

Технологии программирования

Занятие 2

Системы контроля версий. git

Поговорим о сдаче ДЗ

На почту (будет указана почта для каждого задания) необходимо прислать ссылку на репозиторий

Архивы приниматься не будут!

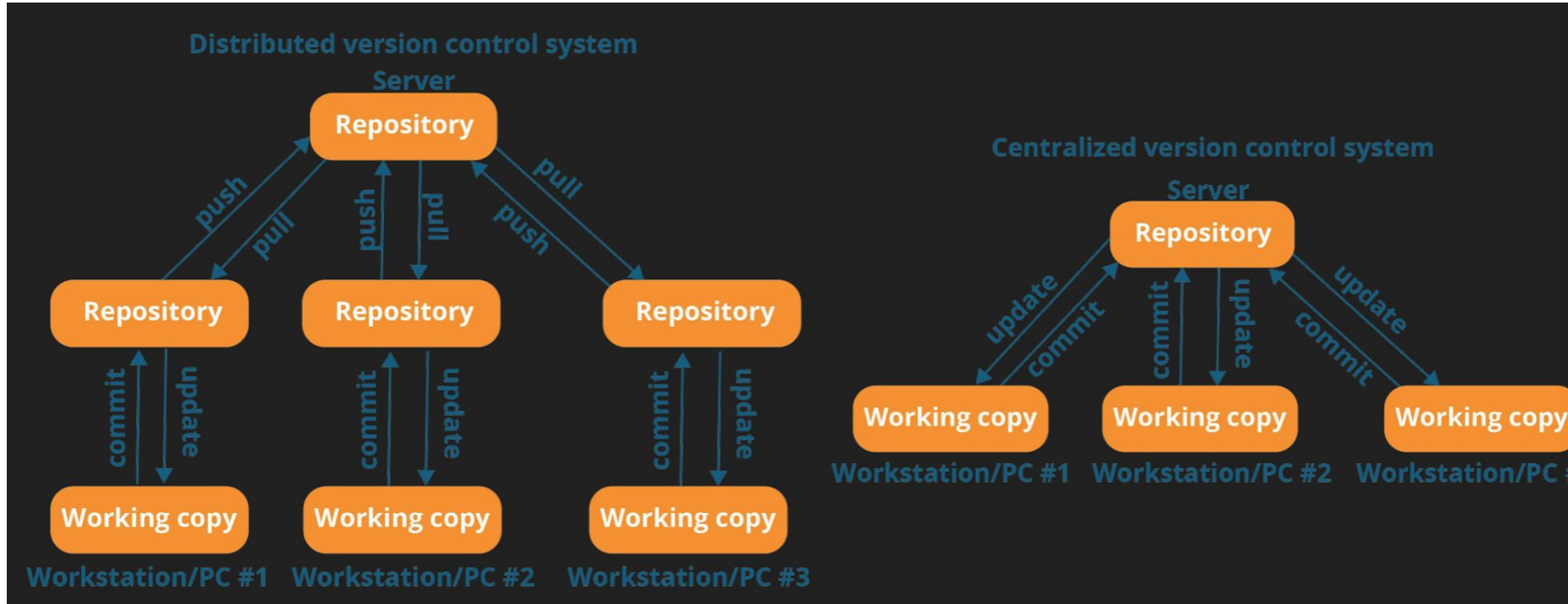
А теперь - почему ДЗ не в архивах??



Зачем использовать систему контроля версии?

- Сотрудничество!
- Возможность к откатам
- Отслеживание прогресса выполнения

Как выглядит?



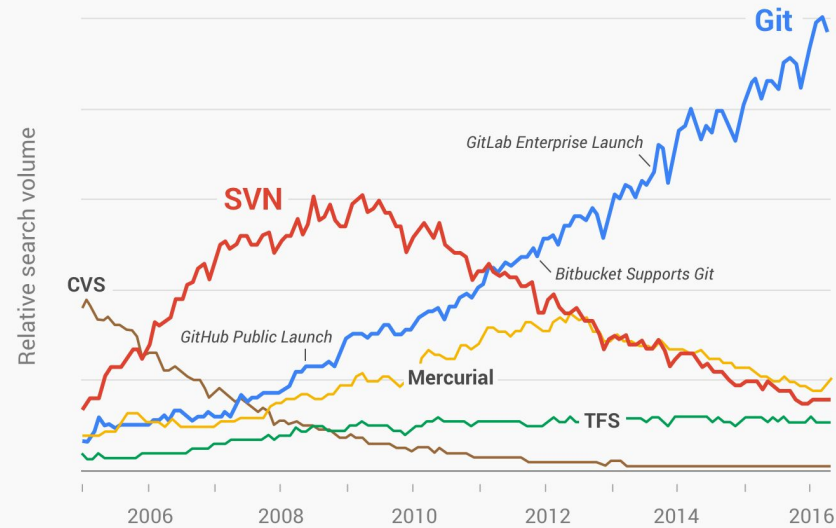
Системы контроля версий

- SVN
- Mercurial
- Git

Trends

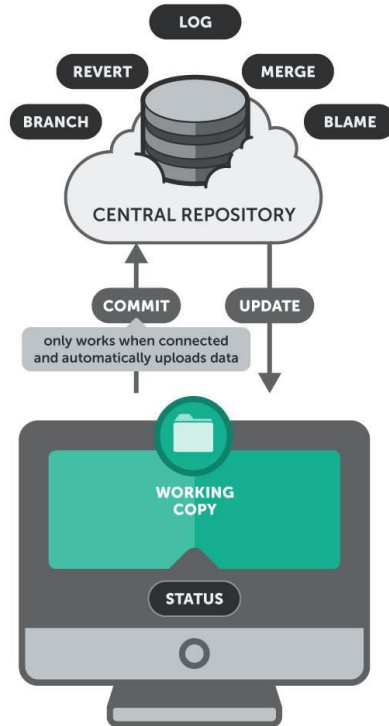
How Did SVN Manage to Lose?

Version control interest over time

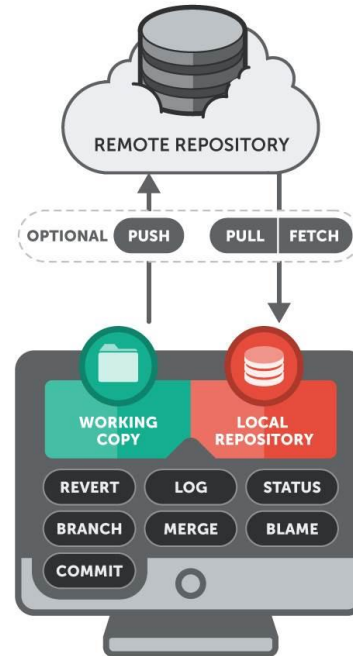


Git vs SVN


SUBVERSION




GIT




Хранилища Git-репозиториев: Github




[/](#) [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)





 **akhtyamovpavel** ▾


Repositories


New


 [VicManDevelop... /nnDatasetValidator](#)


 [VicManDevelop... /nnStyleTransferIos](#)

 [AndrewErmakov/PythonTrainingBasics](#)

 [akhtyamovpavel/ParallelComputationEx...](#)

 [akhtyamovpavel/TestingRepo](#)


 [akhtyamovpavel/BuildExamples-TP](#)


 [gostkin/parallel-repo](#)


Show more


Your teams


You don't belong to any teams yet!



**clefebvre** pushed to **linuxmint/Cinnamon** 2 days ago


1 commit to [master](#)
 [13ee4b8](#) [ImgBot] Optimize images (#8379)

**clefebvre** pushed to **linuxmint/Cinnamon** 4 days ago

1 commit to [master](#)
 [0382602](#) gwl: Fix the thumbnail menu not closing on leave after closing a wind...

**clefebvre** pushed to **linuxmint/Cinnamon** 5 days ago


4 commits to [master](#)
 [a03978f](#) Use icon-changed signal
 [637c078](#) gwl: Fix icon-changed handler
2 more commits »


**clefebvre** pushed to **linuxmint/Cinnamon** 5 days ago


1 commit to [master](#)

Welcome to the new dashboard. Get closer to the stuff you care about most. ✕

Discover repositories

h2oai/pystacknet
 Jupyter Notebook ★ 88

catboost/catboost
A fast, scalable, high performance Gradient Boosting on Decision Trees library, used for ranking, classification, regression and other machine learning tasks for Python, R, Java, C++. Supports comp...
 C++ ★ 3.7k

visipedia/tfrecords
Functions for creating tfrecords for TensorFlow models.
 Python ★ 76
Go to Explore →

Хранилища Git-репозиториев: Gitlab



Projects ▾

Groups ▾

Activity

Milestones

Snippets



 ▾

Search or jump to... 

 7



 2

 ▾

 ▾

M Scripts making the work of mentors more simple and quick. ☆ 0 👤 0 📄 0 📅 Updated 3 weeks ago ☆ Star

F Piscenco Margarita / **formalki**  Guest  Star

★ 0 🗑️ 0 📄 0 📁 0 Updated 1 month ago

 Yura / [regex_hw](#)  Maintainer  Star

 Pavel Akhtyamov / **BuildExamples-TP**  Maintainer

Examples of bash, make, cmake building

★ 0 🍴 0 📄 0 📦 0  Updated 3 weeks ago

 Star

 Pavel Akhtyamov / **GitlabTest**  Maintainer  Star

 **VeLKerr / pardistrib**  Developer
Parallel & Distributed computing materials
★ 6 🔒 0 📄 0 Updated 3 weeks ago ★ Unstar

Pavel Akhtyamov / [AmazonLasagneInstance](#)  Maintainer

- Работа с репозиториями в Git
- создание репозитория
- понятие коммита
- создаем первый коммит
- присоединяем удаленный репозиторий
- заливаем первые изменения
- создаем новую ветку
- получаем изменения удаленно из новой ветки
- сливаем ветки руками
- узнаем про gitignore

Создаем репозиторий

- Создаем папку FirstGitRepo (mkdir FirstGitRepo)
- Переходим в нее (cd FirstGitRepo)
- Создаем git-репозиторий (git init)

```
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial [18:04:28]
→ mkdir FirstGitRepo
└─┘
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial [18:04:39]
→ cd FirstGitRepo
└─┘
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial/FirstGitRepo [18:04:42]
→ ls
└─┘
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial/FirstGitRepo [18:04:42]
→ git init
Initialized empty Git repository in /Users/akhtyamovpavel/Teaching/TechProg/2020/02-git-tutorial/FirstGitRepo/.git/
└─┘
```

Понятие коммита

Коммит - это фиксированное изменение. Представляется в виде вершины.

diff между коммитами - изменение dx , применяемое к коммиту c_1 , приводит к коммиту c_2 (обычно - последовательность добавления, удаления строк)

Создадим первый коммит

1. Запишем “# Git Tutorial” в readme.md
2. Посмотрим изменения (git status)

```
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial/FirstGitRepo on git:master ● [18:10:02]
→ echo "# Git Tutorial" > readme.md
>
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial/FirstGitRepo on git:master ✖ [18:10:31]
→ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    readme.md

nothing added to commit but untracked files present (use "git add" to track)
>
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutorial/FirstGitRepo on git:master ✖ [18:11:10]
→ █
```

Видим красным readme.md

Добавим файл в коммит

1. Добавим файл в изменение (tracked file): `git add readme.md`
2. Видим, что файл стал tracked (`git status`)

```
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching
```

```
→ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   readme.md
```



```
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching
```

```
→
```



Создаем коммит

1. Для того, чтобы создать коммит, необходимо зафиксировать tracked files. Это делается командой **git commit** При этом высветится страшное окно редактора... Поэтому выполняем **git commit -m "Название коммита"**

```
→ git commit -m "Added readme"  
[master (root-commit) e0cd0b2] Added readme  
1 file changed, 1 insertion(+)  
create mode 100644 readme.md
```

2. Чтобы убедиться, что коммит - это нечто физическое, посмотрим git log (список коммитов)

Если коммит не прошел...

Необходимо представиться git-у.

```
git config --global user.email "your email"
```

```
git config --global user.name "your name"
```

Советую сделать --global на --local, поскольку на каждый git-репозиторий могут быть свои почты/

Убеждаемся, что коммит создан

Видим hash commit-а (уникальный идентификатор - по нему можно откатиться)

Также, автор коммита, дата коммита, HEAD -> master (последний коммит указывает на master)

```
commit e0cd0b24d76aea9f62db5594f150508d43aebff8 (HEAD -> master)
Author: Pavel Akhtyamov <akhtyamovpavel@gmail.com>
Date: Thu Feb 6 18:45:31 2020 +0300
```

Added readme

(END)

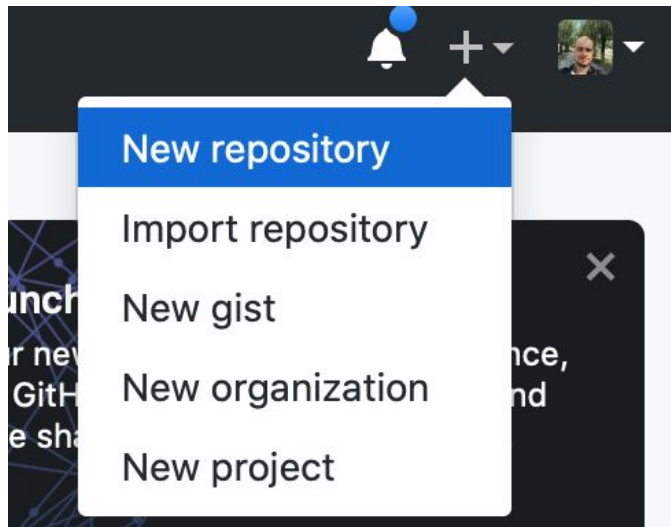
Заливаем изменения в удаленный репозиторий

У нас - локальный репозиторий, при этом у каждого локального репозитория может быть несколько удаленных репозиториев.

По истории, удаленный репозиторий называется origin.

Чтобы подключить удаленный репозиторий, его нужно создать.

Создадим репозиторий на github.com



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner



akhtyamovpavel ▾

Repository name *

FirstGitRepo ✓

Great repository names are short and memorable. Need inspiration? How about [scaling-octo-couscous?](#)

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

Появится такое окно...

...or push an existing repository from the command line

```
git remote add origin https://github.com/akhtyamovpavel/FirstGitRepo.git  
git push -u origin master
```



`git remote add <name of remote repo> <link of remote repo>` - добавить удаленный репозиторий

`git push <name of remote repo> <branch name>` - залить коммиты в удаленный репозиторий (-u привязать удаленный репозиторий по умолчанию к ветке)

Убеждаемся, что все получено

```
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutori
→ git remote add origin https://github.com/akhtyamovpavel/FirstGitRepo.git
>
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutori
→ git push -u origin master
Username for 'https://github.com': akhtyamovpavel
Password for 'https://akhtyamovpavel@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/akhtyamovpavel/FirstGitRepo.git/'
>
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teaching/TechProg/2020/02-git-tutori
→ git push -u origin master
Username for 'https://github.com': akhtyamovpavel
Password for 'https://akhtyamovpavel@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 233 bytes | 233.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/akhtyamovpavel/FirstGitRepo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Убеждаемся, что все получено в Github

akhtyamovpavel / FirstGitRepo Private

Unwatch 1 Star 0 Fork 0

<> Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

Edit

Manage topics

1 commit

1 branch

0 packages

0 releases

1 contributor

Branch: master


New pull request

Create new file

Upload files

Find file

Clone or download


 akhtyamovpavel Added readme Latest commit e0cd0b2 27 minutes ago

readme.md

Added readme

27 minutes ago

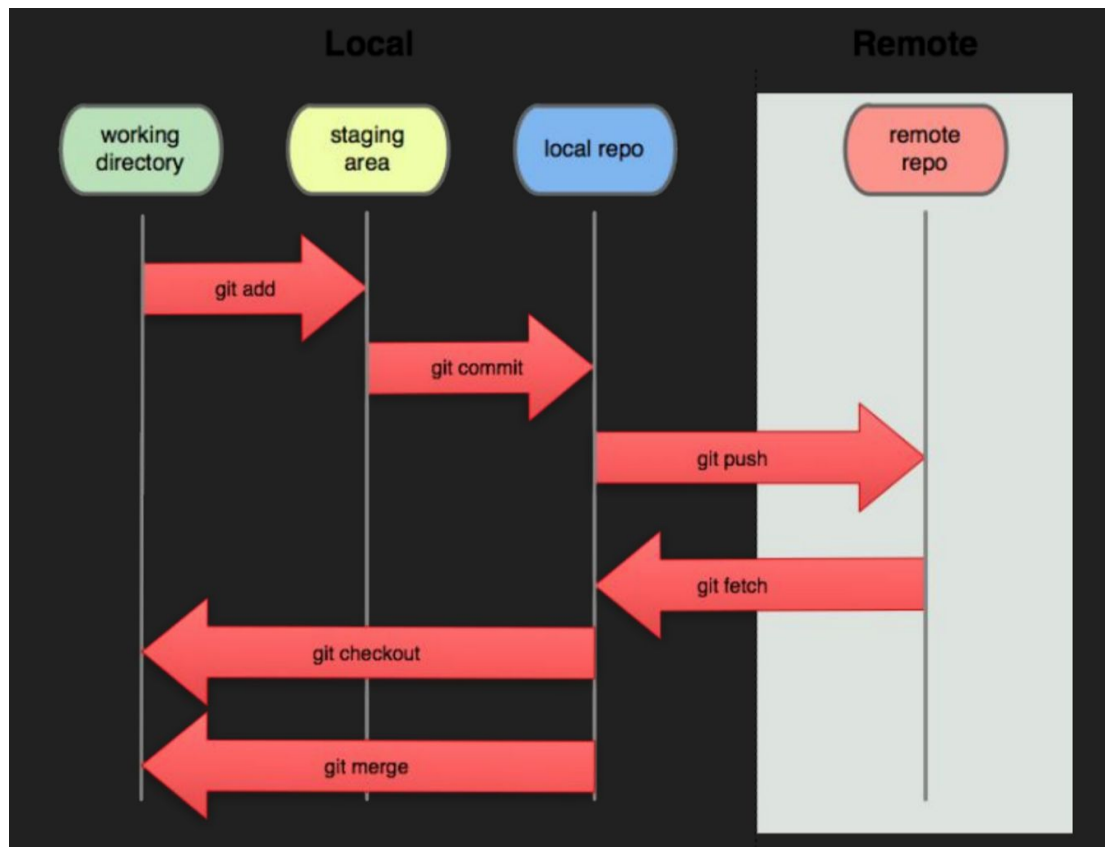
readme.md



Git Tutorial

Вопросы???

Структура Git



Работа с ветками

Ветка - сущность, вдоль которой проходят коммиты. Изначально есть ровно одна ветка - **master** (см. выше)

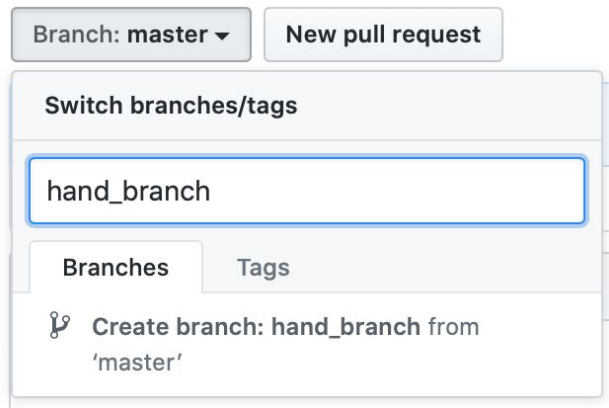
Как можно создавать новые ветки?

Работа с ветками

Ветка - сущность, вдоль которой проходят коммиты. Изначально есть ровно одна ветка - **master** (см. выше)

Как можно создавать новые ветки?

1. Создать из удаленного репозитория
 - а. Интерфейс переключается на новую ветку



Работа с ветками

Ветка - сущность, вдоль которой проходят коммиты. Изначально есть ровно одна ветка - **master** (см. выше)

Как можно создавать новые ветки?

1. Создать из удаленного репозитория

- Интерфейс переключается на новую ветку
- В консоли необходимо выключить команду `git fetch origin` (выкачать все ветки из удаленного репозитория origin)

```
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teachin
→ git fetch origin
From https://github.com/akhtyamovpavel/FirstGitRepo
* [new branch]      hand_branch -> origin/hand_branch
>
# akhtyamovpavel at MacBook-Pro-Pavel-2.local in ~/Teachin
→ █
```

Работа с ветками

Ветка - сущность, вдоль которой проходят коммиты. Изначально есть ровно одна ветка - **master** (см. выше)

Как можно создавать новые ветки?

1. Создать из удаленного репозитория
 - a. Интерфейс переключается на новую ветку
 - b. В консоли необходимо выключить команду `git fetch origin` (выкачать все ветки из удаленного репозитория `origin`)
 - c. Переключиться на новую ветку: `git checkout hand_branch`

Работа с ветками

Ветка - сущность, вдоль которой проходят коммиты. Изначально есть ровно одна ветка - **master** (см. выше)

Как можно создавать новые ветки?

2. Создать из консольного интерфейса:

- `git checkout -b console_hand`
- `git branch` - узнать текущую ветку
- Залить ветку **`git push origin console_hand`**

```
* console_hand  
  hand_branch  
  master  
(END)
```

```
→ git checkout -b console_hand  
Switched to a new branch 'console_hand'
```

```
# akhtyamovpavel at MacBook-Pro-Pavel-2.1  
→ git branch
```

Слияние веток

Ветки можно сливать двумя способами:

- Локально - `git merge`
- Из удаленного репозитория: `git pull = git fetch + git merge`

Слияние веток

Ветки можно сливать двумя способами:

- Локально - `git merge`
- Из удаленного репозитория: `git pull = git fetch + git merge`
 - Создаем новый файл

Branch: hand_branch ▾

New pull request



Create new file


Upload files

Find file


Clone or download ▾

This branch is even with master.

 Pull request  Compare


 **akhtyamovpavel** Added readme


Latest commit e0cd0b2 2 hours ago

 [readme.md](#)

Added readme

2 hours ago

 [readme.md](#)



Слияние веток

FirstGitRepo / new.txt Cancel

<> Edit new file Preview Spaces 2 No wrap

```
1 hello world
```

Ветки можно сливать двумя способами:

- Локально - git merge
- Из удаленного репозитория: git pull = git fetch + git merge
 - Создаем новый файл
 - Коммитим его



Commit new file

Create new.txt

Add an optional extended description...

akhtyamovpavel@gmail.com

Choose which email address to associate with this commit

☒ Commit directly to the hand_branch branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file

Cancel

ALERT!

Этот пример был показан как только пример!

Коммиты из веб-интерфейса будут обнулять балл за задание без права пересылки задания!

Все коммиты должны быть из консольного интерфейса!

Слияние веток

Ветки можно сливать двумя способами:

- Локально - `git merge`
- Из удаленного репозитория: `git pull = git fetch + git merge`

- Создаем новый файл
- Коммитим его
- `git pull origin hand_branch`

```
# akhtyamovpavel at MacBook-Pro-  
→ git pull origin hand_branch |
```

Слияние веток

Ветки можно сливать двумя способами:

- Локально - git merge
- Из удаленного репозитория: git pull = git fetch + git merge

- Создаем новый файл
- Коммитим его
- git pull origin hand_branch

```
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/akhtyamovpavel/FirstGitRepo
* branch                hand_branch -> FETCH_HEAD
   e0cd0b2..728bde9      hand_branch -> origin/hand_branch
Updating e0cd0b2..728bde9
Fast-forward
 new.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 new.txt
```

Слияние веток

Ветки можно сливать двумя способами:

- Локально - `git merge`
- Из удаленного репозитория: `git pull = git fetch + git merge`
 - Создаем новый файл
 - Коммитим его
 - `git pull origin hand_branch`

.gitignore

В репозитории можно создать отдельный файл, в котором говорим, какие файлы не должны отслеживаться через git.

Поддерживает работу с шаблонами (см. прошлый семинар)

Пример .gitignore файла

*.рус

.ipynb_checkpoints

other_repo

stands/*

!stands/example_stands/

datasets/

runs/

checkpoints/

Какие файлы отслеживаются, а какие - нет?

Last - but not least

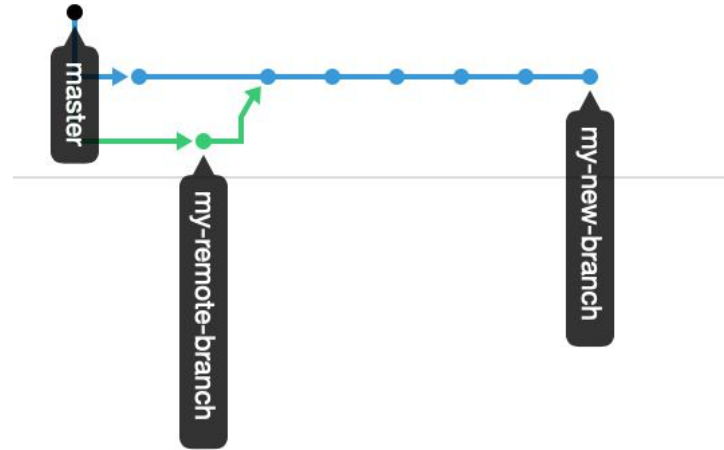
Не забывайте добавить .gitignore файл в репозиторий!

Как вывести граф изменений в консоли?

```
git log --oneline --decorate --graph --all
```

git advanced* (rebase and squash commits)

1. Сделайте в ветке my-new-branch следующие пять файлов: a.py, b.py, c.py, d.py, e.py - лучше через веб интерфейс
2. Скачайте изменения локально



git advanced

3. Выполните команду git rebase -i HEAD~5

```
1 | pick d679f0a Create a.py
2 | pick 5cb02a5 Create b.py
3 | pick e9ae5bc Create c.py
4 | pick a77ba23 Create d.py
5 | pick ef49fca Create e.py
6
7 # Rebase 41ca702..ef49fca onto 41ca702 (5 commands)
8 #
9 # Commands:
10 # p, pick = use commit
11 # r, reword = use commit, but edit the commit message
12 # e, edit = use commit, but stop for amending
13 # s, squash = use commit, but meld into previous commit
14 # f, fixup = like "squash", but discard this commit's log message
15 # x, exec = run command (the rest of the line) using shell
16 # d, drop = remove commit
17 #
18 # These lines can be re-ordered; they are executed from top to bottom.
19 #
20 # If you remove a line here THAT COMMIT WILL BE LOST.
21 #
22 # However, if you remove everything, the rebase will be aborted.
23 #
24 # Note that empty commits are commented out
```

Изменяем названия коммитов

4. Изменим названия коммитов “Create b.py и d.py” (поменяйте на edit)

```
# akhteyanov@pavlov:~/facebook/fbo/faces/2.0/cats/1
→ git rebase -i HEAD~5
Stopped at 5cb02a5... Create b.py
You can amend the commit now, with

    git commit --amend

Once you are satisfied with your changes, run

    git rebase --continue
```

Выполняем git commit --amend

Adding feature b

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Sep 9 19:40:50 2018 +0300
#
# interactive rebase in progress; onto 41ca702
# Last commands done (2 commands done):
#   pick d679f0a Create a.py
#   edit 5cb02a5 Create b.py
# Next commands to do (3 remaining commands):
#   pick e9ae5bc Create c.py
#   edit a77ba23 Create d.py
# You are currently editing a commit while rebasing branch 'my-new-branch' on '41ca702'.
#
# Changes to be committed:
#   new file:   src/b.py
#
```

Продолжаем

6. Выполняем `git rebase --continue`.
7. Повторяем действие со вторым коммитом.
8. Выполнится ли `git push origin my-new-branch`?

А история изменена!

```
* 19f84bb (HEAD -> my-new-branch) Create e.py
* d6dcff2 Adding feature d
* 4a3590d Create c.py
* 5f1f8e9 Adding feature b
| * ef49fca (origin/my-new-branch) Create e.py
| * a77ba23 Create d.py
| * e9ae5bc Create c.py
| * 5cb02a5 Create b.py
|/
* d679f0a Create a.py
* 41ca702 Merge branch 'my-remote-branch' into my-new-branch
|\
| * efc87b4 (origin/my-remote-branch, my-remote-branch) Create README.md
* | 02e83cf Hello world added
|/
* 4e6b308 (origin/master, local/master, master) My first commit
```

Не заливайте ветки, если делаете rebase!

Сожмем все в один коммит (squash)

1. В git rebase -i HEAD~5 сделаем такую вещь:

```
pick d679f0a Create a.py
squash 5f1f8e9 Adding feature b
squash 4a3590d Create c.py
squash d6dcff2 Adding feature d
squash 19f84bb Create e.py
#
# Rebase 41ca702..19f84bb onto 41ca702 (5 commands)
#
```


Получим такую портянку

```
# This is a combination of 5 commits.
# This is the 1st commit message:

Create a.py
# This is the commit message #2:
█
Adding feature b

# This is the commit message #3:

Create c.py
# This is the commit message #4:

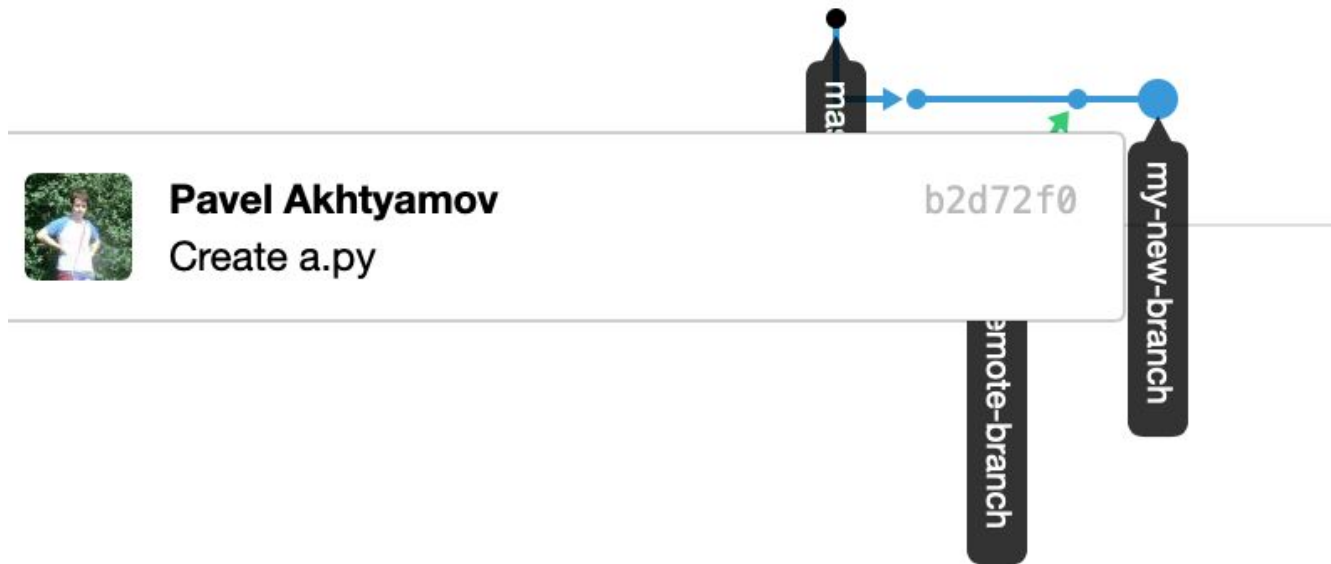
Adding feature d

# This is the commit message #5:

Create e.py

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Sep 9 19:40:39 2018 +0300
#
# interactive rebase in progress; onto 41ca702
# Last commands done (5 commands done):
#   squash d6dcff2 Adding feature d
#   squash 19f84bb Create e.py
# No commands remaining.
# You are currently rebasing branch 'my-new-branch' on '41ca702'.
#
# Changes to be committed:
#   new file:   src/a.py
#   new file:   src/b.py
#   new file:   src/c.py
#   new file:   src/d.py
#   new file:   src/e.py
```

Что получили в итоге!



Что мы узнали сегодня?

- Как работать с git:
 - создавать репозитории
 - публиковать изменения
 - создавать ветки с разными версиями кода
 - сливать ветки
 - не добавлять лишние файлы
 - что делать, если что-то пошло не так