

Πηγή

<https://training.github.com/downloads/gr/github-git-cheat-sheet/>

Εγκατάσταση

Το Git είναι το σύστημα ελέγχου έκδοσης διανομής ανοιχτού κώδικα που διευκολύνει τις δραστηριότητες του GitHub στον φορητό ή επιτραπέζιο υπολογιστή σας. Αυτό το φύλλο σύντομων εντολών συνοψίζει τις οδηγίες γραμμής εντολών Git που χρησιμοποιούνται συνήθως για γρήγορη αναφορά.

GitHub Desktop

desktop.github.com

Git για όλες τις πλατφόρμες

git-scm.com

Διαμόρφωση “εργαλείων” (Configure tooling)

Ρυθμίστε τις πληροφορίες χρήστη για όλα τα τοπικά αποθετήρια (repositories)

```
$ git config --global user.name "[name]"
```

Ορίζει το όνομα που θέλετε να εμφανίζεται όταν κάνετε ένα commit

```
$ git config --global user.email "[email address]"
```

Ορίζει το email που θέλετε να εμφανίζεται όταν κάνετε ένα commit

```
$ git config --global color.ui auto
```

Εφαρμόζει χρήσιμο χρωματισμό στην γραμμή εντολών

Παρακλάδια (branches)

Τα Παρακλάδια (branches) είναι ένα σημαντικό μέρος της συνεργασίας με το Git. Οποιοσδήποτε δεσμεύσεις κάνετε θα γίνουν στο παρακλάδι (branch) στο οποίο είστε “check out”. Χρησιμοποιήστε το `git status` για να δείτε ποιο παρακλάδι (branch) είναι.

```
$ git branch [branch-name]
```

Δημιουργεί ένα νέο παρακλάδι (branch)

```
$ git switch -c [branch-name]
```

Εναλλάζει στο καθορισμένο παρακλάδι (branch) και ενημερώνει τον κατάλογο εργασίας (working directory)

```
$ git merge [branch]
```

Συνδυάζει το ιστορικό του συγκεκριμένου branch με το τρέχων παρακλάδι (branch). Αυτό γίνεται συνήθως όταν θέλουμε να υποβάλλουμε τις αλλαγές μας για έλεγχο (pull requests), αλλά είναι μια εξίσου σημαντική λειτουργία του Git.

```
$ git branch -d [branch-name]
```

Διαγράφει το καθορισμένο παρακλάδι (branch)

Δημιουργία αποθετηρίων (repositories)

Όταν ξεκινάτε με ένα νέο αποθετήριο (repository), πρέπει να το κάνετε μόνο μία φορά; είτε τοπικά, στη συνέχεια αποθηκεύστε τις αλλαγές σας (push) στο GitHub ή κλωνοποιώντας ένα υπάρχον αποθετήριο.

```
$ git init
```

Αφού χρησιμοποιήσετε την εντολή `git init`, συνδέστε το τοπικό αποθετήριο (repository) σε ένα κενό αποθετήριο GitHub χρησιμοποιώντας την ακόλουθη εντολή:

```
$ git remote add origin [url]
```

Μετατρέψτε έναν υπάρχοντα κατάλογο σε ένα αποθετήριο (repository) Git

```
$ git clone [url]
```

Κλωνοποιήστε (κατεβάστε) ένα αποθετήριο (repository) που υπάρχει ήδη στο GitHub, συμπεριλαμβανομένων όλων των αρχείων, των branches και των commits

Το αρχείο .gitignore

Μερικές φορές μπορεί να είναι καλή ιδέα να αποκλείσετε την παρακολούθηση αρχείων από το Git. Αυτό συνήθως γίνεται σε ένα ειδικό αρχείο που ονομάζεται .gitignore.

Μπορείτε να βρείτε χρήσιμα πρότυπα για αρχεία .gitignore στη διεύθυνση github.com/github/gitignore.

Συγχρονισμός αλλαγών

Συγχρονίστε το τοπικό αποθετήριο (repository) σας με το απομακρυσμένο αποθετήριο (repository) στο GitHub.com

```
$ git fetch
```

Κατεβάζει όλο το ιστορικό από τα απομακρυσμένα παρακλάδια (branches)

```
$ git merge
```

Συνδυάζει απομακρυσμένα παρακλάδια (branches) με το τοπικό παρακλάδι (branch)

```
$ git push
```

Ανεβάζει όλα τα τοπικά παρακλάδια (branch) commits στο GitHub

```
$ git pull
```

Ενημερώνει το τρέχοντα τοπικό παρακλάδι (branch) με όλα τα νέα commits από το αντίστοιχο παρακλάδι (branch) στο GitHub. Το git pull είναι ένας συνδυασμός του git fetch και του git merge

Κάνοντας αλλαγές

Περιηγηθείτε και ελέγξτε την εξέλιξη των αρχείων του project

```
$ git log
```

Εμφανίζει το ιστορικό έκδοσης για το τρέχοντα παρακλάδι (branch)

```
$ git log --follow [file]
```

Εμφανίζει το ιστορικό εκδόσεων για ένα αρχείο, συμπεριλαμβανομένων των μετονομασιών του

```
$ git diff [first-branch]...[second-branch]
```

Εμφανίζει διαφορές περιεχομένου μεταξύ δύο παρακλαδιών (braches)

```
$ git show [commit]
```

Εμφανίζει μεταδεδομένα και αλλαγές περιεχομένου του καθορισμένου commit

```
$ git add [file]
```

“Αποθηκεύει” (snapshots) το αρχείο προετοιμάζοντας το για ανέβασμα (versioning)

```
$ git commit -m "[descriptive message]"
```

“Αποθηκεύει” (snapshots) μόνιμα τα αρχεία στο ιστορικό εκδόσεων

Επαναφέροντας commits

Διαγράφει τυχόν λάθη και δημιουργεί μια αντικατάσταση ιστορικού

```
$ git reset [commit]
```

Επαναφέρει όλα τα commits μετά το [commit], διατηρώντας όλες τις αλλαγές τοπικά

```
$ git reset --hard [commit]
```

Απορρίπτει όλο το ιστορικό και επιστρέφει πίσω στο καθορισμένο commit

ΠΡΟΣΟΧΗ! Η αλλαγή του ιστορικού μπορεί να έχει δυσάρεστα αποτελέσματα. Εάν πρέπει να αλλάξετε commits που υπάρχουν στο GitHub (το remote), προχωρήστε με προσοχή. Εάν χρειάζεστε βοήθεια, επικοινωνήστε με την [github.community](https://github.com) ή επικοινωνήστε με την υποστήριξη.