**Практическая работа. Создание репозитория в GitHub Pages?**

[GitHub](https://github.com/) — это сайт «социальной разработки». Он позволяет загружать репозитории кода для хранения **в системе управления версиями** [Git](http://git-scm.com/%22%20%5Ct%20%22_blank). После загрузки вы можете сотрудничать с другими участниками для работы над общим проектом, а ещё это система с открытым исходным кодом по умолчанию: любой человек в мире может найти ваш код на GitHub, использовать его, учиться по нему и улучшать его. Вы тоже можете взаимодействовать с кодом других людей! В этой статье представлено базовое руководство по публикации контента с использованием функции gh-pages Github.

[**Публикация контента**](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Tools_and_setup/Using_Github_pages#%D0%BF%D1%83%D0%B1%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%BA%D0%BE%D0%BD%D1%82%D0%B5%D0%BD%D1%82%D0%B0)

GitHub — очень важное и полезное сообщество для участия, а Git/GitHub — очень популярная [система управления версиями](http://git-scm.com/book/ru/v2/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%9E-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B5-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F-%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9) — большинство технологических компаний теперь использует её в своём рабочем процессе. У GitHub есть очень полезная функция [GitHub Pages](https://pages.github.com/%22%20%5Ct%20%22_blank), которая позволяет публиковать живую демонстрацию кода в виде сайта в Интернете.

[**Базовая установка Github**](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Tools_and_setup/Using_Github_pages#%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F_%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0_github)

1. Прежде всего, [установите Git](https://git-scm.com/downloads) на свой компьютер. Это базовое программное обеспечение для управления версиями, поверх которого работает GitHub.
2. Затем [зарегистрируйтесь на GitHub](https://github.com/). Это просто.
3. После того, как вы зарегистрировались, войдите в свою учётную запись на [github.com](https://github.com/) с помощью своих имени пользователя и пароля.

[**Подготовка кода для загрузки**](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Tools_and_setup/Using_Github_pages#%D0%BF%D0%BE%D0%B4%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BA%D0%BE%D0%B4%D0%B0_%D0%B4%D0%BB%D1%8F_%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8)

В репозитории на Github вы можете хранить любой код, который вам нравится, но для полноценного использования функции GitHub Pages ваш код должен быть структурирован как типичный веб-сайт, то есть, основной точкой входа должен быть HTML-файл с именем index.html.

Прежде чем двигаться дальше, вам необходимо инициализировать вашу папку (директорию) с кодом как репозиторий Git. Как это сделать:

1. Направьте командную строку в вашу папку test-site (или папку с любым другим названием, содержащую ваш сайт). Для этого используйте команду cd (т.е. «**c**hange **d**irectory» — «поменяй директорию»). Вот что вы бы набрали, если бы разместили свой сайт в папке test-site на рабочем столе:

BASHCopy to Clipboard

cd Desktop/test-site

1. Когда командная строка указывает внутрь вашей папки веб-сайта, введите следующую команду, которая сообщает инструменту git, что папку нужно превратить в репозиторий git:

BASHCopy to Clipboard

git init

**Дополнительно об интерфейсах командной строки**

Лучший способ загрузить код в Github — использовать командную строку. Это окно, в котором вы вводите команды, чтобы делать такие вещи, как создавать файлы и запускать программы, вместо того чтобы кликать внутри пользовательского интерфейса. Оно будет выглядеть примерно так:



**Примечание:** вы также можете использовать [графический пользовательский интерфейс Git](https://git-scm.com/downloads/guis) для выполнения тех же задач, если вам неудобно работать с командной строкой.

Каждая операционная система поставляется с инструментом командной строки:

* **Windows**: **Командную строку** (**Command Prompt**) можно вызвать, нажав клавишу Windows, набрав *Command Prompt* и выбрав её из появившегося списка. Обратите внимание, что в Windows свои собственные условные обозначения команд, отличные от Linux и OS X, поэтому приведённые ниже команды могут отличаться на вашем компьютере.
* **OS X**: **Терминал** (**Terminal**) можно найти в *Applications > Utilities*.
* **Linux**: Обычно терминал можно вызвать с помощью *Ctrl + Alt + T*. Если это не сработает, найдите **Terminal** в панели приложений или меню.

Сначала это может показаться немного пугающим, но не волнуйтесь — вы быстро разберётесь в основах. Вы говорите компьютеру что-то сделать в терминале, введя команду и нажав Enter, как показано выше.

[**Создание репозитория для вашего кода**](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Tools_and_setup/Using_Github_pages#%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D1%8F_%D0%B4%D0%BB%D1%8F_%D0%B2%D0%B0%D1%88%D0%B5%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0)

1. Затем вам нужно создать новый репозиторий для ваших файлов. Нажмите «Плюс» (+) в правом верхнем углу главной страницы GitHub и выберите «Новый репозиторий» (*New Repository*).
2. На этой странице в поле «Имя репозитория» (*Repository name*) введите имя для своего репозитория, например *my-repository*.
3. Также добавьте описание, чтобы уточнить, что будет содержать ваш репозиторий. Ваш экран должен выглядеть так:
4. Нажмите «Создать репозиторий» (*Create repository*); откроется следующая страница:

[**Загрузка файлов на GitHub**](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Tools_and_setup/Using_Github_pages#%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2_%D0%BD%D0%B0_github)

1. На текущей странице вас интересует раздел *…or push an existing repository from the command line* («...или загрузите существующий репозиторий с помощью командной строки»). В этом разделе вы должны увидеть две строки кода. Скопируйте всю первую строку, вставьте её в командную строку и нажмите Enter. Команда должна выглядеть примерно так:
2. git remote add origin https://github.com/chrisdavidmills/my-repository.git
3. Затем введите следующие две команды, нажимая Enter после каждой из них. Они готовят код для загрузки в GitHub и просят Git управлять этими файлами.

BASHCopy to Clipboard

git add --all

git commit -m 'adding my files to my repository'

1. Наконец, загрузите код на GitHub, для чего вернитесь к текущей веб-странице GitHub и введите в терминал вторую из двух команд, которые мы увидели в секции *…or push an existing repository from the command line*:

BASHCopy to Clipboard

git push -u origin main

1. Теперь вам нужно включить функционал GitHub Pages для своего репозитория. Чтобы сделать это, зайдите на страницу своего репозитория и выберите «Настройки» (*Settings*), затем выберите *Pages* на панели слева. В разделе *Source* выберите ветку main. Страница должна обновиться.
2. Перейдите в раздел GitHub Pages снова. Вы должны увидеть надпись «Ваш сайт готов к публикации на https://xxxxxx» («Your site is ready to be published at https://xxxxxx»).
3. Если вы кликните на ссылку, вам должна открыться страница с живой демонстрацией вашего примера, при условии, что домашняя страница называется index.html — по умолчанию ссылка ведет именно на неё. Если точка входа на ваш сайт называется как-то иначе, например myPage.html, то чтобы увидеть её, вам нужно открыть https://xxxxxx/myPage.html.

[**Больше информации о GitHub**](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Tools_and_setup/Using_Github_pages#%D0%B1%D0%BE%D0%BB%D1%8C%D1%88%D0%B5_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8_%D0%BE_github)

Если вы хотите поменять что-то на своем тестовом сайте и загрузить результат на GitHub, вам просто нужно внести изменения в свои файлы, как и раньше. Затем вам потребуется ввести следующие команды (нажимая Enter после каждой), чтобы загрузить их на GitHub:

git add --all

git commit -m 'another commit'

git push

Вы можете написать более подходящее сообщение вместо *another commit*, чтобы уточнить, что именно вы только что изменили.

Мы едва коснулись Git. Чтобы узнать больше, начните с нашей статьи [Git и GitHub](https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/GitHub).