheat Sheet»

Κωδικός Μαθήματος: EEE-9.2.4, EEE-9.3.7

Ακαδημαϊκό Έτος: 2023-2024

Διδάσκων:

Δρ. ΚΟΥΛΟΥΡΑΣ ΓΡΗΓΟΡΙΟΣ  
Αναπληρωτής Καθηγητής

email: [gregkoul@uniwa.gr](mailto:gregkoul@uniwa.gr)

# Περιεχόμενα

- ▶ Τι είναι το GIT;
- ▶ Εγκατάσταση του Git
- ▶ Λειτουργία του Git
  - ▶ 01. Git configuration
  - ▶ 02. Starting a project
  - ▶ 03. Day-to-day work
  - ▶ 04. Storing your work
  - ▶ 05. Git branching model
  - ▶ 06. Inspect history
  - ▶ 07. Tagging commits
  - ▶ 08. Reverting changes
- ▶ 09. Synchronizing repositories
- ▶ 10. Git installation
- ▶ 11. Ignoring files
- ▶ Δημιουργία λογαριασμού GitHub

# Τι είναι το Git;

## Ας ξεκινήσουμε

# Τι είναι το Git;



- ▶ Σε αυτό το μάθημα θα εγκαταστήσουμε το Git τοπικά στο Προσωπικό Περιβάλλον Ανάπτυξης (Personal Development Environment – PDE) που έχουμε υλοποιήσει.
- ▶ Το Git είναι ένα δωρεάν και ανοιχτού κώδικα κατανεμημένο σύστημα ελέγχου εκδόσεων. Το GitHub και το GitLab χρησιμοποιούν αυτό το εργαλείο. Το Git το χρειαζόμαστε τοπικά στον υπολογιστή μας προκειμένου να συντηρούμε τον κώδικά μας.
- ▶ Σε αυτό το Hand-on LAB θα γνωρίσουμε τις πιο σημαντικές και κοινώς χρησιμοποιούμενες εντολές Git.

# Εγκατάσταση του Git



- ▶ Το Git είναι κατά βάση εργαλείο της γραμμής εντολών, ενώ υπάρχουν και προγράμματα τα οποία παρέχουν ένα γραφικό περιβάλλον χρήστη για καθημερινή αλληλεπίδραση, ανασκόπηση και συγχρονισμό του αποθετηρίου.

# Εγκατάσταση του Git



6

Για όλες τις εκδόσεις Linux και Solaris, η τελευταία έκδοση είναι διαθέσιμη στο επίσημο Git web site.

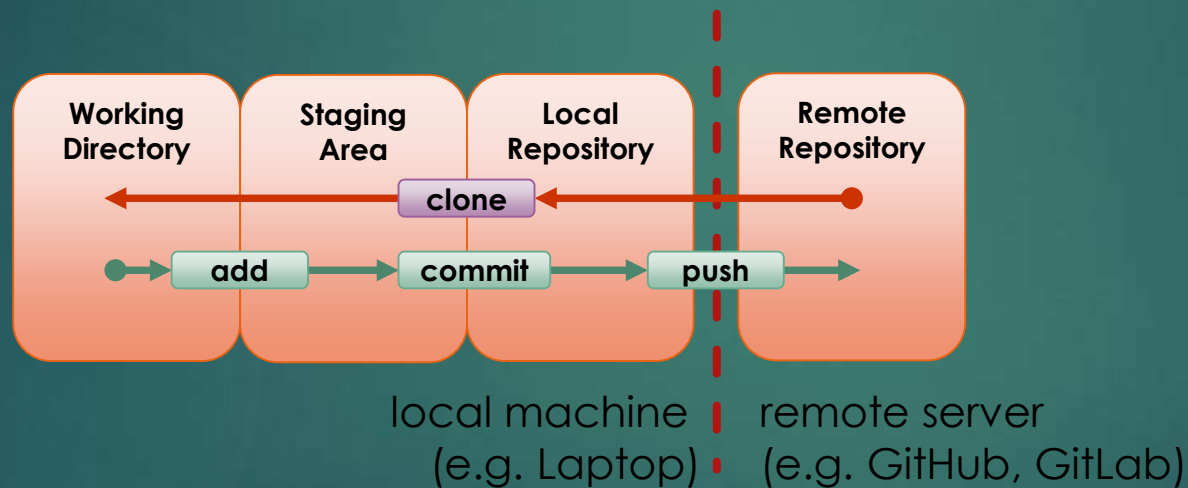
- ▶ Git για όλα τα Λειτουργικά:
  - ▶ <https://git-scm.com>
- ▶ GitHub για Windows:
  - ▶ <https://windows.github.com>
- ▶ GitHub για Mac
  - ▶ <https://mac.github.com>

# Λειτουργία του Git



7

Σε αυτή την ενότητα θα δούμε τις βασικές λειτουργίες του Git.



# 01. Git configuration



Command	Description
<b>git config --global user.name "Your Name"</b>	Set the name that will be attached to your commits and tags.
<b>git config --global user.email "you@example. com"</b>	Set the e-mail address that will be attached to your commits and tags.
<b>git config --global color.ui auto</b>	Enable some colorization of Git output.



## 02. Starting a project



Command	Description
<b>git init [project name]</b>	Create a new local repository in the current directory. If [project name] is provided, Git will create a new directory named [project name] and will initialize a repository inside it.
<b>git clone &lt;project url&gt;</b>	Downloads a project with the entire history from the remote repository.

# 03. Day-to-day work



10

Command	Description
<b>git status</b>	Displays the status of your working directory. Options include new, staged, and modified files. It will retrieve branch name, current commit identifier, and changes pending commit.
<b>git add [file]</b>	Add a file to the staging area. Use. in place of the full file path to add all changed files from the current directory down into the directory tree.
<b>git diff [file]</b>	Show changes between working directory and staging area.
<b>git diff --staged [file]</b>	Shows any changes between the staging area and the repository.
<b>git checkout -- [file]</b>	Discard changes in working directory. This operation is unrecoverable.
<b>git reset -&lt;path&gt;--.]</b>	Revert some paths in the index (or the whole index) to their state in HEAD.
<b>git commit</b>	Create a new commit from changes added to the staging area. The commit must have a message!
<b>git rm [file]</b>	Remove file from working directory and staging area.

# 04. Storing your work



11

Command	Description
<b>git stash</b>	Put current changes in your working directory into stash for later use.
<b>git stash pop</b>	Apply stored stash content into working directory, and clear stash.
<b>git stash drop</b>	Delete a specific stash from all your previous stashes.

# 05. Git branching model



12

Command	Description
<b>git branch [-a]</b>	List all local branches in repository. With -a: show all branches (with remote).
<b>git branch [branch_name]</b>	Create new branch, referencing the current HEAD.
<b>git rebase [branch_name]</b>	Apply commits of the current working branch and apply them to the HEAD of [branch] to make the history of your branch more linear.
<b>git checkout [-b] [branch_name]</b>	Switch working directory to the specified branch. With -b: Git will create the specified branch if it does not exist.
<b>git merge [branch_name]</b>	Join specified [branch_name] branch into your current branch (the one you are on currently).
<b>git branch -d [branch_name]</b>	Remove selected branch, if it is already merged into any other. -D instead of -d forces deletion.

# 06. Inspect history



13

Command	Description
<b>git log [-n count]</b>	List commit history of current branch. -n count limits list to last n commits.
<b>git log --oneline --graph --decorate</b>	An overview with reference labels and history graph. One commit per line.
<b>git log ref..</b>	List commits that are present on the current branch and not merged into ref. A ref can be a branch name or a tag name.
<b>git log ..ref</b>	List commit that are present on ref and not merged into current branch.
<b>git reflog</b>	List operations (e.g. checkouts or commits) made on local repository.

# 07. Tagging commits



14

Command	Description
<b>git tag</b>	List all tags.
<b>git tag [name] [commit sha]</b>	Create a tag reference named name for current commit. Add commit sha to tag a specific commit instead of current one.
<b>git tag -a [name] [commit sha]</b>	Create a tag object named name for current commit.
<b>git tag -d [name]</b>	Remove a tag from local repository.

# 08. Reverting changes



15

Command	Description
<b>git reset [--hard] [target reference]</b>	Switches the current branch to the target reference, leaving a difference as an uncommitted change. When --hard is used, all changes are discarded. It's easy to lose uncommitted changes with --hard.
<b>git revert [commit sha]</b>	Create a new commit, reverting changes from the specified commit. It generates an inversion of changes.

# 09. Synchronizing repositories



16

Command	Description
<b>git fetch [remote]</b>	Fetch changes from the remote, but not update tracking branches.
<b>git fetch --prune [remote]</b>	Delete remote Refs that were removed from the remote repository.
<b>git pull [remote]</b>	Fetch changes from the remote and merge current branch with its upstream.
<b>git push [--tags] [remote]</b>	Push local changes to the remote. Use --tags to push tags.
<b>git push -u [remote] [branch]</b>	Push local branch to remote repository. Set its copy as an upstream.



# 10. Git installation



17

- ▶ For GNU/Linux distributions, Git should be available in the standard system repository.
- ▶ For example, in Debian/Ubuntu please type in the terminal:
  - ▶ `sudo apt install git`

# 11. Ignoring files



18

- ▶ To ignore files, create a `.gitignore` file in your repository with a line for each pattern. File ignoring will work for the current and sub directories where `.gitignore` file is placed. In this example, all files are ignored in the `logs` directory (excluding the `.gitkeep` file), whole `tmp` directory and all files `*.swp`

```
cat <<EOF > .gitignore
/logs/*
!logs/.gitkeep
/tmp
*.swp
EOF
```

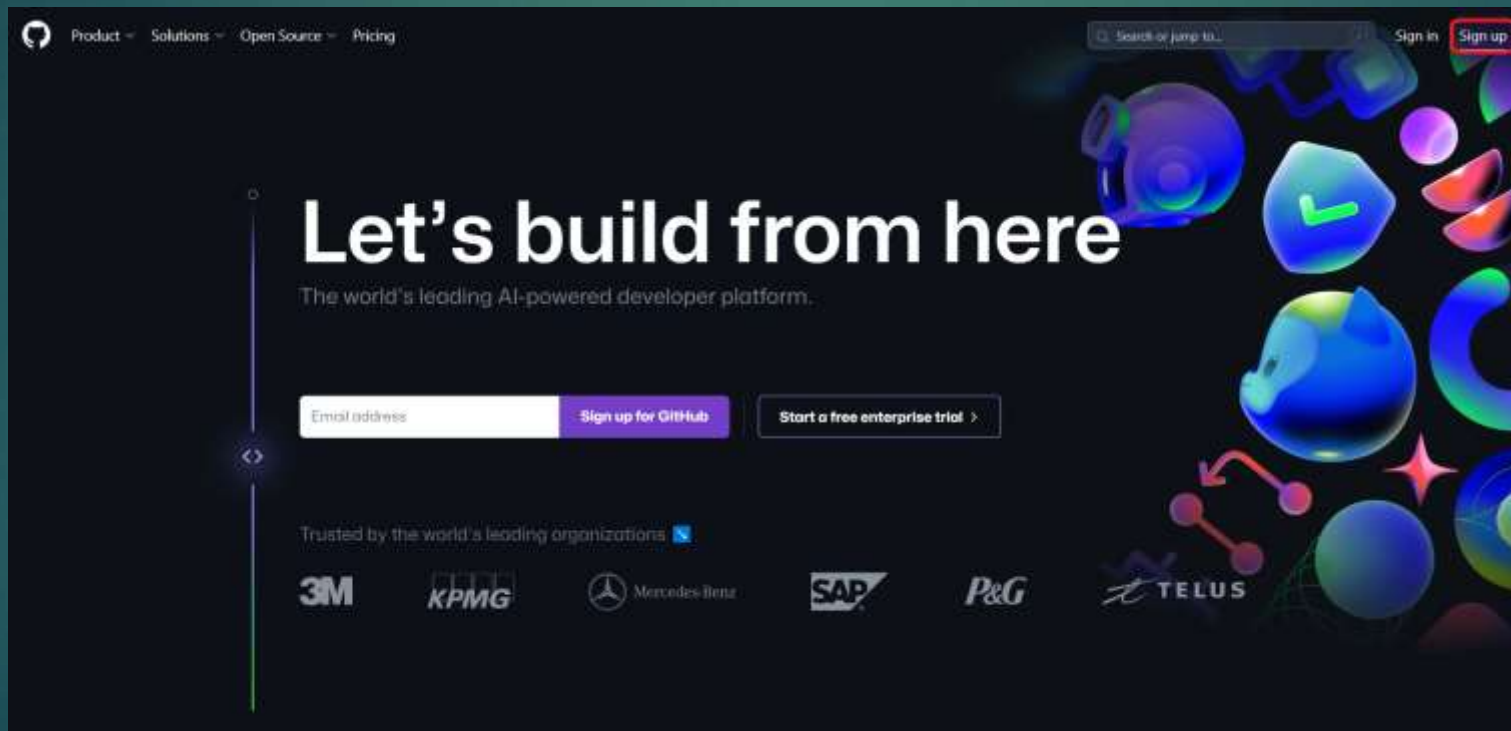
# Δημιουργία λογαριασμού GitHub



19

University of West Attica - Dr. Grigorios Koulouras

Σε αυτό το σημείο θα πρέπει να δημιουργήσουμε ένα λογαριασμό στο GitHub.



# Δημιουργία λογαριασμού GitHub



20

Ενημέρωση του λογαριασμού GitHub με το SSH public key σας.

Settings

Go to your personal profile

SSH keys

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

☐ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits. [Learn about vigilant mode](#).

SSH and GPG keys

New SSH key

New GPG key

# Επίλογος

21

- ▶ Σε αυτό το μάθημα έγινε εγκατάσταση του εργαλείου Git τοπικά στο Προσωπικό Περιβάλλον Ανάπτυξης (Personal Development Environment – PDE) που έχουμε υλοποιήσει.

# Αναφορές

22

- ▶ <https://about.gitlab.com/images/press/git-cheat-sheet.pdf>
- ▶ <https://education.github.com/git-cheat-sheet-education.pdf>