

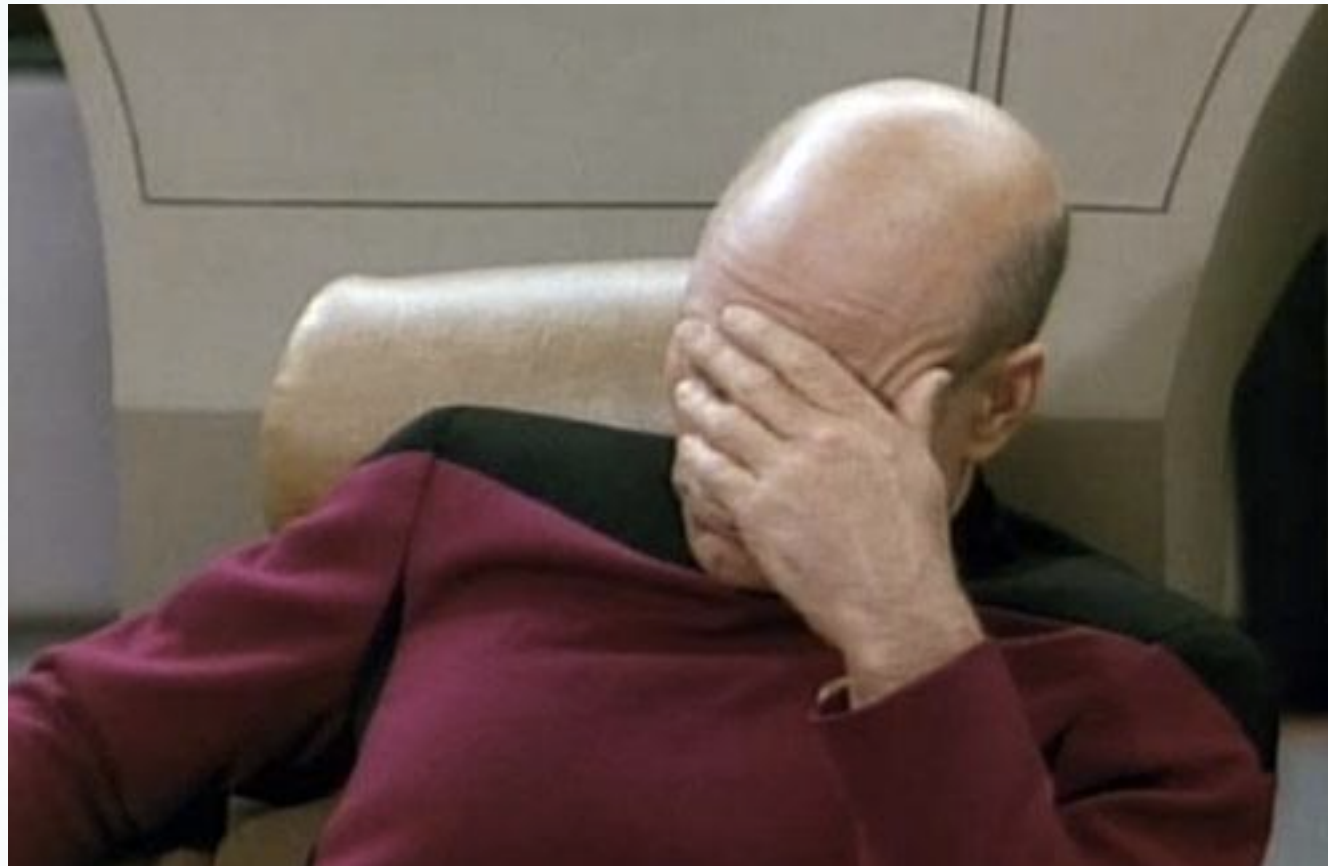


НИ Е ВЪРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

# GIT



**accidentally deleted the project file**



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

# Какво е GIT

*"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency."*

– <https://git-scm.com/>



# Какво е DVCS

- Distributed Version Control System
- 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 няколко човека едновременно да работим (променяме) по едни и същи ресурси (файлове, документи, програмен код и т.н.)
- GIT една такава система

# Какво е "контрол на версиите"

- DVCS ни дават възможност във всеки един момент да си направим "сейв" (от save в компютърните игри) на текущото състояние на файла/проекта
- Системата пази всеки един "сейв", за да може по-късно, ако решим - да се върнем към него
- По този начин, ако внезапно нещо много се обърка с проекта, можем да възстановим нещата в "последното работещо състояние"
- Тези "сейвове" се наричат версии (както и: commit, changeset)



Това, случайно ...

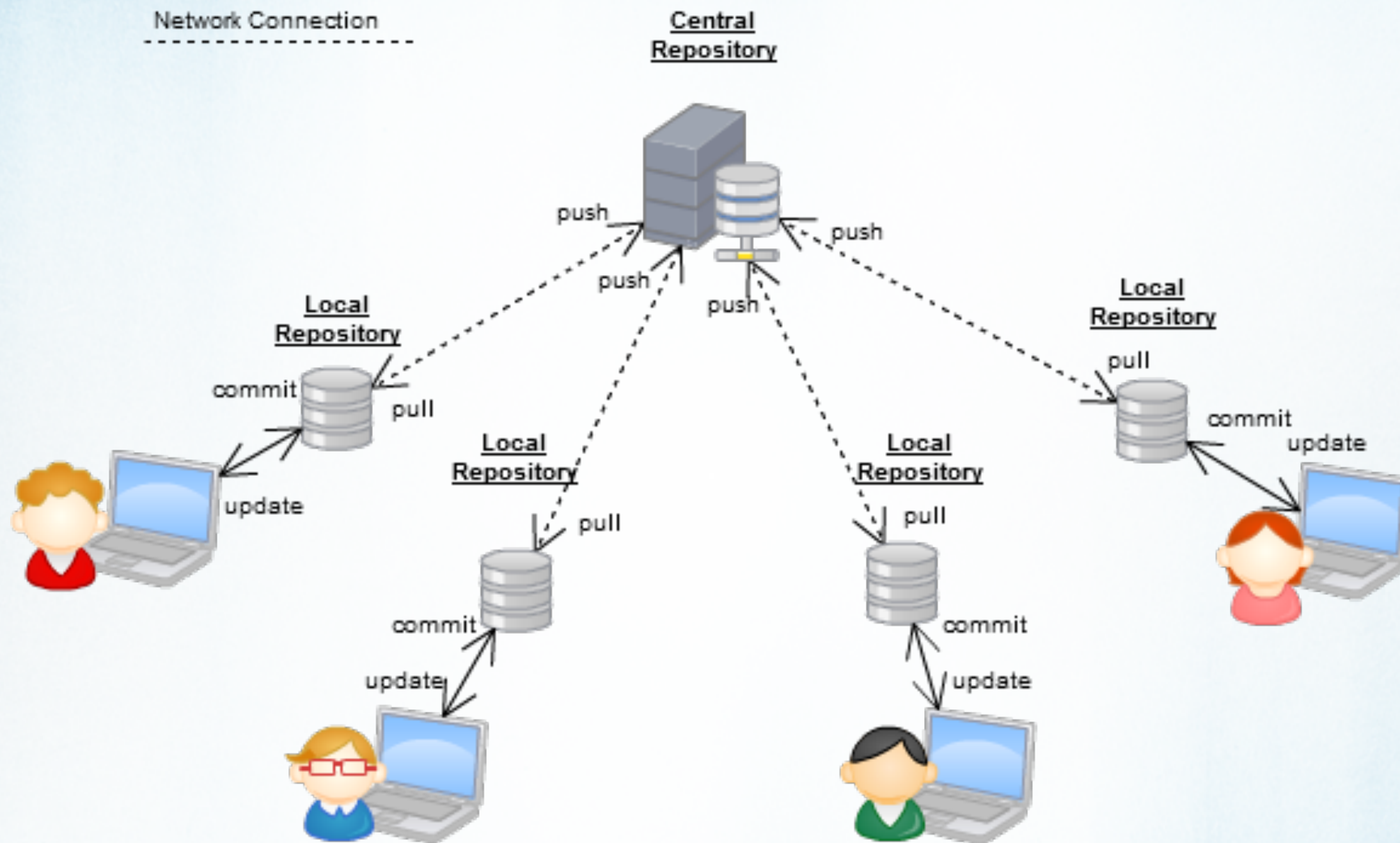


... да ви е познато?

# Какво е "разпределена разработка"

- Когато няколко човека работят едновременно по един и същ проект (както е в големите фирми)
- Основният проблем е, че те имат нужда постоянно да синхронизират работата си, за да не се окаже че единият е "затрил" промените на другия
- Логиката на това синхронизиране е следната: Има едно място, достъпно за всички, което се нарича *централно хранилище (central repository/remote origin)*  
Там се пази винаги най-актуалната версия на проекта.





# Какво е хранилище

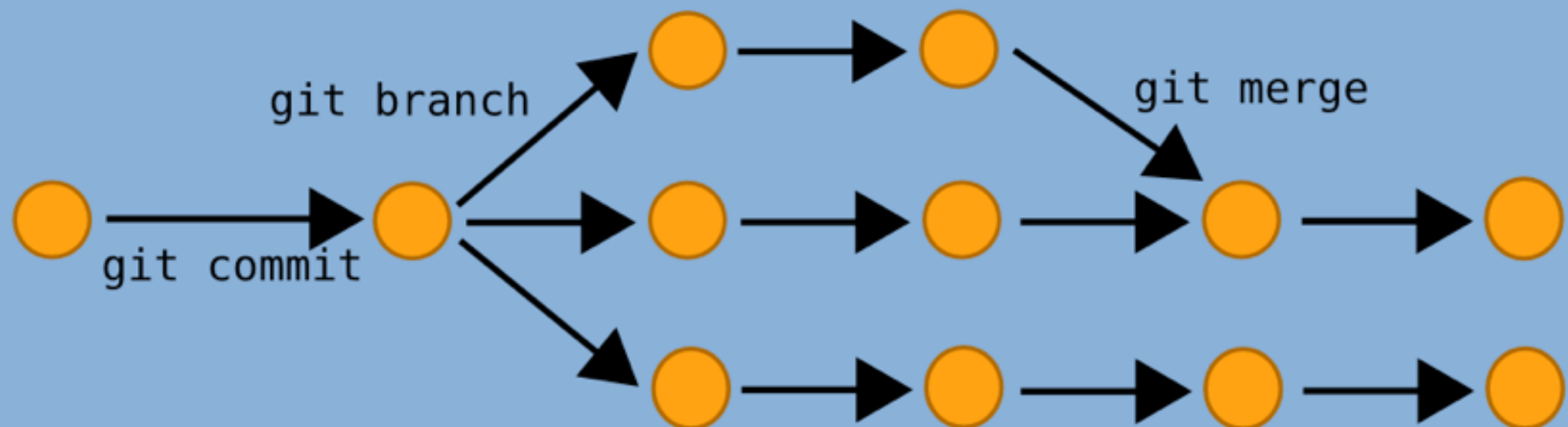
- Хранилище или репозитори, наричаме всяко едно копие на проекта, което е свързано със системата за DVC
- Всеки един разработчик разполага с такова копие
- Има едно специално хранилище, което се нарича *централно хранилище* и то се счита за първоизточника на всички копия (remote origin)
- То се намира на GIT сървър и служи за синхронизиране на всички копия



# Основни Git команди и действия

- **pull** (в *Github Desktop* е *sync*) - изтегляне на всички нови промени от централното репозитори
- **commit** - запис на текущото състояние на проекта
- **push** (в *Github Desktop* е *publish*) - изпращане на направените промени към сървъра
- **merge** - "смесване" на промените на различните хора
- **pull request** - заявка за приемане на промените

## Repository





# Други системи за DVC

- SubVersion
- Mercurial
- Dropbox
- GitLab
- Интересна дискусия за качествата на различните DVCS:  
<http://programmers.stackexchange.com/questions/35074/im-a-subversion-geek-why-should-i-consider-or-not-consider-mercurial-or-git-or>



**github**  
SOCIAL CODING



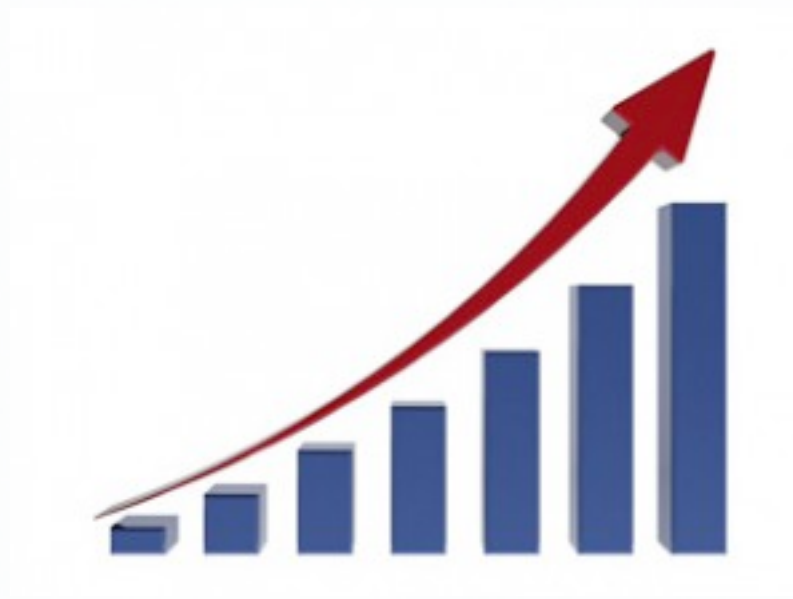
GitHub е №1 hosting за **open source** проекти

# 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*

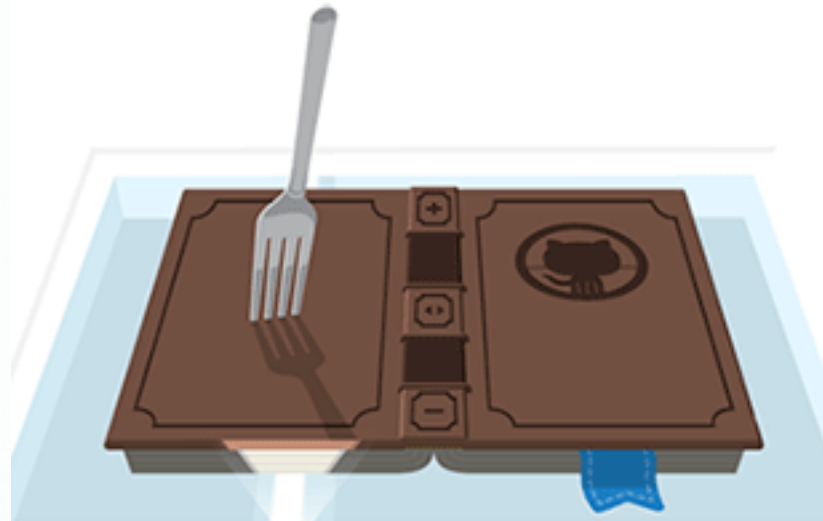


# Защо искаме да работим с GitHub?

- 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

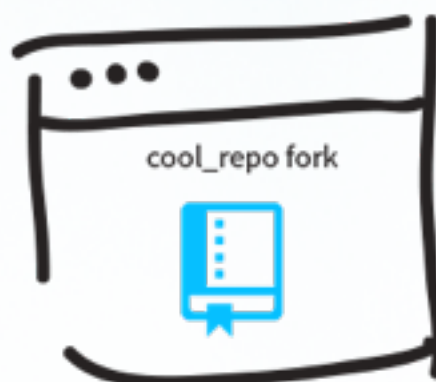
Someone else's repository.



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?



# Git задача

- 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: <https://desktop.github.com/>
- 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 задача

- 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
- Създайте нова папка с име - вашето малко име (без главни букви)
- Отворете Brackets и добавете новата папка като отделен проект (File -> Open folder)
- в нея, добавете още 2 нови папки: hw01, hw02
- във всяка копирайте съответното домашно (ако ви липсва домашно, може просто да го добавите по-късно, когато го направите)



# Git задача

- Върнете се в Github Desktop-а и кликнете на *Changes* в основната навигация
- 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 задача

- горе вдясно ще ви се появят бутони и *Publish* и *Pull-request*
- натиснете първо *Publish(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 брауъра и да видите вашето копие на swift-academy-homeworks репозитория
- уверете се че файловете ви са качени в проекта в брауъра
- изпратете ми линк към проекта ви



# HTML и HTML5

- HTML е маркър език (също като XML и XHTML)

*"Markup languages are designed for the processing, definition and presentation of text. The language specifies code for formatting, both the layout and style, within a text file. The code used to specify the formatting are called tags. HTML is a an example of a widely known and used markup language."*

source: <http://www.webopedia.com/>



# Основни компоненти на HTML документа

- html - root елементът (съдържа head и body елементите)
- head - невидимото съдържание на HTML документа
  - мета съдържание - ползва се за описание на страницата и предаване на информация към браузъра (пример `<meta charset="utf-8">`)
  - свързани файлове - стилове, фавиконка, фонтове
  - title - заглавие на документа
- body - видимото съдържание
  - форматиращи елементи (ul, a, p, div, section, .. etc)
  - вмъкнати изображения, мултимедия и др (img, iframe, object, video, audio)
  - текст

# Особености на HTML елементите

- състоят се от отварящ и затварящ таг  
`<body></body>`
- имат атрибути  
`<html lang="en">`  
``
- имат семантика (предназначение)  
`<p>` - параграф  
`<li>` - list item
- имат дървовидна структура, т.е. могат да се влагат един в друг
- има няколко специални елемента (като `img`, `hr` и мета таговете), които се състоят само от един таг - т.е. нямат затварящ таг



# ОСНОВНИ HTML елементи

- `<h1></h1>`, `<h2></h2>`, ... `<h5></h5>`
- `<div></div>`, `<span></span>`, `<p></p>`, `<button></button>`
- `<img>`, `<hr>`, `<input>`
- `<ul><li></li>.., <ol><li></li>..`
- `<main> </main>`, `<section></section>`, `<article></article>`
- `<nav></nav>`, `<header></header>`, `<footer></footer>`
- `<figure></figure>`, `<figcaption></figcaption>`

# Блокови и inline елементи

- В HTML форматирането има два основни типа елементи: блокови и inline
- Блоковите елементи са характерни с това, че заемат цялото пространство от страницата по хоризинтала. Те могат да се нареждат единствено *един под друг*
- Inline елементите могат да се разполагат хоризонтално *един след друг*
- Примери за блокови елементи: `div`, `p`, `h1`
- Примери за inline елементи: `img`, `span`



# Семантика

- Семантика, наричаме конкретното предназначение на всеки един html елемент
- Всеки елемент има някаква семантика, но можем да кажем че някои елементи имат богата, а други бедна семантика
- Например елемента **div** - означава деление (division) и не дава много информация какво е неговото съдържание. Съответно, той има бедна семантика. За сметка на това, елементи като **section**, **header** и **figure** имат богата семантика
- **Запомнете:** никой не ни задължава да пишем семантически издържано  
*И все пак добрите програмисти винаги предпочитат семантически богатите елементи и ги ползват по предназначение*

<header></header>

<nav></nav>

<section  
id="sidebar">  
</section>

Subscribe to RSS  
Subscribe via Email

<section id="content"></section>

Story Title  
Short description of story

Story Title  
Short description of story

Story Title  
Short description of story

<aside></aside>

Upcoming Events  
Event A  
Event B  
Event C

<footer></footer>



# HTML attributes

- **html** елементите могат да имат атрибути, които се задават по следния начин:  
`<име-на-елемент attribute1="value1" attribute2="value2" ... >`
- т.е. във формат: **ключ="стойност"**
- примери за често използвани атрибути: **id, src, href, alt, title ...**
- за атрибутите важат следните правила:
  - винаги трябва да ползваме кавички за стойността на атрибута
  - винаги трябва да изписваме ключа с малки букви
  - атрибутите могат да се слагат **единствено** в *отварящия таг* на елемента



# Глобални атрибути

- Повечето от html атрибутите са специфични за конкретни *html* елементи, като например **href** на линковете, **src** на картинките и т.н.
- Има обаче набор от атрибути, които могат да се ползват от **абсолютно** всички *html* елементи и те се наричат **глобални**
  - глобални атрибути са: **id, class, style, title** и т.н.
  - специфични атрибути:
    - **src**: използва се от **img, video, audio, iframe, script** и т.н.
    - **href**: използва се от **a** и **link**
    - и още много други: **type, alt, width, height, name, value** и т.н.

[https://developer.mozilla.org/en-US/docs/Web/HTML/Global\\_attributes](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes)  
[http://www.w3schools.com/tags/ref\\_standardattributes.asp](http://www.w3schools.com/tags/ref_standardattributes.asp)

# HTML стандарти

- HTML е стандартизиран от W3 Consortium (**W**orld **W**ide **W**eb Consortium) чрез множество спецификации (стандарти), описващи всяка версия на HTML
- целта на стандарта е
  1. да информира: Представява документация, която уеб разработчиците използват при изготвянето на уеб приложения
  2. да контролира: Задължава уеб браузърите да поддържат конкретна версия на HTML (конкретен стандарт)
- <https://www.w3.org/standards/webdesign/htmlcss>



# HTML5

- Това е най-новата версия на HTML, която влезе в сила преди около година и половина (края на 2014г.)
- Тя въвежда много подобрения върху HTML4. Като:
  - нова семантика (нови елементи и предназначение)
  - нови формати за мета съдържание
  - вградена валидация на уеб форми
- <https://www.w3.org/TR/2014/REC-html5-20141028/>
- **Важно:** валидните HTML5 документи (т.е. html страници) започват с декларацията: `<!DOCTYPE html>`

въпроси?..



# Основи на SEO

- SEO означава Search Engine Optimisation
- Не всяка страница която е публикувана в интернет може да бъде открита чрез метатърсачките
- За да излиза като резултат при търсенето, една страница първо трябва да бъде "индексирана" от съответната търсачка
- Позицията, на която страницата се появява в резултатите от търсенето (SERP) се нарича *ranking*
- Целта на SEO е да подобри ранкинга на дадена страница







За това как се индексират  
web сайтове, ще говорим в друга лекция,  
но ако ви е наистина много интересно,  
можете да разучите **Google Search Console**:

<https://www.google.com/webmasters/tools>

# Мета съдържание

- Част от meta съдържанието, което можем да добавим към една страница е ориентирано към SEO-то
- Това са няколко meta таг-а, които задават описание, автор и ключови думи за съответната страница
- Тази meta информация се взима предвид при формирането на SERP от meta търсачките
- Забележка: от известно време насам Google и други големи meta търсачки пренебрегват keywords meta тагът!



# SEO мета тагове

- Автор на документа (страницата):  
`<meta name="author" value="Evgenia Manolova">`
- Описание на документа:  
`<meta name="description" value="A complete front-end development course for beginners">`
- Ключови думи за този документ:  
`<meta name="keywords" value="course,front-end,development,html,html5,css,javascript,complete">`

въпроси?..



# Примери

<http://zenlabs.pro/courses/sa-fe-04/lessons/lesson3/examples.zip>

# Домашно

<http://zenlabs.pro/courses/sa-fe-04/lessons/lesson3/hw-task.html>