Урок 2. Настройка GitHub. Введение в HTML.

# Настройка GitHub

GitHub – это место для хранения кода. Для первоначальной настройки необходимо выполнить следующие действия:

1. Зарегистрироваться на <https://github.com>
2. Создать новый репозиторий (репозиторий – хранилище для отдельного проекта).
3. Склонировать (перенести) созданный репозиторий на компьютер

Рассмотрим шаги более подробно.

## Регистрация на GitHub

Для регистрации необходимо зайти на <https://github.com> и нажать кнопку Sign Up (Регистрация) в правом верхнем углу. После чего нужно ввести требуемые данные и завершить регистрацию. Процесс достаточно быстрый и простой. Более подробно это рассмотрено в видео по ссылке (часть с регистрацией GitHub): <https://www.youtube.com/watch?v=7Auu4VOsoQw&t=59s>

## Создание репозитория

После регистрации на GitHub нам необходимо создать новый репозиторий. Всего за время обучения у нас будет 2 репозитория: в одном будут храниться все наши домашние работы, в другом – итоговая работа. Для начала создадим новый репозиторий для домашних работ.

После окончания регистрации вы попадёте на главную страницу и нужно найти там зелёную кнопку, на которой будет написано New (или «Новый репозиторий»). Нажимаем эту кнопку и заполняем данные:

1. Repository Name (имя репозитория)  
   Обязательное поле. Мы условились, что репозитории называем в следующем формате: **4p22-frontend-andrew-gulin,** где вместо andrew-gulin ваше имя и фамилия.
2. Description (Описание)  
   Можно оставить пустым, либо написать пояснение, что данный репозиторий для домашних работ.
3. Public / Private (Общедоступный / Частный)  
   Здесь оставляем как изначально выбрано: **public** (публичный, общедоступный)
4. Ставим галочку **Add a README file (добавить файл README)**
5. Остальное оставляем как есть и нажимаем **Create Repository (Создать Репозиторий).**

Более подробно можно посмотреть в видео по ссылке (часть с созданием репозитория):

<https://www.youtube.com/watch?v=7Auu4VOsoQw&t=470s>

## Клонирование репозитория

Чтобы вносить какие-то изменения, нам нужно получить локальную копию репозитория на свой компьютер. В целом принцип работы такой же, как и у Dropbox, Яндекс.Диск, только синхронизация не выполняется автоматически: нам необходимо самим создать локальную копию и при внесении изменений синхронизировать с удалённой копией в GitHub.

Процесс клонирования состоит из нескольких этапов:

1. Удостовериться в том, что Git установлен на компьютере
2. Добавить ключ SSH (для безопасного взаимодействия вашего компьютера с GitHub)
3. Склонировать репозиторий

Давайте подробнее рассмотрим каждый шаг.

1 Установка Git на компьютер

Если у вас MacOS или любая сборка Linux, то делать ничего не нужно: Git уже должен быть предустановлен. На этом этапе нужно просто открыть приложение Терминал.

Если у вас Windows, то необходимо установить Git. Для этого я рекомендую скачать Git с сайта <https://gitforwindows.org> и установить его. Процесс должен быть лёгким. Более подробно это рассмотрено в видео по ссылке: <https://www.youtube.com/watch?v=7Auu4VOsoQw&t=626s>

2 Добавить ключ SSH

Для безопасного взаимодействия вашего компьютера с GitHub (или любым другим хранилищем) необходимо пользоваться SSH ключами. Нужно сгенерировать ключи (генерируется публичный и приватный ключ) и затем публичный ключ разместить в GitHub).

Открываем GitBash для Windows и Терминал для Linux и MacOS (можно воспользоваться поиском по приложениям) и пишем в открывшемся окне **ssh-keygen.**

После этого консоль задаст несколько «вопросов», писать ничего не нужно, просто нажимать enter.

Далее необходимо получить публичный ключ. Это мы делаем с помощью команды

**cat ~/.ssh/id\_rsa.pub**

После этого в консоль выведется ключ, который нужно скопировать (начинается с ssh-rsa).

Копировать лучше не через ctrl-c, а с помощью правой кнопки мыши -> скопировать.

Для добавления публичного ключа в GitHub необходимо зайти в GitHub, нажать на иконку профиля в правом верхнем углу, далее **Settings (Настройки)** и в левом столбце выбрать **SSH and GPG keys.** Нажать зелёную кнопку **New SSH key**.

Вы должны увидеть два поля ввода: маленькое и большое. В маленьком нужно написать название для этого ключа. Так как ключей может быть несколько, я обычно использую в качестве названия имя устройства, на котором этот ключ создавал: например, MacBook Pro 15. В нижнее большое поле нужно вставить скопированный из Git Bash или из Терминала сгенерированный публичный ключ. После чего нажать кнопку **Add SSH key (добавить SSH ключ).**

Более подробно можно посмотреть в видео по ссылке <https://www.youtube.com/watch?v=7Auu4VOsoQw&t=1730s>

3 Клонирование репозитория

Теперь самый последний шаг, который нужно сделать в первоначальной настройке: клонирование.  
Для начала нам нужно создать отдельную папку для курса, где будут удобно размещаться все материалы и наши репозитории. Создать папку можно где угодно.

После этого нужно открыть эту папку в GitBash или в Терминале. Для GitBash всё достаточно просто: зайти в эту папку в проводнике, нажать правую кнопку мыши и выбрать GitBash Here. Откроется окно GitBash.

Для терминала можно скопировать путь до папки в Finder, открыть терминал и написать команду

**cd ПУТЬ**

где ПУТЬ – скопированный путь для созданной папки.

Теперь мы готовы клонировать репозиторий.

Нужно сначала скопировать ссылку репозитория на GitHub.com. Для этого нужно зайти в сам репозиторий (обычно их список располагается слева на главной странице github). Нажать зелёную кнопку **Code,** выбрать **SSH** и скопировать ссылку, появившуюся ниже.

В уже открытом Терминале или GitBash написать команду

**git clone АДРЕС**

где адрес – ссылка, которую мы только что скопировали.  
Нажимаем Enter и репозиторий успешно клонируется, после чего в нашей папке создастся ещё одна папка с названием нашего репозитория. Зайдя в неё, можно увидеть файл README.

Подробнее можно посмотреть в видео по ссылке: <https://www.youtube.com/watch?v=7Auu4VOsoQw&t=1858s>

# Работа с GitHub

После настройки GitHub и клонирования репозитория нам нужно вносить изменения в этот репозиторий (выполнять домашние работы) и синхронизировать то, что создали локально с GitHub.

## Структура домашних работ

Для каждой домашней работы внутри репозитория необходимо создать папку с названием **homework\_1,** где цифра в конце – номер домашней работы.

Создавать папки нужно прямо в репозитории (там, где вы увидели файл README).

## Синхронизация с GitHub

Создадим папку homework\_1 для теста. Зайдём в неё и создадим файл **hello.txt.**

Мы внесли изменения в репозиторий, теперь нужно их синхронизировать.

Заходим в проводнике в папку с репозиторием (не с домашней работой, а именно с репозиторием, вы должны видеть файл README и список папок домашних работ).

Открываем GitBash в этой папке (правой кнопкой мыши и GitBash here).

Для MacOS и Linux – копируем путь до папки и в открытом терминале переходим в эту директорию с использованием команды **cd.**

Теперь нам нужно выполнить 3 действия:

1. Добавить файлы в «индекс» – чтобы указать файлы, которые мы будем синхронизировать. Обычно это все файлы, которые мы изменили, поэтому команда будет выглядеть всегда одинаково:  
   **git add .   
   (точка обязательна в конце)**
2. Далее нам нужно создать коммит (список изменений с каким-то комментарием)  
   **git commit -m “add homework\_1”**В кавычках пишем на английском языке, что было сделано, что мы только что изменили и хотим залить в GitHub.
3. После создания коммита нужно отправить изменения в GitHub:  
   **git push**

Если всё прошло успешно, то теперь, открыв репозиторий в GitHub, вы увидите рядом с файлом README папку homework\_1, а зайдя в неё – файл hello.txt. Успех!

Более подробно можно посмотреть в видео по ссылке <https://www.youtube.com/watch?v=7Auu4VOsoQw&t=2114s>

# Сдача домашних работ

Для сдачи домашних работ необходимо загружать не только ссылку на код, но ещё и сам файл. Поэтому нам потребуется на компьютере зайти в папку репозитория, выбрать папку нужной домашней работы и заархивировать (архив в репозитории не оставлять, просто можно его потом удалить).

Архивировать просто: правой кнопкой на папке с домашней работой и там будет что-то наподобие «Добавить в архив» (зависит от системы и языка). После этого созданный файл архива прикрепить в системе (инструкция должна быть от организаторов).

Далее в описание к домашней работе нужно вставить ссылку из GitHub. Для этого нужно открыть репозиторий и зайти в папку homework\_1 (или в ту, которую сдаём). И просто скопировать сверху адрес, потом вставить ссылку в описание к домашней работе.

Более подробно это будет рассмотрено непосредственно на **уроке 2**.

# HTML

Также сегодня поговорим о языке разметки HTML. В качестве самостоятельной подготовки предлагаю несколько источников, которые можно изучить:

1. <https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML>
2. <http://htmlbook.ru/samhtml/vvedenie-v-html>