

Git intro

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

Изучить базовые команды, настроить среду

ПЛАН ЗАНЯТИЙ

При необходимости генерация ssh
ключа

Изучение разницы между видами
репозиторий: локальный vs
удаленный

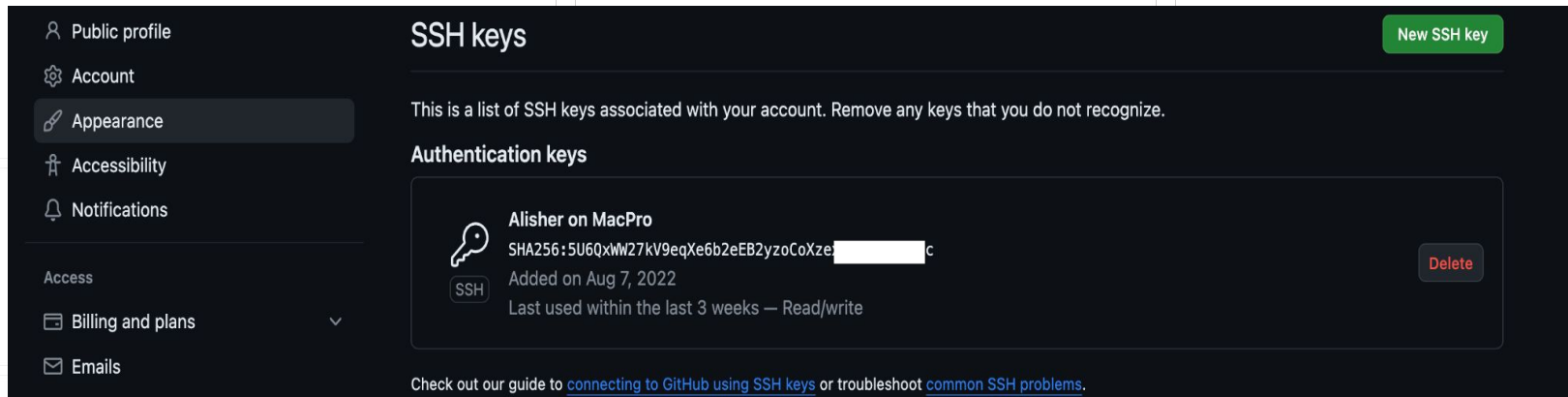
Повторение/Изучение базовых
команд git и основных
определений

Разница между git и github

clone

Настройка среды

- Сгенерируем ключ при необходимости и добавим его в гитхаб



- Команды для этого можно найти по ссылке:
<https://docs.github.com/ru/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent?platform=mac>

Настройка среды

- **git config --global user.name "[firstname lastname]"**
устанавливает имя пользователя, которое будет указано в коммите (указывайте без квадратных скобок)
- **git config --global user.email "[valid-email]"**
указывает имейл от гитхаб, который будет использоваться в истории коммитов
- **git config --global color.ui**
опционально можно выбрать цветовую схему для более комфортной работы в терминале

Локальный vs удаленный репозитории

- локальный (расположен непосредственно в памяти компьютера разработчика, в нем происходит разработка и фиксация изменений, после чего можно отправить на удаленный репозиторий)
- удаленный (находится на сервере, может быть приватным – доступным ограниченному числу лиц, и публичным – open source)

Про папки и репозитории

Если папка — это то, к чему мы все привыкли как пользователи компьютеров, то репозиторий — это что-то новое, что нужно создать, инициализировать. Сам по себе репозиторий без наших указаний не появляется. Репозиторий в наших задачах — это папка, над которой были произведены некоторые действия, и Git в ней начинает выполнять свои задачи, например:

- **отслеживать изменения файлов;**
- **хранить информацию о ветках.**

Важно! Репозиторий не возникает сам по себе, его нужно создать

Как понять, в репозитории мы находимся или в папке?

Самый простой способ это сделать — набрать в терминале команду «git status». Если в ответ вы увидите ошибку «fatal: not a git repository (or any of the parent directories): .git», значит, в терминале вы вызываете команду не из репозитория, а из обычной папки. Если вы увидели что-то другое, то вы находитесь в репозитории или внутри одной из папок, которая находится в нем.

Важно! Репозиторий отслеживает изменения во всех вложенных в него папках.

Если вы сделаете репозиторием корневую папку на диске C (не делайте этого!), то весь ваш диск станет репозиторием и Git будет пытаться отслеживать все изменения на этом диске. Создаем репозитории очень аккуратно.

Как можно создать репозиторий?

Чаще всего на начальных этапах рассматривают два способа создания репозитория:

- Если мы находимся в папке (!) и хотим сделать из нее репозиторий, то вызываем команду «**git init**», и эта папка становится репозиторием.
- Если мы хотим клонировать репозиторий из GitHub на свой ПК, то мы пользуемся командой «**git clone**».

(При этом обратите внимание: не нужно пользоваться командой «git init», команда clone не только скачивает файлы из интернета, но и инициализирует репозиторий в скачанной папке. На самом деле она делает сильно больше, но нам важно, что в скачанной папке у нас уже будет репозиторий и никак дополнительно инициализировать его не надо).

Внимательно следим за тем, из какой папки вы вызываете команды

Терминал всегда показывает, в какой папке вы сейчас находитесь, но первое время студенты чаще смотрят на то, какая папка открыта в визуальном интерфейсе редактора (например, VSCode), а не на то, что написано в терминале. Обращайте, пожалуйста, внимание на название папки, которая указана в приглашении к вводу команд терминала. До тех пор, пока вы не привыкнете к работе с терминалом, внимательно следите за тем, что вы создаете репозитории только во вновь созданных для урока папках. Не нужно создавать репозитории из рабочего стола или других больших папок.

Внимательно следим за тем, из какой папки вы вызываете команды

В данном примере терминал открыт в папке `git_example_04`

```
○ aliserkhamidov@MacBook-Pro-2 git_example_04 % █
```

Папка vs локальный репозиторий

Когда вы создаете локальный репозиторий, у вас в папке появляется новая **скрытая папка с названием «.git»**. Это специальная папка, в которой хранится все, что необходимо для работы системы контроля версий. **Если вы удалите эту папку, то потеряете всю историю, которую Git успел сохранить, но при этом превратите ваш репозиторий обратно в папку.**

Итак, чтобы из репозитория снова сделать папку, достаточно всего лишь удалить скрытую папку «.git». При этом вы потеряете историю, которую собрал Git (все коммиты, ветки и т. п.), но файлы в самой папке останутся в том же виде, в котором они были в момент удаления папки «.git».

(Если вы работаете на Windows, включите отображение скрытых файлов и папок, так как папка .git скрытая. Это можно сделать в верхнем меню. На маке достаточно нажать **'cmd + shift + .'**

Как переключиться на нужный коммит?

Git: Для переключения на нужный коммит используется действие checkout.

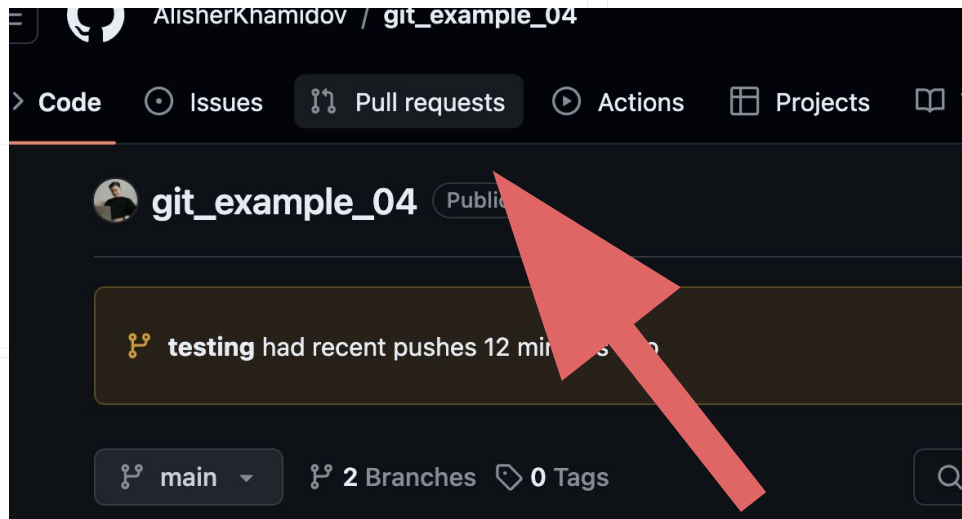
После переключения, все файлы в проекте станут такими, какими они были в данном коммите.

```
git checkout <commit>
```

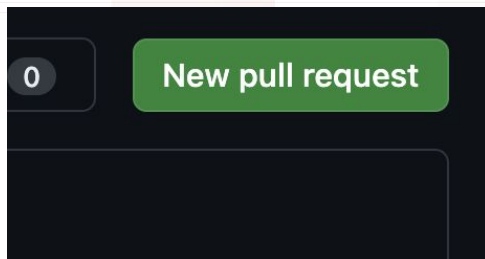
Где **<commit>** - это хеш (обозначение, имя) коммита, причем можно указывать не весь хеш, а несколько начальных символов хеша.

Pull request

Заходите в репозиторий на
github во вкладу **Pull requests**

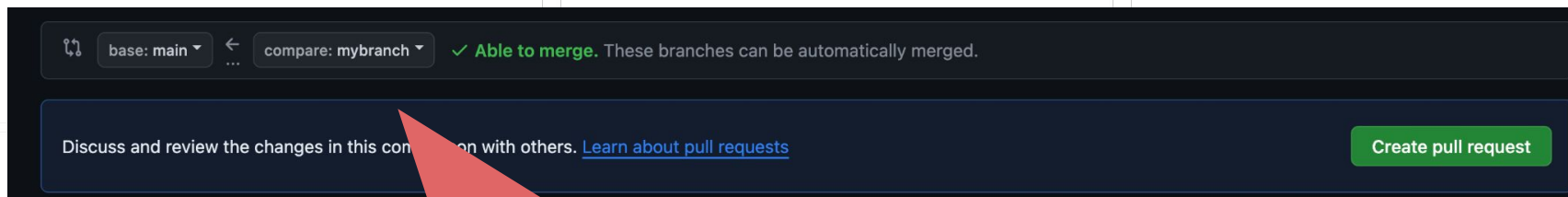


Нажимаете на кнопку New
pull request



Pull request

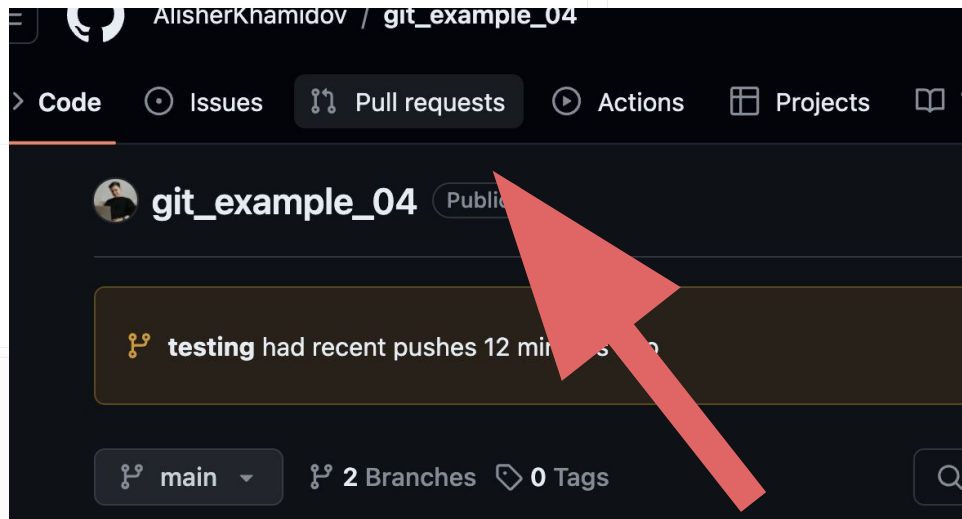
Выберите из какой ветки в какую вы собираетесь перенести изменения



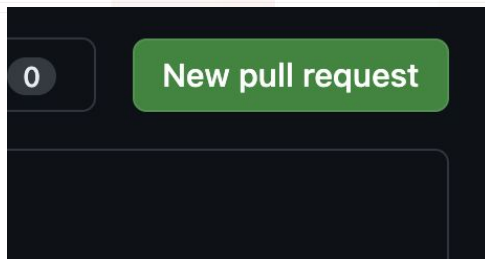
В данном примере мы переносим изменения из ветки **mybranch** в ветку **main**

Pull request

Заходите в репозиторий на
github во вкладу **Pull requests**



Нажимаете на кнопку New
pull request





Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH