

Практическая работа № __

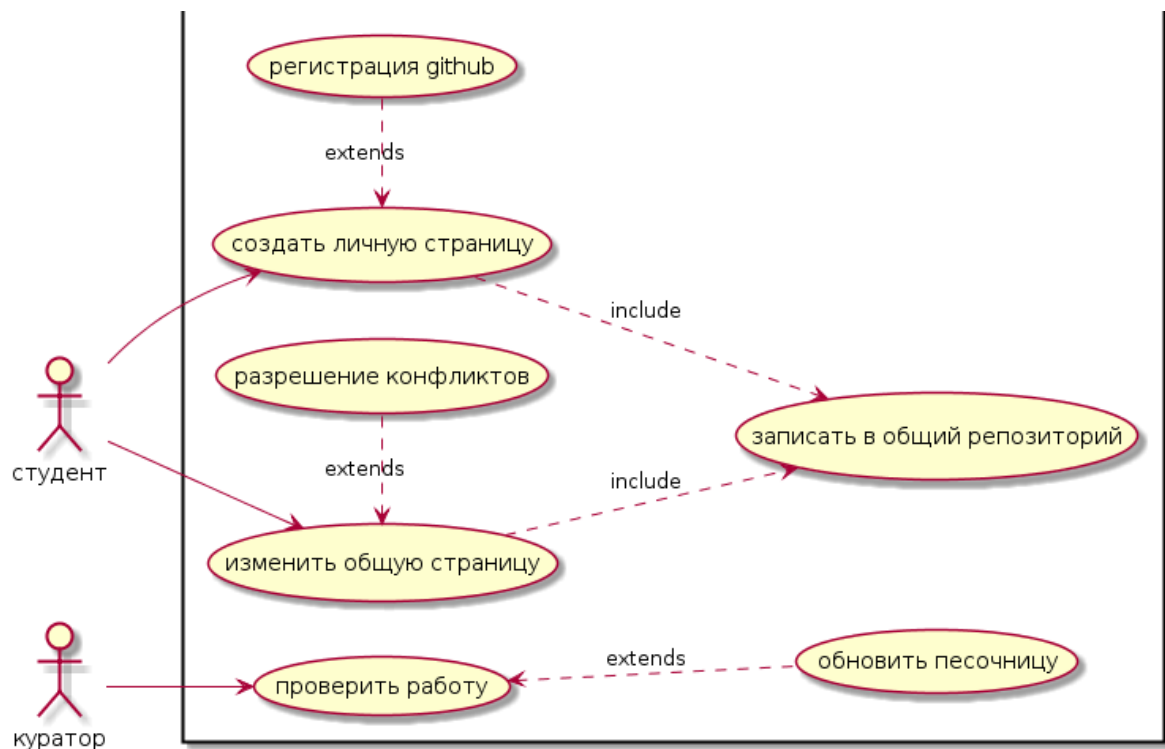
Знакомство с github, общими репозиториями и проектами

Цель работы.

создание личной страницы в режиме hotfix через интерфейс github

отладка личной страницы в локальной директории с сохранением (commit) в общий репозиторий

Ход работы:



Git — это система управления версиями, которая пришлась по душе практически всем — от разработчиков до дизайнеров. GitHub можно считать соцсетью для хранения кода. Здесь вы можете попрактиковаться в разработке и придумать что-то свое, найти множество open-source проектов, передовых технологий, различных функций и дизайнов. На GitHub вы учитесь и участвуете в других проектах, храните код для работы или учебы, берете код других проектов и вникаете во все детали. А еще вы можете создавать сайты бесплатно напрямую из репозитория!

Не лишним будет разобраться с терминалом, поскольку терминальные команды действительно упрощают жизнь.

Если в работе вы видите команду с угловыми скобками: `< >`, то смело удаляйте эти скобки и меняйте их содержимое на нужный вам текст.

Пример: `git add <имя_файла>`. Здесь вы можете написать нечто подобное: `git add index.html`. Это означает, что вы хотите добавить в репозиторий файл под названием `index.html`/

Для начала необходимо запомнить следующие терминальные команды:

`git clone`

`git status`

git add

git commit -m “ “

git push

Затем к ним добавим еще вот эти:

git init

git branch

git merge

git checkout

Эти команды вам пригодятся в случае, если вы будете работать с другими людьми или захотите внести какие-то изменения в проект и протестировать их до создания коммита.

Не лишней будет и вот такая команда:

git help

Шаг 1: Регистрация и установка

Зайдите на GitHub и создайте свой аккаунт. Можете скачать установщик или установить файлы через менеджер пакетов.

Теперь перейдите в терминал, и начнем работу. Если хотите задать одно имя пользователя для всех репозиториях на компьютере, то напишите:

```
git config — global user.name “<ваше_имя>”
```

замените <ваше_имя> на свое имя в кавычках. Можете написать все, что угодно. Если хотите задать имя только для одного репозитория, то удалите из команды слово global.

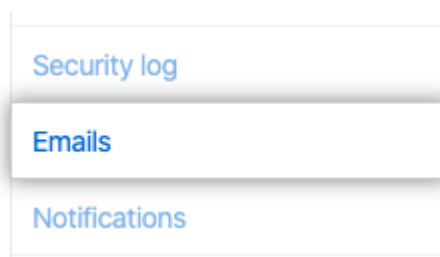
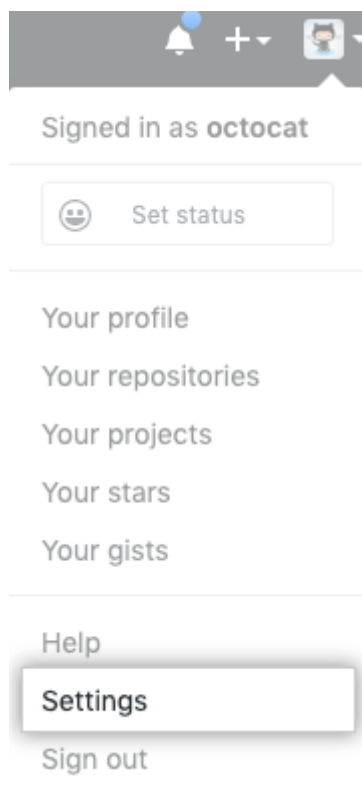
Теперь напишите свой адрес электронной почты. Проследите, чтобы он совпадал с адресом, указанным при регистрации на GitHub.

```
git config — global user.email “<адрес_почты@адрес>”
```

При желании можете скрыть свой электронный адрес.

1. В правом верхнем углу любой страницы нажмите на фотографию своего профиля, затем нажмите « **Настройки** » .

2. На левой боковой панели нажмите **Электронная почта** .



3. Чтобы сохранить конфиденциальность вашего адреса электронной почты при выполнении веб-операций Git, нажмите « **Сохранить мой адрес электронной почты закрытым** » .

Allow all verified emails



Save

☒ **Keep my email address private**

We will use `octocat@users.noreply.github.com` when performing web-based Git operations and sending email on your behalf. If you want command line Git operations to use your private email you must [set your email in Git](#).

☐ **Block command line pushes that expose my email**

If you push commits that use a private email as your author email we will block the push and warn you about exposing your private email.

4. Чтобы сохранить конфиденциальность вашего адреса электронной почты в коммитах, которые вы отправляете из командной строки, выберите **Блокировать запросы командной строки, которые будут отображать мою электронную почту**.

Allow all verified emails



Save

☒ **Keep my email address private**

We will use `octocat@users.noreply.github.com` when performing web-based Git operations and sending email on your behalf. If you want command line Git operations to use your private email you must [set your email in Git](#).

☒ **Block command line pushes that expose my email**

If you push commits that use a private email as your author email we will block the push and warn you about exposing your private email.

Теперь вы готовы к работе с Git на локальном компьютере.

Начнем с создания нового репозитория на сайте GitHub. Вы также можете выполнить `git init` и создать новый репозиторий из директории проекта.

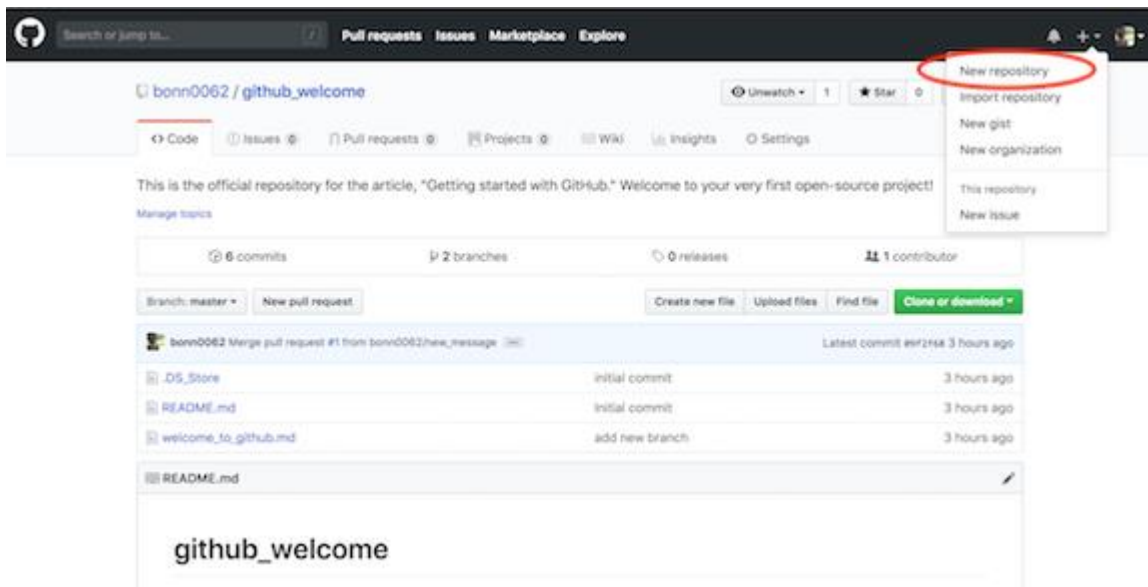
Репозиторий состоит из трех «деревьев». Первое «дерево» — это рабочая директория, в которой хранятся актуальные файлы. Второе — это `index` или область подготовленных файлов. А еще есть `head` — указатель на ваш последний коммит.

Репозиторий — это место, в котором вы систематизируете свой проект. Здесь вы храните файлы, папки, видео, изображения, блокноты Jupyter Notebook, наборы данных и т.д. Перед началом работы с Git необходимо инициализировать репозиторий для проекта и правильно его подготовить. Это можно сделать на сайте GitHub.

Лучше сразу добавлять в репозиторий **README**-файл с информацией о проекте. Это можно сделать в момент создания репозитория, поставив галочку в соответствующем поле.

- Перейдите на сайт GitHub. Нажмите на значок + в верхнем правом углу, а затем выберите **New repository**.
- Придумайте имя репозитория и добавьте короткое описание.
- Решите, будет ли этот репозиторий размещаться в открытом доступе или останется закрытым для просмотра.

- Нажмите **Initialize this repository with a README** для добавления README-файла. Настоятельно рекомендую снабжать все ваши проекты файлом-описанием, ведь README — это первая вещь, на которую люди обращают внимание при просмотре репозитория. К тому же, здесь можно разместить нужную информацию для понимания или запуска проекта.



Новый репозиторий

Create a new repository

A repository contains all project files, including the revision history.

Owner: bonn0062

Repository name *: github_welcome ✓

Great repository names are short and memorable. Need inspiration? How about fantastic-meme?

Description (optional): This is the official repository for the article, "Getting started with GitHub." Welcome to your very first open

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

Private: You choose who can see and commit to this repository.

Initialize this repository with a README: This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None

Create repository

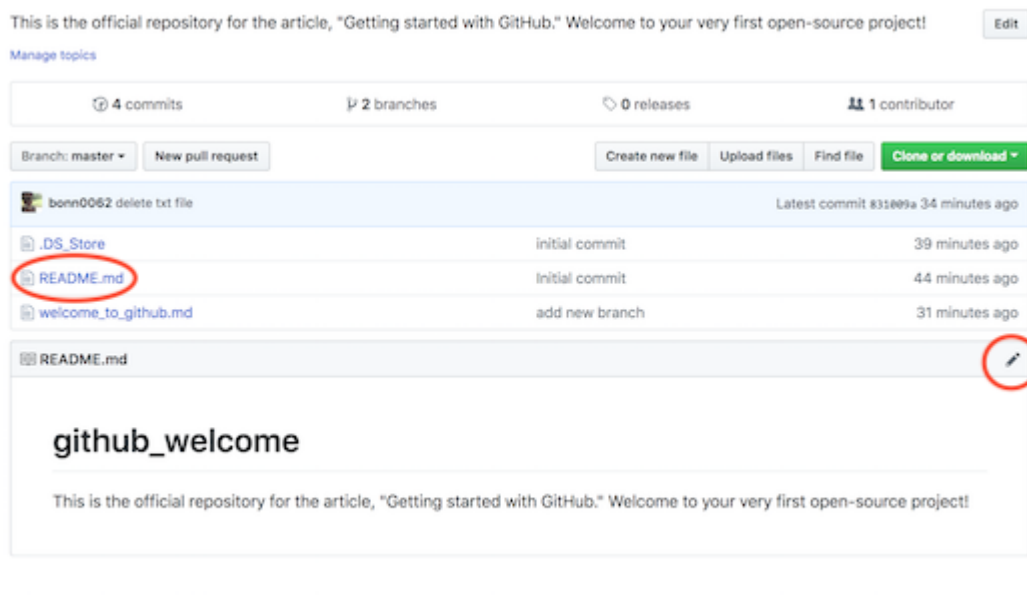
Создание нового репозитория

При желании можете уже сейчас начинать работать над проектом. Добавляйте файлы, вносите в них изменения и т.д. напрямую с сайта GitHub. Однако конечный результат подобной деятельности может вас немного огорчить.

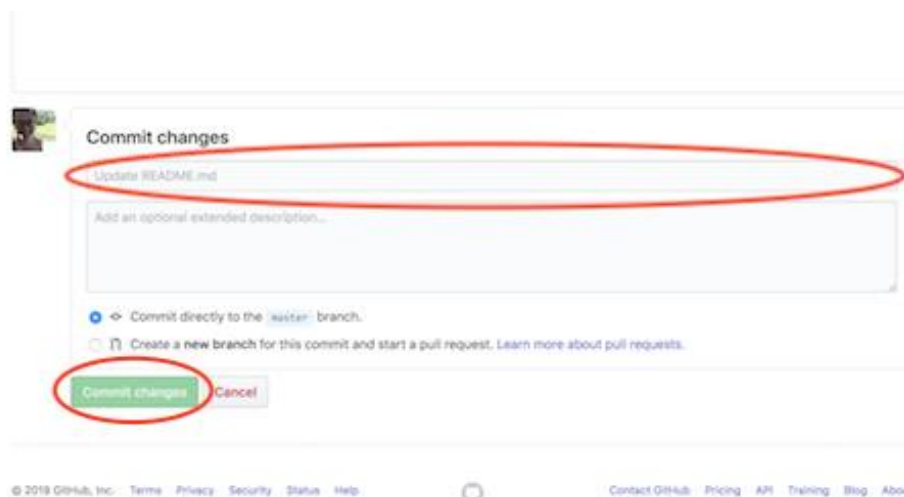
Вносить изменения в проект можно двумя способами. Вы можете изменять файлы/блокноты на компьютере либо делать это на сайте GitHub.

Допустим, вам захотелось подкорректировать README-файл на сайте GitHub.

- Для начала перейдите в ваш репозиторий.
- Для выбора файла кликните по его названию (например, кликните по **README.md** для перехода к файлу-описанию).
- В верхнем правом углу вы увидите иконку с карандашом. Нажмите на нее для внесения изменений.
- Напишите короткое сообщение, передающее суть изменений (и подробное описание, если сочтете это нужным).
- Нажмите кнопку **Commit changes**.



Изменение файла на GitHub

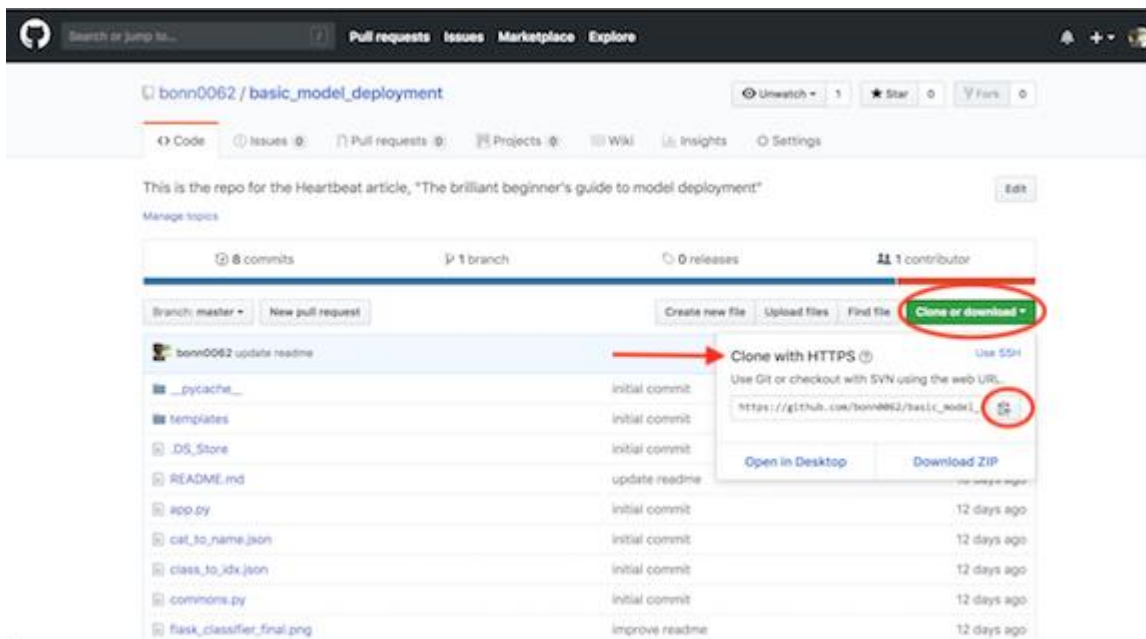


Подготовка коммита с изменениями

Вы успешно внесли изменения в README-файл своего нового репозитория! Обратите внимание на небольшую кнопку на картинке выше. Она позволяет создавать новую ветку этого коммита и добавлять Pull request (залить изменения, а вот заливать или не заливать решает владелец репозитория).

Для **клонирования репозитория** на компьютер перейдите в репозиторий на GitHub и нажмите большую зеленую кнопку под названием **Clone or download** (разумеется, вы можете просто скачать

репозиторий и избежать всех заморочек с терминалом. Но я в вас верю, поэтому не будем сдаваться!). Проследите, чтобы появилась надпись **Clone with HTTPS**. Теперь нажмите на иконку буфера обмена для копирования-вставки (либо выделите ссылку и скопируйте ее).



Клонирование или скачивание репозитория

Откройте **терминал** и перейдите в директорию для копирования репозитория. Например, для перехода на **Рабочий стол** напечатайте вот это:

```
cd Desktop
```

Затем клонируйте туда репозиторий по следующей команде:

```
git clone <то_что_вы_только_что_скопировали>
```

Все просто! Не забудьте изменить информацию в угловых скобках на нужную вам. И удалите сами скобки < >.

Если вы не очень хорошо ориентируетесь в терминале, то переход по директориям можно осуществлять через команду `cd`. Например, откройте терминал и напечатайте `ls` для отображения перечня доступных директорий. Вполне возможно, что в этом списке вы сразу увидите директорию `Desktop`. Либо напечатайте `cd Desktop`. Далее выполните команду `git clone` и склонируйте репозиторий на Рабочий стол.

Бывает и так, что вместо перечня расположений, вы видите различные имена пользователей. Тогда до того, как перейти в `Desktop`, вам потребуется выбрать нужного пользователя через команду `cd <пользователь>` (замените <пользователь> на нужное вам имя). Затем снова напечатайте `ls`, чтобы увидеть весь список. И вот теперь, увидев в списке `Desktop`, смело печатайте `cd Desktop`. Сейчас уже можно выполнять `git clone`!

Если вдруг в терминале вы захотите «откатиться» на шаг назад, то напишите `cd ..`

Новый GitHub-репозиторий, клонированный на рабочий стол, готов! Данная команда создает точную копию репозитория в вашей системе. Здесь вы сможете с ним работать, редактировать, индексировать изменения, создавать коммиты с изменениями и отправлять их на GitHub.

Зарегистрируйтесь на ресурсе [github](https://github.com), сообщите свой логин преподавателю для настройки общего репозитория <https://github.com/sushkovayulia/p17-1.git>.

Переместите локальный репозиторий на диск `d` в папку класса.