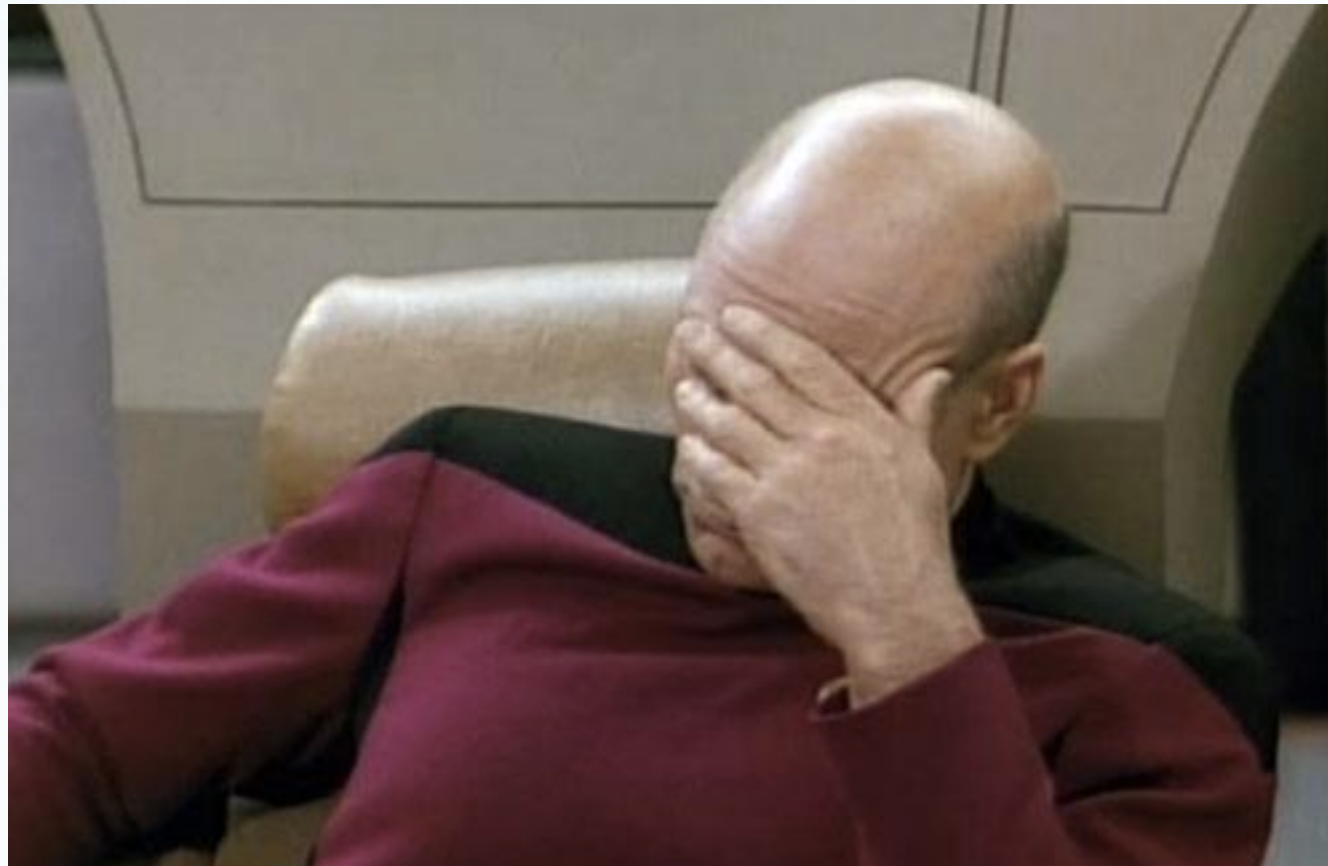




НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

**forgot to plug the laptop in the outlet**



**30 hours of my life - lost forever...**



# GIT

# Какво представлява

- Distributed Version Control System (DVCS)
- Т.е. система за разпределена разработка на софтуер и за контрол на версиите
- Или на човешки език - нещо, което ни пази бекъп на работата и ни позволява да работим (променяме) няколко човека едновременно по едни и същи ресурси (файлове, документи, програмен код и т.н.)

# Защо искаме да ползваме GIT

- Гъвкава система за управление на версиите на кода (***version control***)
- Позволява дистанционно и централизирано съхраняване на проектите (*споделени хранилища, remote origin*)
- Позволява големи групи от софтуерни разработчици да работят едновременно върху споделени проекти (*distributed development*)



# Version Control

Това, случайно ...



... да ви е познато?



- Докато работим, постоянно правим промени по кода
- Понякога отделяме часове за да направим някаква промяна, която после решаваме да отхвърлим
- Ако не сме си направили backup преди да започнем въпросната промяна, ще ни се наложи да отделим ужасно много време, за да върнем всичко така както си е било преди
- Може и никога да не успеем да възстановим системата в положението в което последно е работила добре
- Ако имаме контрол на версиите на нашата система/проект и знаем как да го ползваме - гореописаният проблем няма как да се случи



```
mkdir my_project
```

```
cd my_project
```

```
git init
```

```
# develop things that work
```

```
git commit -all -message "my project in a working state"
```

```
# develop things that are completely wrong
```

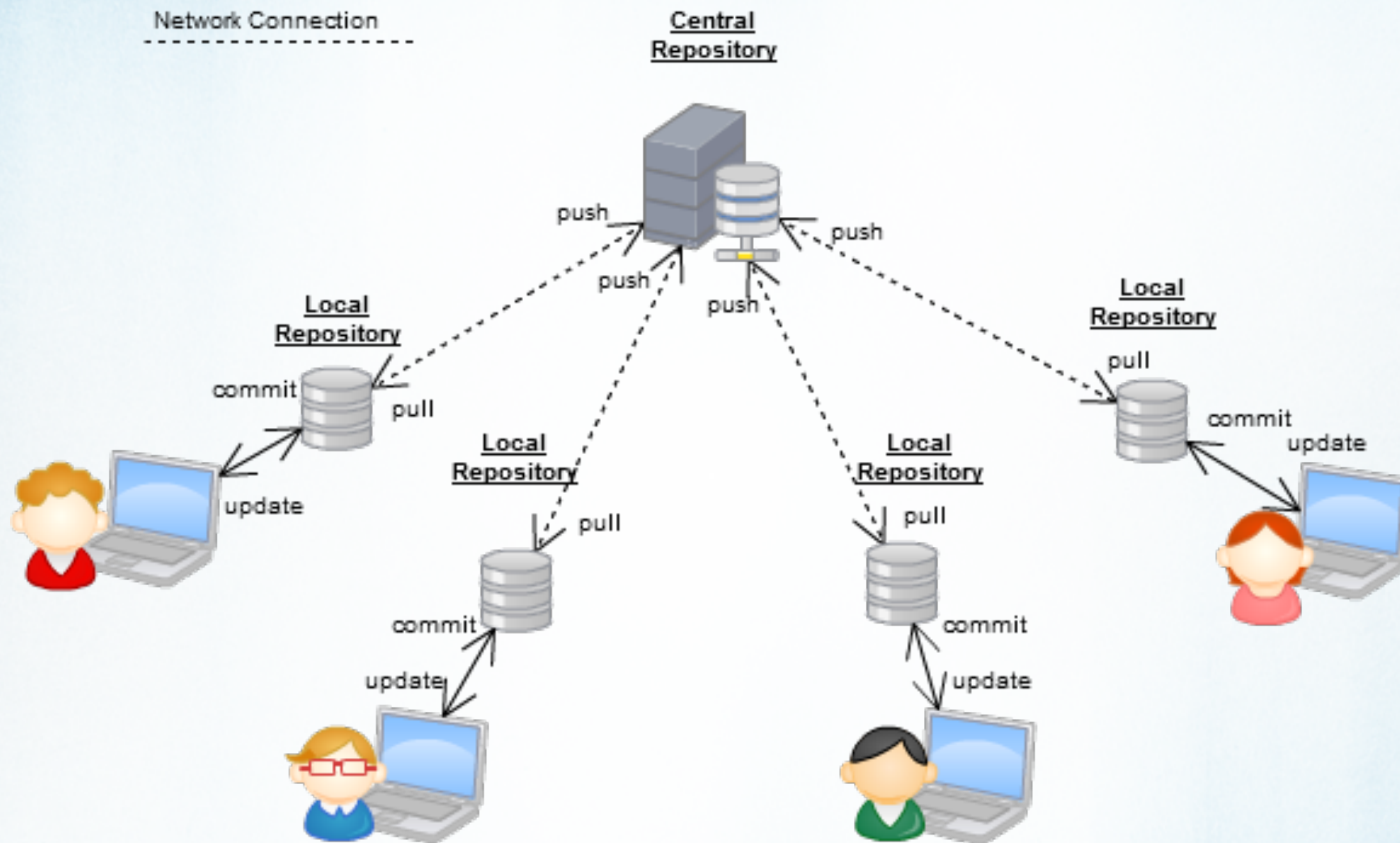
```
git checkout .
```

```
# now the state of my project is restored
```

# remote origin



- Това е url адреса на мястото, където се съхранява централното (т.е. главното) хранилище (*repository*) на нашия проект
- Репозиторието е *remote* (отдалечено), защото ако беше разположено само на нашият компютър, щеше да е недостъпно за останалият свят
- Нарича се *origin*, защото всички, които работят по проекта, ъпдейт-ват своите проекти от там
- Централното репозитори е просто едно от копията на проекта, но се счита за първоизточник и всички останали се ъпдейтват от него
- По този начин всички които работят по проекта са синхронизирани (имат винаги последната версия на кода)





- Проекта представлява набор от файлове (например файловете на един уебсайт). Хората, които работят по този проект може да работят по различни части от него, но може и да работят по едни и същи файлове
- За да може всички да разполагат с най-актуалният код на проекта, всеки трябва веднага да качва в централно репозитори, промените които е направил
- Също така всички трябва редовно да ъпдейтват своите копия
- Принципа на работа е:
  - ъпдейт: **git pull**
  - edit files on my local copy (***brackets***)
  - пак ъпдейт: **git pull**
  - запис на промените ми в локалното копие (create changeset): **git commit**
  - upload to remote origin: **git push**

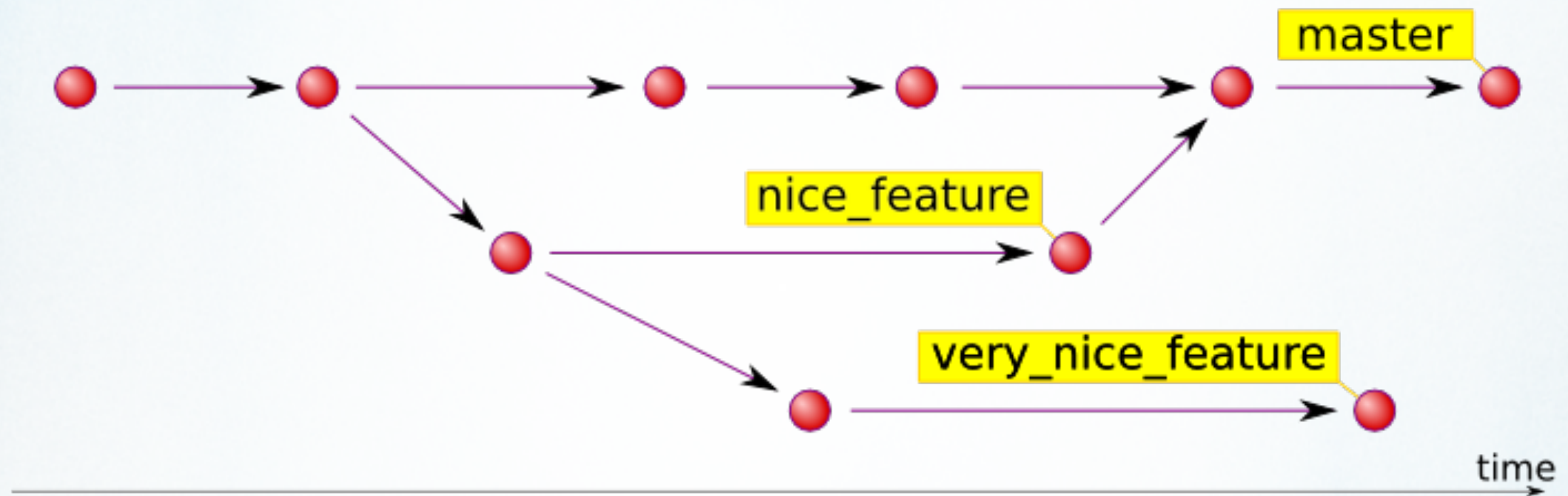
# Хранилища

- Мястото, на което се намира централното репозитори (remote origin-a), наричаме *git server*
- Както web server-a предоставя web услуга, така - git server-a, предоставя git услуга
- Един такъв сървър е GitHub
- <https://github.com>



# Distributed development

# Branching





- Всеки програмист разполага със свое копие на проекта, което може да ъпдейтва по всяко време от origin:

```
git pull # (git)
```

```
sync (GitHub Desktop)
```

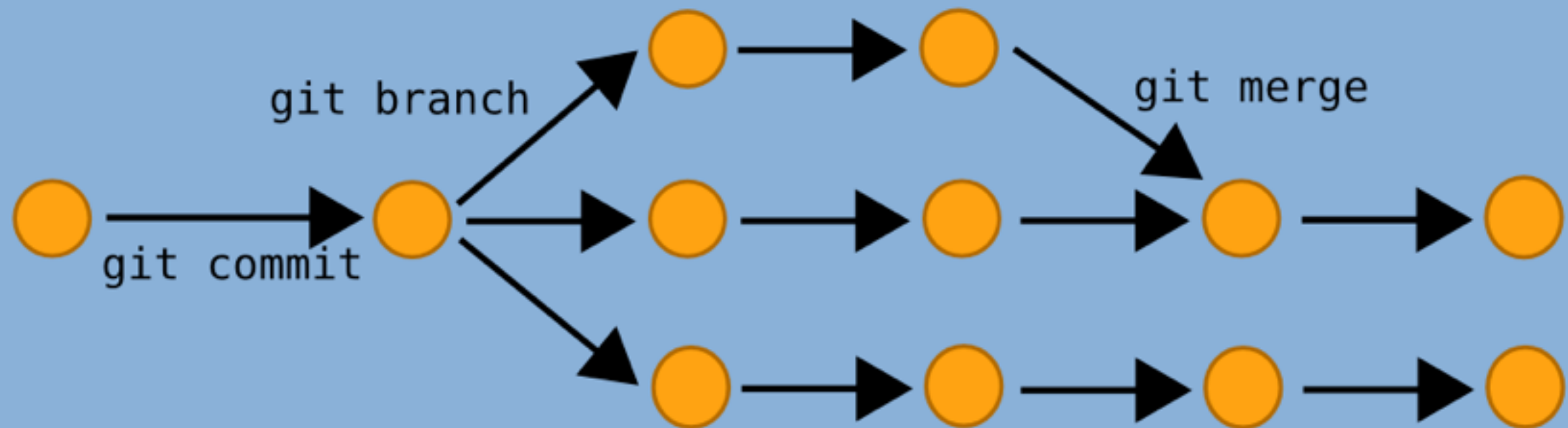
- Всеки програмист може да работи в свой *branch*, така че да не пречи на останалите и да може да следи версиите на кода си:

```
git branch new_feature
```

- Всеки “готов” feature или bugfix може лесно бъде “инжектиран” в главния branch (master) чрез *merge*:

```
git co master && git merge new_feature
```

## Repository





# Има ли други варианти за DVCS освен GIT?

- Да
- Подобни системи са SubVersion, Mercurial, дори и Dropbox
- По различни съображения в различните софтуерни разработки се използват различни DVCS
- И все пак през годините GIT успя да се наложи като най-доброто решение за DVCS
- Повече за приликите и разликите между тези системи тук:  
<http://programmers.stackexchange.com/questions/35074/im-a-subversion-geek-why-should-i-consider-or-not-consider-mercurial-or-git-or>



**github**  
SOCIAL CODING



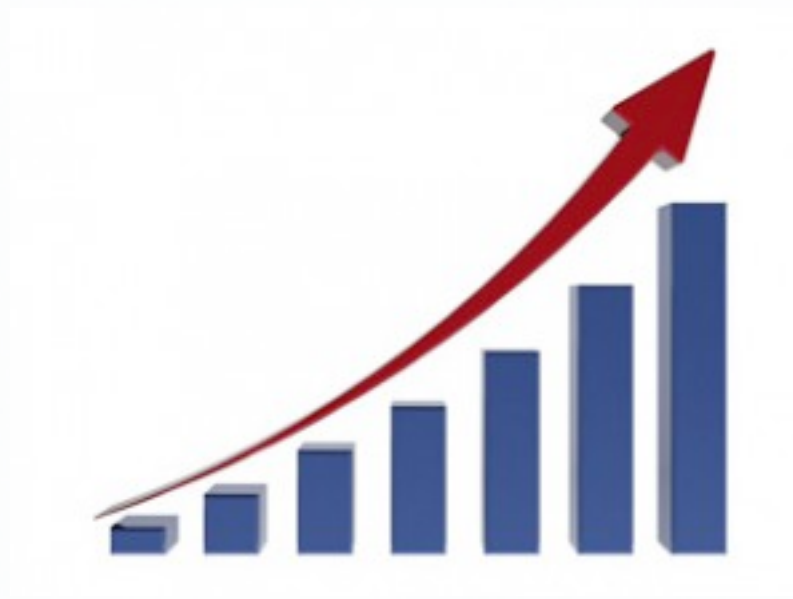
GitHub е №1 hosting за **open source** проекти

# No 1 Project hosting (Wikipedia)

Name ⇄	Users ⇄	Projects ⇄	Alexa rank (lower = more popular) ⇄
<b>Alioth</b>	22,731 <sup>[52]</sup>	1,106 <sup>[52]</sup>	N/A (subdomain not tracked)
<b>Assembla</b>	800,000+ <sup>[53]</sup>	60,000+ <sup>[54]</sup>	6,010 as of 3 July 2014 <sup>[55]</sup>
<b>BerliOS</b>	52,811 <sup>[56]</sup>	4,863 <sup>[56]</sup>	54,107 as of 3 July 2014 <sup>[57]</sup>
<b>Bitbucket</b>	2,500,000	93,661 <sup>[58]</sup>	2,696 as of 3 July 2014 <sup>[59]</sup>
<b>CodePlex</b>	151,782	36,472 <sup>[60]</sup>	2,392 as of 3 July 2014 <sup>[61]</sup>
<b>Fedora Hosted</b>	?	411 <sup>[62]</sup>	
<b>GitHub</b>	10,600,000 <sup>[63]</sup>	26,200,000 <sup>[63][n 1]</sup>	91 as of 24 August 2015 <sup>[64]</sup>
<b>GitLab</b>	20,000 <sup>[65]</sup>	100,000+ <sup>[65][n 1]</sup>	8,563 as of 21 September 2015 <sup>[66]</sup>
<b>Gna!</b>	8,511	1,437	130,683 as of 3 April 2015 <sup>[67]</sup>
<b>GNU Savannah</b>	67,183 <sup>[68]</sup>	3,696 <sup>[68]</sup>	61,710 as of 4 October 2015 <sup>[69]</sup>
<b>GNOME Git Repositories</b>	Unknown <sup>[n 13]</sup>	600+	N/A (subdomain not tracked)
<b>Google Code</b>	Unknown <sup>[n 13]</sup>	250,000+ <sup>[70]</sup>	N/A (subdomain not tracked)
<b>JavaForge</b>			816,023 as of 4 April 2015 <sup>[71]</sup>
<b>Launchpad</b>	2,145,028 <sup>[72]</sup>	32,699 <sup>[73]</sup>	5,618 as of 3 July 2014 <sup>[74]</sup>
<b>OSDN</b>	50,871 <sup>[75]</sup>	5,952 <sup>[75]</sup>	16,814 as of 17 June 2015 <sup>[76]</sup>
<b>Ourproject.org</b>	Unknown <sup>[n 13]</sup>	1,411 <sup>[77]</sup>	N/A (subdomain not tracked)
<b>SourceForge</b>	3,400,000+ <sup>[78]</sup>	324,000 <sup>[78]</sup>	213 as of 18 February 2015 <sup>[79]</sup>



- Над 11М потребители
- Над 28М проекта (repositories) от които над 5 милиона са *open source*

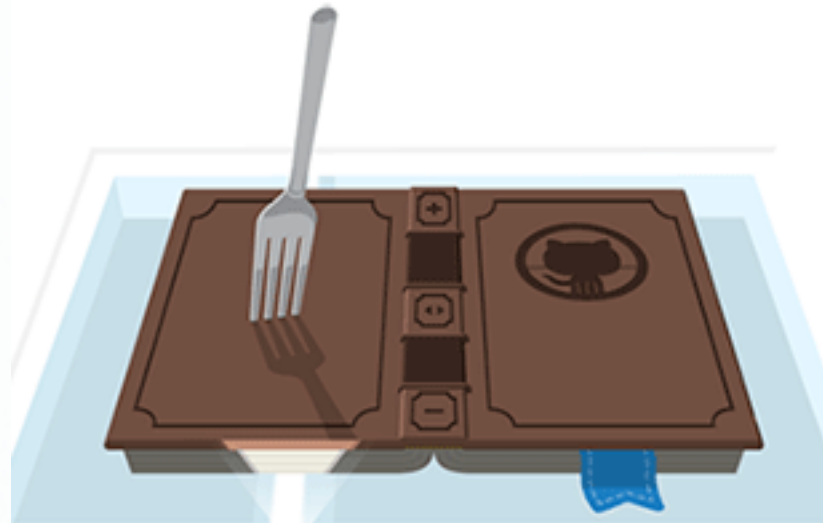


# Защо искаме да работим с GitHub?

- Безплатен GIT хостинг (за public акаунти)
- Удобен web интерфейс + десктоп програма
- Система за следене на бъгове (Issue tracker)
- Преглед на кода (code review)
- Дава възможност за участие в проекти с отворен код (contribution to open source) - fork & pull request
- **Най-голямата open source общност (*open source community*) !**



# Fork it!!!



## REMOTE

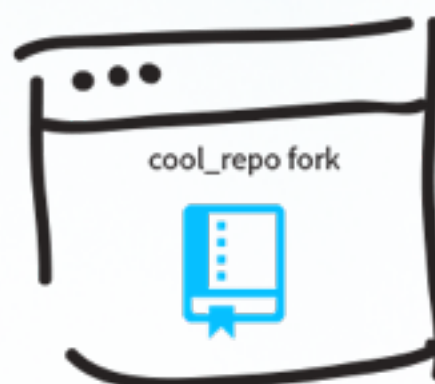
Someone else's repository.



Fork!

## REMOTE

Your fork of the repository.



Clone to  
your computer  
from GitHub.

Push and Pull to your  
fork 'origin'.

Pull from 'upstream'  
changes to original.



## LOCAL

Use your computer's  
**terminal** to talk to  
two repositories via  
**two remotes** to the  
GitHub servers.



[http://jlord.us/git-it/challenges/forks\\_and\\_clones.html](http://jlord.us/git-it/challenges/forks_and_clones.html)

# Въпроси?



**ВРЕМЕ Е ЗА УПРАЖНЕНИЯ!!**



# CSS positioning

- fixed
- absolute
- relative
- static
- z-index
- [http://www.w3schools.com/css/css\\_positioning.asp](http://www.w3schools.com/css/css_positioning.asp)



# Advanced layout (video background)

- Свалете:  
<http://zenlabs.pro/courses/lessons/lesson5/examples.zip>
- разархивирайте го и го отворете като нов проект в Brackets (File -> Open folder)
- В header-а сложете един audio елемент, като линкнете mp3 файла от проекта
- В body сложете видео елемент, като вмъкнете 2-та видео файла от проекта

# Полезни връзки

- <https://www.freesound.org/people/Sandermotions/sounds/278865/>
- <http://www.videezy.com/water/2396-raindrops-on-a-window-royalty-free-hd-stock-video>
- <http://audio.online-convert.com/>
- <http://video.online-convert.com/>
- Пример:  
<http://zenlabs.pro/courses/lessons/lesson5/examples/index.html>



# Git

- направете си акаунт в github
- направете **fork** на swift-academy-homeworks repository-то:  
<https://github.com/zzeni/swift-academy-homeworks>
- инсталирайте си Github Desktop програмата и се логнете в нея с акаунта ви от Github
- Отворете Github Desktop и изберете 'Clone' от бутона '+' горе вляво
- Изберете swift-academy-homeworks
- Клонирайте я в работната си папка, в My Documents или на десктопа (където ви е най-удобно)

# Git

- Отстрани в програмата ще ви се появи swift-academy-homeworks
- Можете да кликнете върху него с десен бутон на мишката и да дадете Show in Explorer
- Отворете Brackets и добавете новата папка като отделен проект (File - > Open folder)
- Създайте нова папка с име - вашето малко име (без главни букви)
- в нея - още 4 нови папки L1, L2, L3, L4
- във всяка копирайте съответното домашно (ако ви липсва домашно, може просто да го добавите по-късно като го направите)



# Git

- Върнете се в Github Desktop-а и кликнете на Changes от заглавната лента
- Би трябвало да видите всичките файлове от домашните ви
- Моля уверете се, че няма файлове извън вашата папка и че вашата папка не се намира в папка *tasks*
- Най отдолу има 2 инпут полета: *Summary* и *Description*
- В *Summary* напишете: Initial commit, а в *Description*: My first commit
- Натиснете бутона Commit отдолу

# Git

- горе вдясно ще ви се появят бутони и *Sync* и *Pull-request*
- натиснете първо *Sync*, после *Pull-request* и потвърдете
- кликнете отново с десния бутон върху swift-academy-homeworks
- изберете View on GitHub
- при натискане, би трябвало да ви се отвори гитхъб в браузъра и да видите вашето копие на swift-academy-homeworks репозиторията
- уверете се че файловете ви са качени в проекта в браузъра
- изпратете ми линк към проекта ви



# Домашно

<https://github.com/zzeni/swift-academy-homeworks/blob/fe-03/tasks/L05>