



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО МГТУ «СТАНКИН»)

Институт
информационных технологий

Кафедра
информационных систем

Отчет по лабораторной работе № 1.1

по дисциплине **«Веб-программирование»**

на тему: Основы работы с системой Git и с сервисом GitHub.

Студентка
группа ИДБ–19–05

Ле Тхи Хань Хуен

Преподаватель

подпись

Кайшев Д.А.

подпись

Москва 2021 г.

ГЛАВА I: ОСНОВНЫ РАБОТЫ С СИСТЕМОЙ GIT

Git – это система управления версиями. У Git две основных задачи: первая - хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая - обеспечение удобства командной работы над кодом.

Репозиторий Git — это место, где хранится ваш код и вся информация о его изменениях. Репозитории могут находиться у вас на компьютере, на компьютерах ваших коллег и на удалённом сервере.

Обычно работа с Git выглядит так:

1. Сверстали шапку сайта – сделали commit
2. Сверстали контент страницы – сделали второй commit
3. Закончили верстать страницу – сделали третий commit и отправили код на сервер, чтобы вашу работу могли увидеть коллеги, либо чтобы опубликовать страницу с помощью.

Основы работы с Git:

1. Инициализируем git:

```
git init
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Website> git init
Initialized empty Git repository in E:/Website/.git/
PS E:\Website> █
```

2. Проверка ветки

Проверьте ветку, на которой вы находитесь. Для этого введите следующую команду в терминал:

```
git branch
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Website> git init
Reinitialized existing Git repository in E:/Website/.git/
PS E:\Website> git branch
* main
PS E:\Website> █
```

Можно написать новый код непосредственно в ветку main при условии, что вы не работаете с директорией приложения, содержащей несколько папок и файлов. Но если вы планируете работу над большим проектом, то лучше этого не делать.

Написание кода прямо в ветку main может повлечь за собой его отправку в онлайн-репозиторий без каких-либо мер предосторожности, а мы же, наоборот, хотим убедиться, что отправленный код работает должным образом.

3. Переключение на новую ветку

Во избежание сложностей, связанных с написанием кода прямо в ветку main, следует создать новую ветку на своем локальном компьютере. Для этого введите следующее в терминал:

```
git checkout -b
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Website> git init
Reinitialized existing Git repository in E:/Website/.git/
PS E:\Website> git branch
* main
PS E:\Website> git checkout -b NewMain
Switched to a new branch 'NewMain'
PS E:\Website> █
```

Имя ветки должно отражать проблему, ради решения которой она и создается.

4. Написание кода

5. Просмотр статуса

Используя команду `git status`, мы сможем увидеть все файлы, которые были добавлены, изменены или удалены во время нашего искрометного сеанса программирования.

```
git status
```

```
PS E:\Website> git status
On branch NewMain
nothing to commit, working tree clean
PS E:\Website> █
```

В филиале `NewMain` нечего делать, рабочее дерево чистое.

Если все в полном порядке, то для индексации этих изменений просто введите в терминал следующую команду:

```
git add .
```

6. Создание коммита

Проиндексировав изменения, мы переходим к их фиксации в коде. `Git commit` сохраняет эти изменения и готовит их для отправки в онлайн-репозиторий. `Git` предлагает написать комментарий к коммиту.

```
git commit -m "useful info about the code | wrote"
```

```
PS E:\Website> git add .
PS E:\Website> git commit -m"add new project"
[main (root-commit) 1f98e3f] add new project
5 files changed, 107 insertions(+)
create mode 100644 main.html
create mode 100644 pic/pic1.JPG
create mode 100644 pic/pic2.jpg
create mode 100644 style.css
```

Флаг `-m` указывает команде `git commit`, что вы добавляете полезный комментарий о коде.

7. Отправка

Подключиться по ссылке для загрузки файла на GitHub.

```
git remote add origin "http-адрес"
```

Загрузить данные на GitHub.

```
git push -u origin
```

```
PS E:\Website> git remote add origin https://github.com/LeHuyenn/NewRespo.git
PS E:\Website> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
```

8. Проверить данные на GitHub

После использования команды `git push -u`, заметил что данные с машины были загружены на GitHub

The screenshot shows the GitHub interface for a repository named 'LeHuyenn add new project'. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below these, there are buttons for 'Go to file', 'Add file', and 'Code'. The repository is currently on the 'main' branch, with 1 branch and 0 tags. The commit history shows a single commit '1f98e3f' from 2 minutes ago with 1 commit. The file list includes 'pic', 'main.html', 'main2.html', and 'style.css', all added 2 minutes ago. At the bottom, there is a prompt to 'Add a README'.

9. Этап 9: Загрузите данные с GitHub на свой компьютер

На данный момент ветка `main` в онлайн-репозитории содержит больше кода, чем в локальном. Надо бы это исправить. Вернитесь на ветку `master` и введите в терминал следующую команду:

```
Git pull
```

```

PS E:\Website> git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 779 bytes | 111.00 KiB/s, done.
From https://github.com/LeHuyenn/NewRespo
  1f98e3f..0c42322  main      -> origin/main
Updating 1f98e3f..0c42322
Fast-forward
 main.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
PS E:\Website> git add .
PS E:\Website> git commit -m"delete 1 tag"
[main 5761009] delete 1 tag
 1 file changed, 1 insertion(+), 1 deletion(-)

```

10.Итого

На этом этапе вы должны определиться с функциональностями, которые хотели бы разрабатывать в дальнейшем. Начинайте процесс с этапа 3, и пусть осознание того, что вы четко выполняете все правила работы с Git.

ГЛАВА II: ОСНОВНЫ РАБОТЫ С СЕРВИСОМ GitHub

GitHub - сервис онлайн-хостинга репозиторий, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Также GitHub может похвастаться контролем доступа, багтрекером, управлением задачами и вики для каждого проекта.

Особенности и ограничения:

Сервис GitHub предоставляет следующие возможности для статических сайтов:

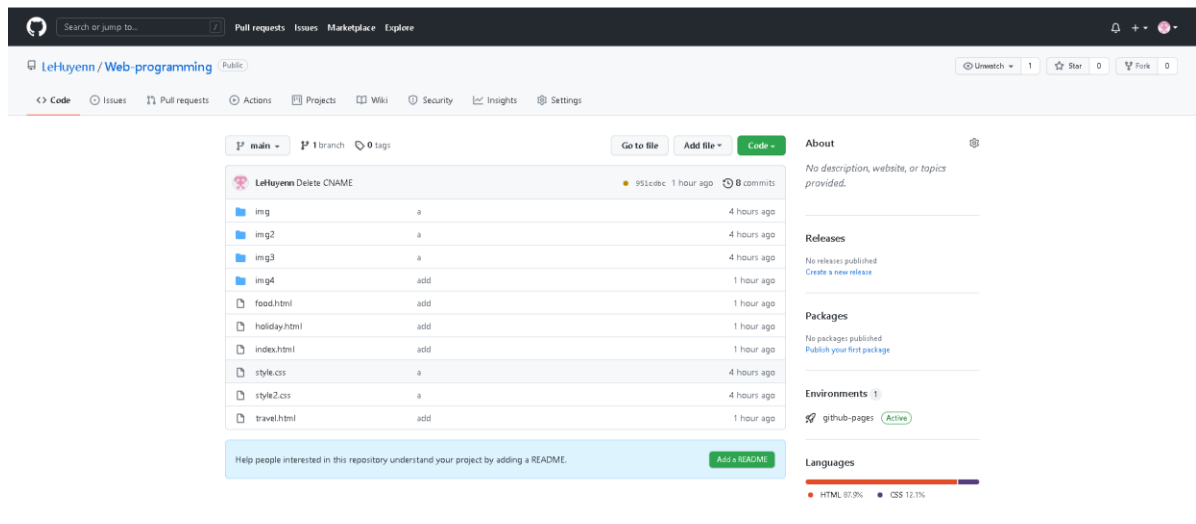
- Использование HTML, CSS, языка разметки Markdown;
- Встраивание изображений и другого медиа;
- Использование JavaScript.

Ограничения:

- Нельзя использовать на сайте PHP либо другие серверные языки;
- Серверный код и серверные скрипты выполняться не будут;
- Cookies не используются.

Как опубликовать сайт на GitHub

1. Создайте аккаунт на GitHub. Предположим, *your-account-name* — название вашего аккаунта.
2. Создайте новый репозиторий для вашего проекта. При создании введите название репозитория (например: **Web-programming**), выберите тип репозитория **Public** и нажмите на кнопку **Create repository**:
3. Загрузите все файлы вашего проекта в ветку **master** созданного репозитория **your-name**. При этом файл **index.html** должен находиться в корневой директории проекта:



4. Перейдите в настройки (**Settings**) созданного репозитория:
5. В настройках репозитория найдите вкладку **Options**:
6. Прокрутите вниз и найдите раздел **Github Pages**. Установите в качестве источника файлов вашей страницы ветку **master** созданного репозитория:
7. Теперь ваш сайт опубликован по адресу вида:

<https://lehuenn.github.io/Web-programming/>

Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Actions
Environments
Secrets
Pages
Moderation settings

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://lehuyenn.github.io/Web-programming/>

Source

Your GitHub Pages site is currently being built from the main branch. [Learn more.](#)

Branch: main

/ (root)

Save

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Custom domain

Custom domains allow you to serve your site from a domain other than `lehuyenn.github.io`. [Learn more.](#)

Save

Remove

☒ Enforce HTTPS

— Required for your site because you are using the default domain (`lehuyenn.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)



Publish privately to people with read access to this repository

Try it risk-free for 14 days using a GitHub Enterprise organization, or [learn more about changing the visibility of your GitHub Pages site.](#)

