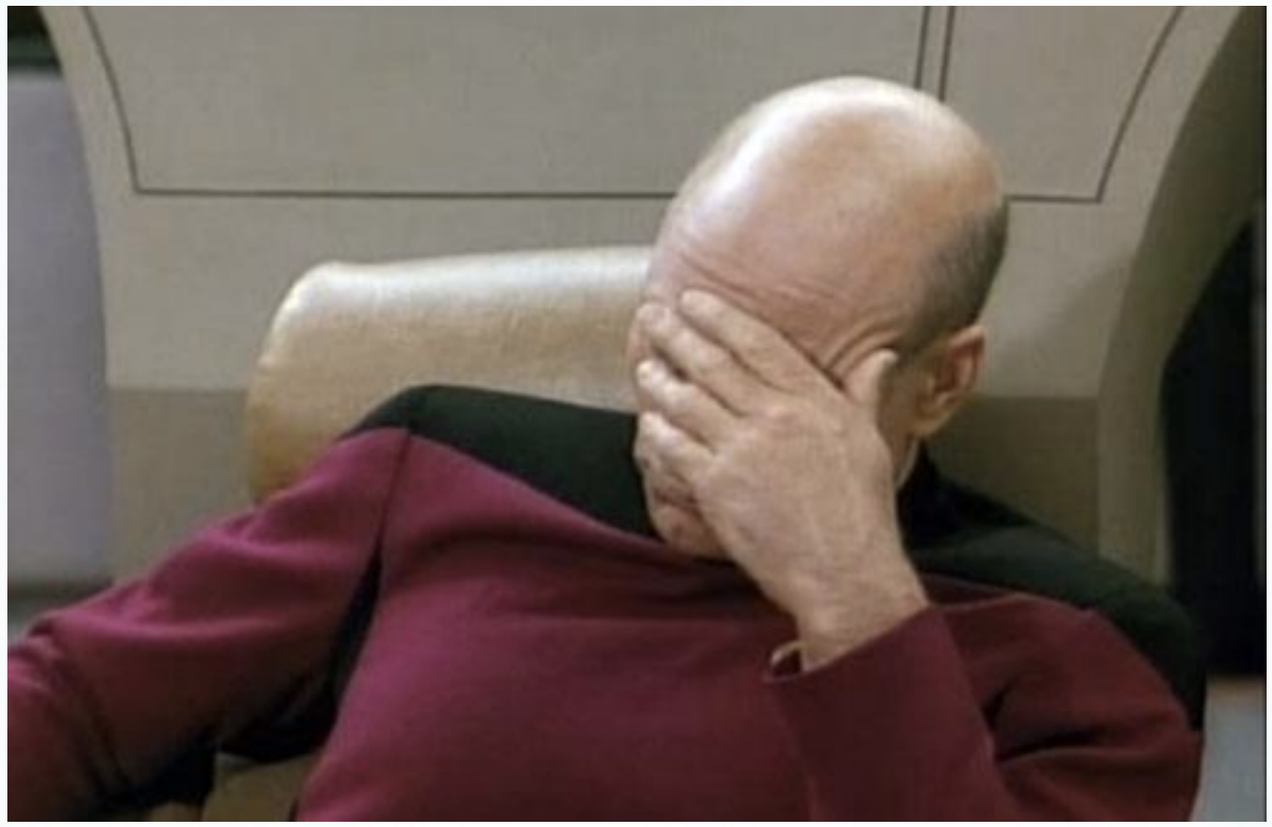
SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

forgot to plug the laptop in the outlet



30 hours of my life - lost forever...

GIT

SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Front-End Development

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

SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Front-End Development

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

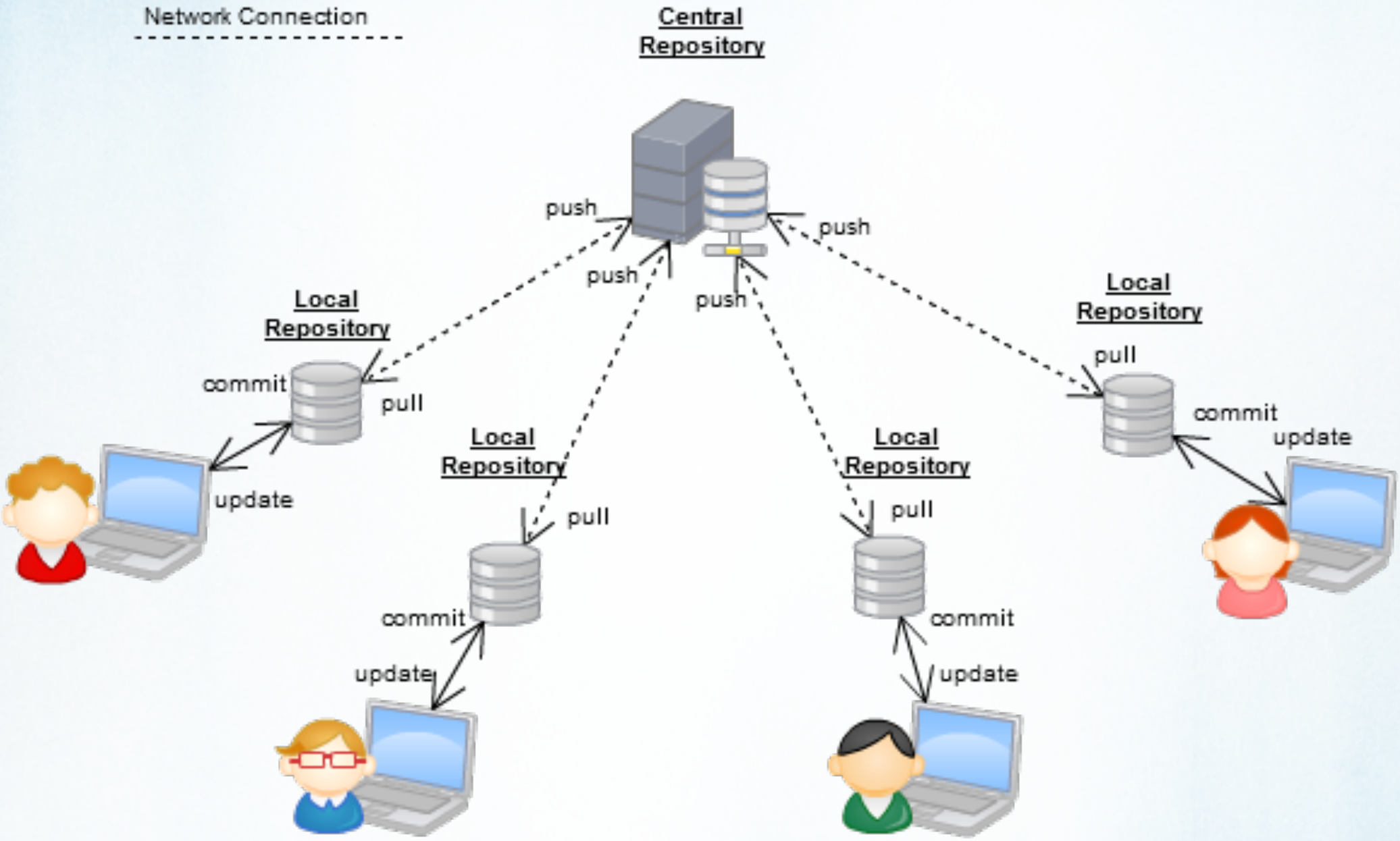
- Това е url адреса на мястото, където се съхранява централното (т.е. главното) хранилище (*repository*) на нашия проект

- Репозиторито 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)


Network Connection
Central Repository
push
push
push
push
Local Repository
commit
pull
update
Local Repository
pull
commit
update
Local Repository
pull
commit
update
Local Repository
pull
commit
update

- 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-а), наричаме *git server*

- Както web server-а предоставя web услуга, така - git server-а, предоставя git услуга

- Един такъв сървър е GitHub

- https://github.com

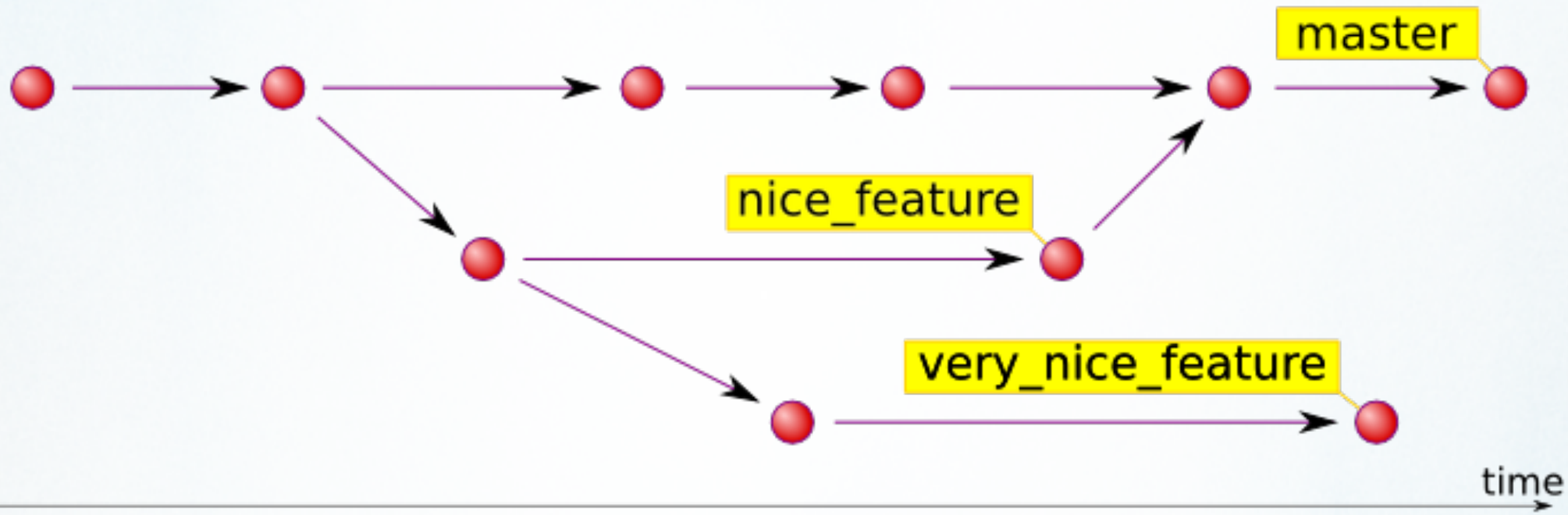
Distributed development

SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Front-End Development

Branching



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

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

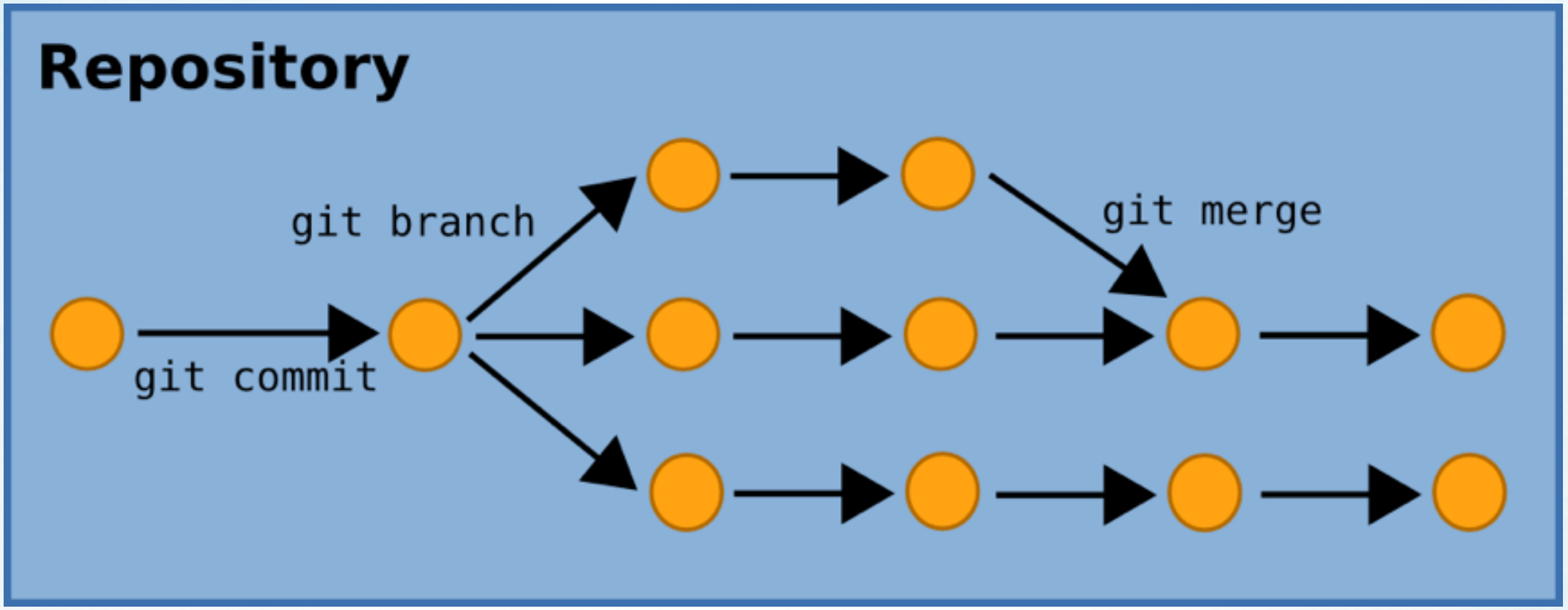
  sync (GitHub Desktop)

- 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
git branch
git merge
git commit

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

SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Front-End Development

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

SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Front-End Development

No 1 Project hosting (Wikipedia)

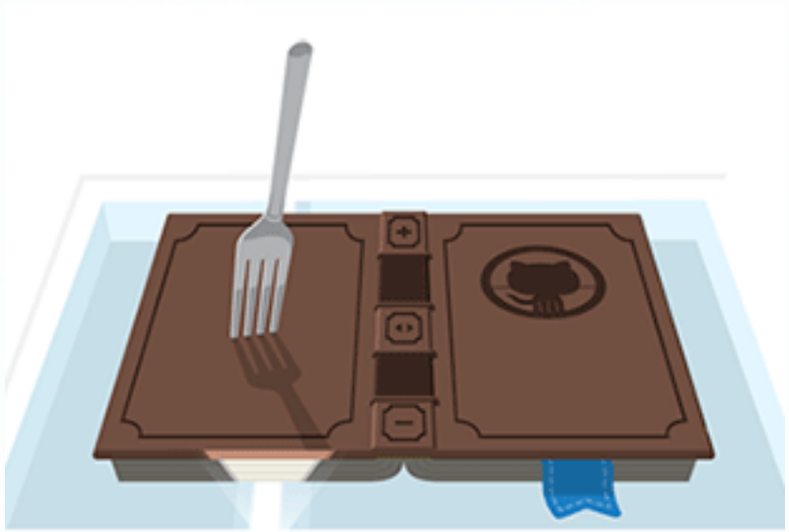
Name	Users	Projects	Alexa rank (lower = more popular)
Alioth	22,731[52]	1,106[52]	N/A (subdomain not tracked)
Assembla	800,000+[53]	60,000+[54]	6,010 as of 3 July 2014[55]
BerliOS	52,811[56]	4,863[56]	54,107 as of 3 July 2014[57]
Bitbucket	2,500,000	93,661[58]	2,696 as of 3 July 2014[59]
CodePlex	151,782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

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

Защо искаме да работим с GitHub?

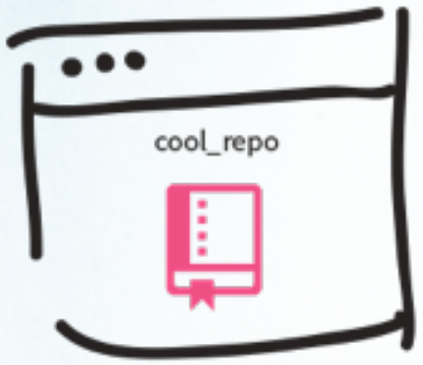
- Безплатен GIT хостинг (за public акаунти)
- Удобен web интерфейс + десктоп програма
- Система за следене на бъгове (Issue tracker)
- Преглед на кода (code review)
- Дава възможност за участие в проекти с отворен код (contribution to open source) - fork & pull request
- **Най-голямата open source общност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

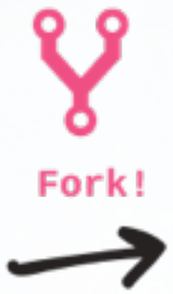
Fork it!!!



REMOTE

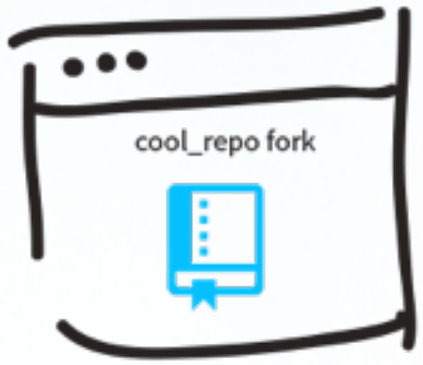
Someone else's repository.





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

Въпроси?

ВРЕМЕ Е ЗА УПРАЖНЕНИЯ!!

CSS positioning

- fixed
- absolute
- relative
- static
- z-index
- 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

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 репозиторито
- 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