

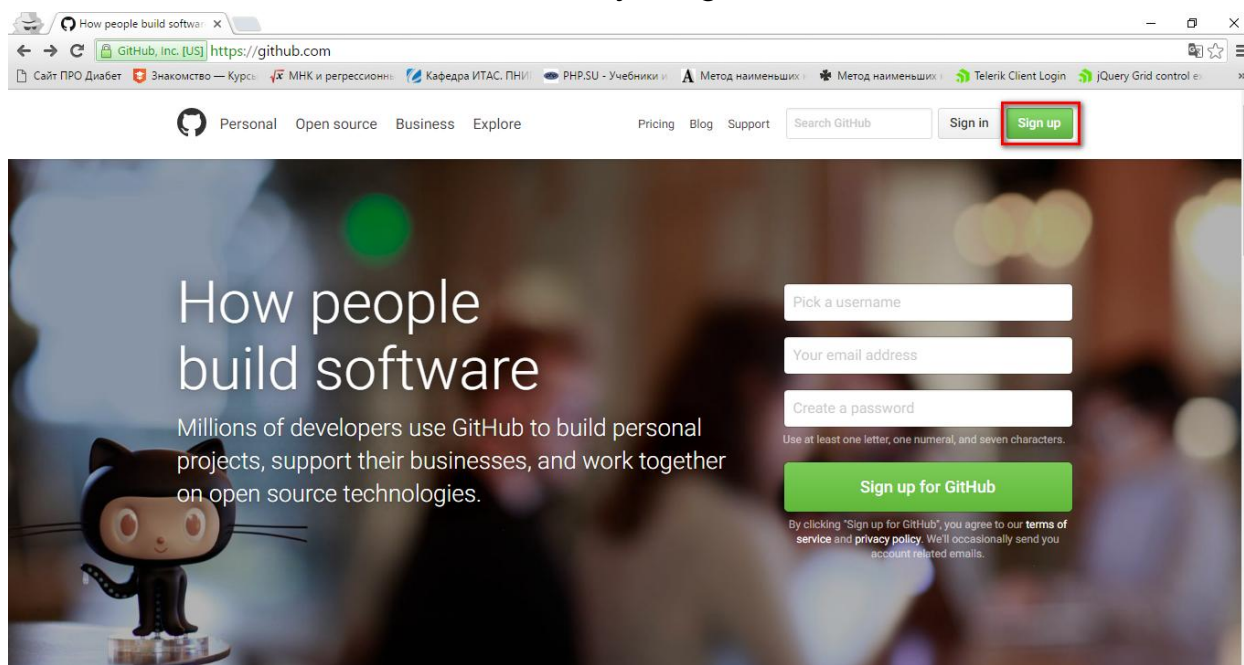
Небольшая инструкция о том, как пользоваться хранилищем GitHub.com

1). Создание хранилища (репозитория).

В подгруппах создается только один репозиторий для всех членов этой подгруппы, то есть этот пункт выполняет один человек из подгруппы, а остальные этот пункт пропускают.

- Заходим на сайт <https://github.com/>
- Регистрируемся на этом сайте.

Для этого нажимаем на кнопку «Sign UP».



После чего откроется окно регистрации (1 шаг) для заполнения информации. После заполнения всех данных, необходимо нажать на кнопку «Create an account».

Join GitHub
The best way to design, build, and ship software.

Step 1: Set up a personal account | Step 2: Choose your plan | Step 3: Tailor your experience

Create your personal account

Username: MatveiKalin23 ✓

This will be your username — you can enter your organization's username next.

Email Address: MatthewKalin@yandex.ru ✓

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password: ✓

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

You'll love GitHub

- Unlimited collaborators
- Unlimited public repositories
- Great communication
- Frictionless development
- Open source community

На 2 шаге мы выбираем неограниченное количество публичных репозиторий. После чего нажимаем кнопку «Continue».

Welcome to GitHub
You've taken your first step into a larger world, @MatveiKalin23.

Completed: Set up a personal account | Step 2: Choose your plan | Step 3: Tailor your experience

Choose your personal plan

☒ Unlimited public repositories for free.

☐ Unlimited private repositories for \$7/month. ([view in RUB](#))

Don't worry, you can cancel or upgrade at any time.

☐ Help me set up an organization next

Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees. [Learn more about organizations.](#)

Continue

Both plans include:

- Collaborative code review
- Issue tracking
- Open source community
- Unlimited public repositories
- Join any organization

На 3 шаге укажите свои предпочтения, это информация нужна для статистики руководству GitHub. Если хотите пропустить этот шаг, можно нажать на ссылку «skip this step», внизу.

You'll find endless opportunities to learn, code, and create, @MatveiKalin23.

Completed Set up a personal account Step 2: Choose your plan Step 3: Tailor your experience

How would you describe your level of programming experience?

☐ Very experienced ☐ Somewhat experienced ☒ Totally new to programming

What do you plan to use GitHub for? (check all that apply)

☒ School projects ☐ Design ☐ Research
☒ Project Management ☒ Development ☐ Other (please specify)

Which is closest to how you would describe yourself?

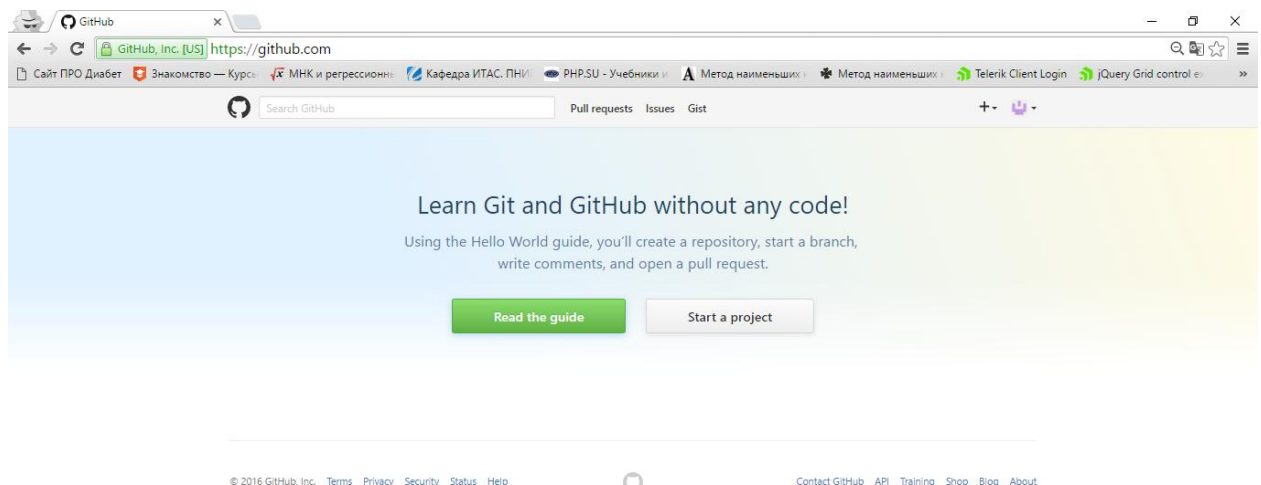
☐ I'm a hobbyist ☒ I'm a student ☐ I'm a professional
☐ Other (please specify)

What are you interested in?

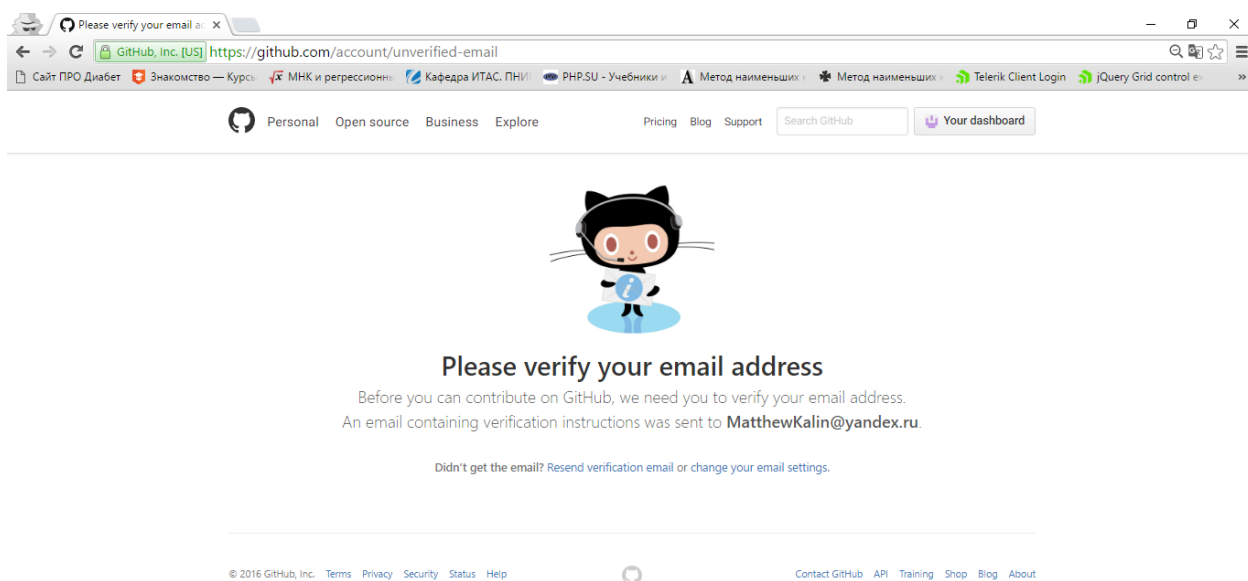
e.g. tutorials, android, ruby, web-development, machine-learning, open-source

Submit skip this step

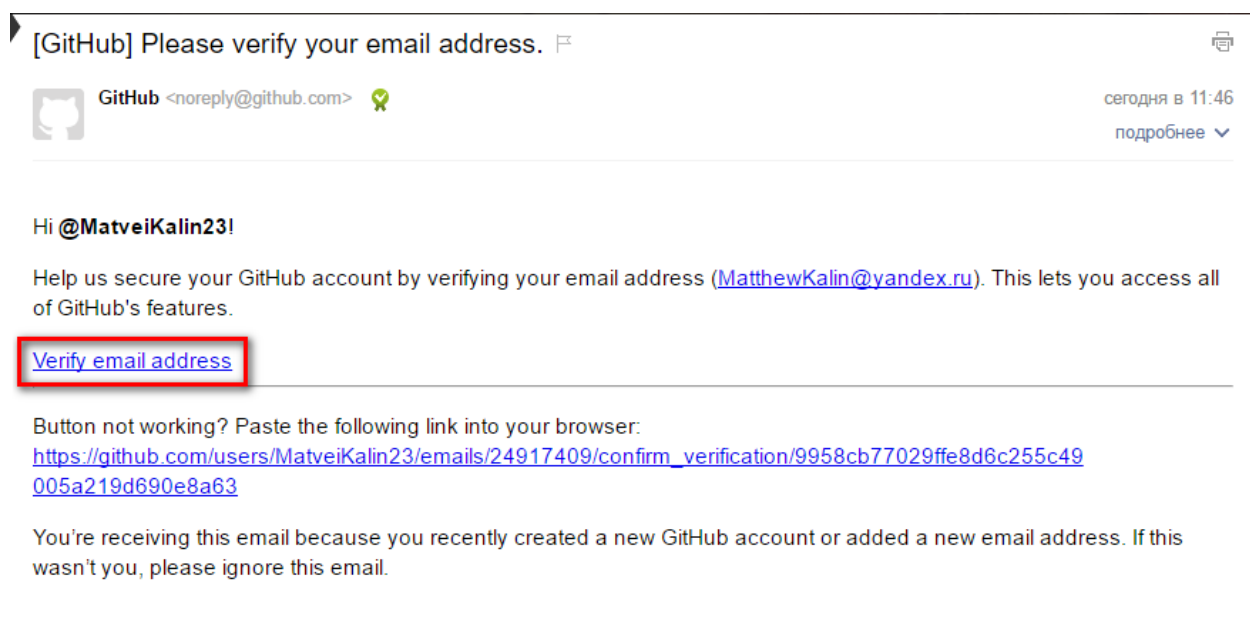
- После чего появится вот такое окно:
Зеленая кнопка ведет к руководству, но она на английском языке.
А эта ссылка ведет к руководству на русском языке <https://git-scm.com/book/ru/v1>
Моя же инструкция краткая, чтобы использовать только те возможности хранилища на GitHub, которые пригодятся для лабораторных (Я могу упустить что-то важное). Ссылка выше для тех, кто хочет полностью узнать возможности GitHub.



Если же нажать на кнопку «Start a project», то GitHub попросит проверить свою электронную почту, указанную во время регистрации.



На этот указанный адрес должно придти письмо от GitHub, нужно подтвердить о том, что это Вы регистрировались, нажав на ссылку «Verify email address» в письме.



- После подтверждения личности Вас перебросит снова на страницу, которую мы уже видели, но если мы нажмем на кнопку «Start a project». GitHub попросит ввести имя нового репозитория (хранилища).

2). Создание репозитория (хранилища).

Этот пункт также выполняется только одним человеком из подгруппы.

Create a new repository

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

Owner: MatveiKalin23

Repository name: my_repository

Great repository names are short and memorable. Need inspiration? How about [bookish-guide](#).

Description (optional): Тестовое хранилища

☒ 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

После чего репозиторий создастся.

MatveiKalin23 / my_repository

Watch 0 Star 0 Fork 0

Code Issues Pull requests Wiki Pulse Graphs Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/MatveiKalin23/my_repository.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# my_repository" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/MatveiKalin23/my_repository.git
git push -u origin master
```

...or push an existing repository from the command line

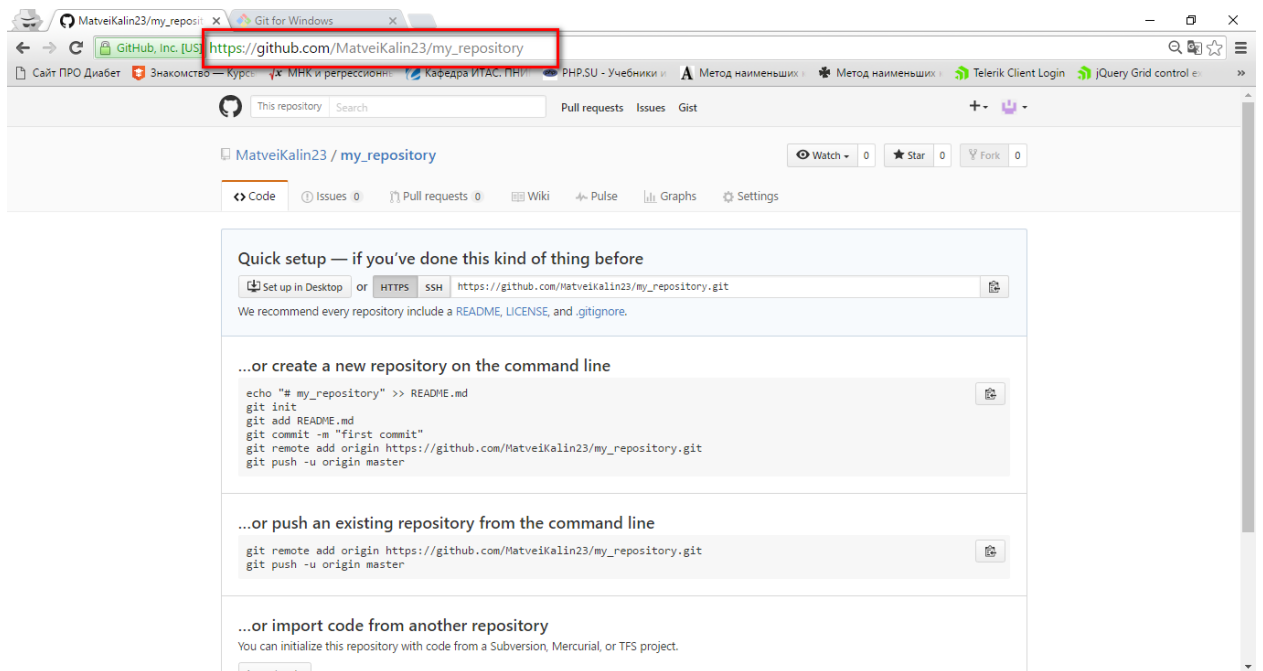
```
git remote add origin https://github.com/MatveiKalin23/my_repository.git
git push -u origin master
```

...or import code from another repository

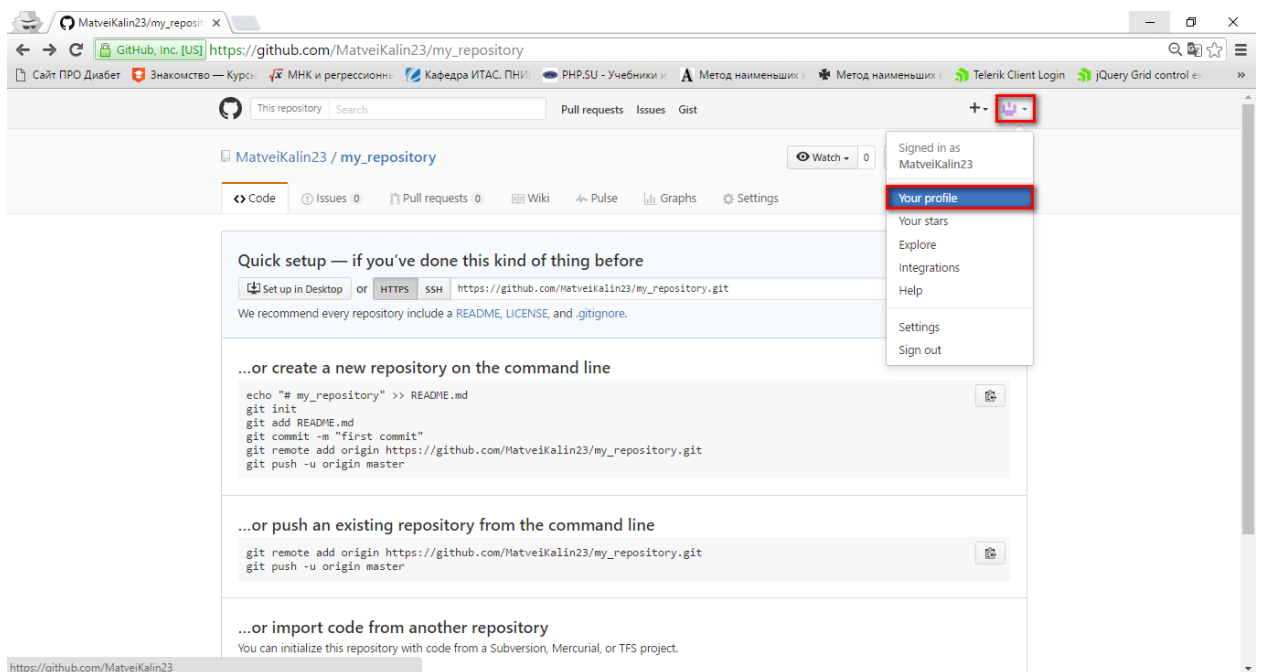
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

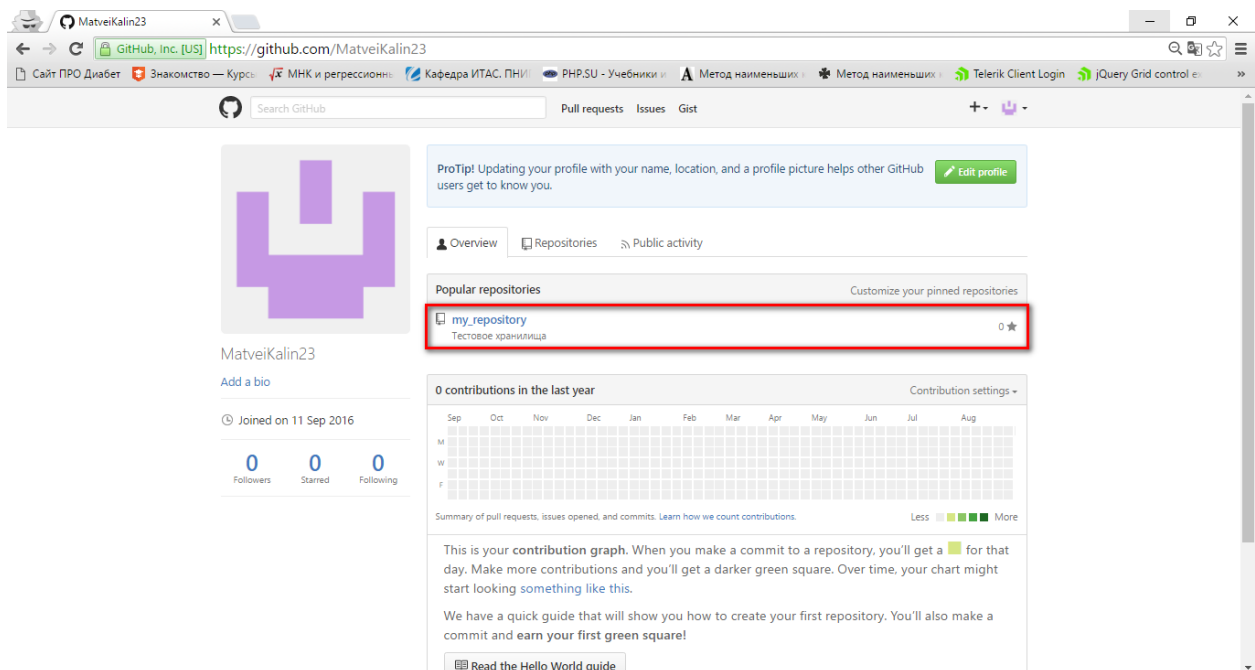
Import code

Адрес, выделенный красной рамкой мы должны отправить преподавателю. В Вашем случае адрес будет другой (Отправляет 1 человек, не вся подгруппа).



Чтобы просмотреть какие еще есть репозитории нужно перейти в мой профиль. В данном случае имеется всего один репозиторий.

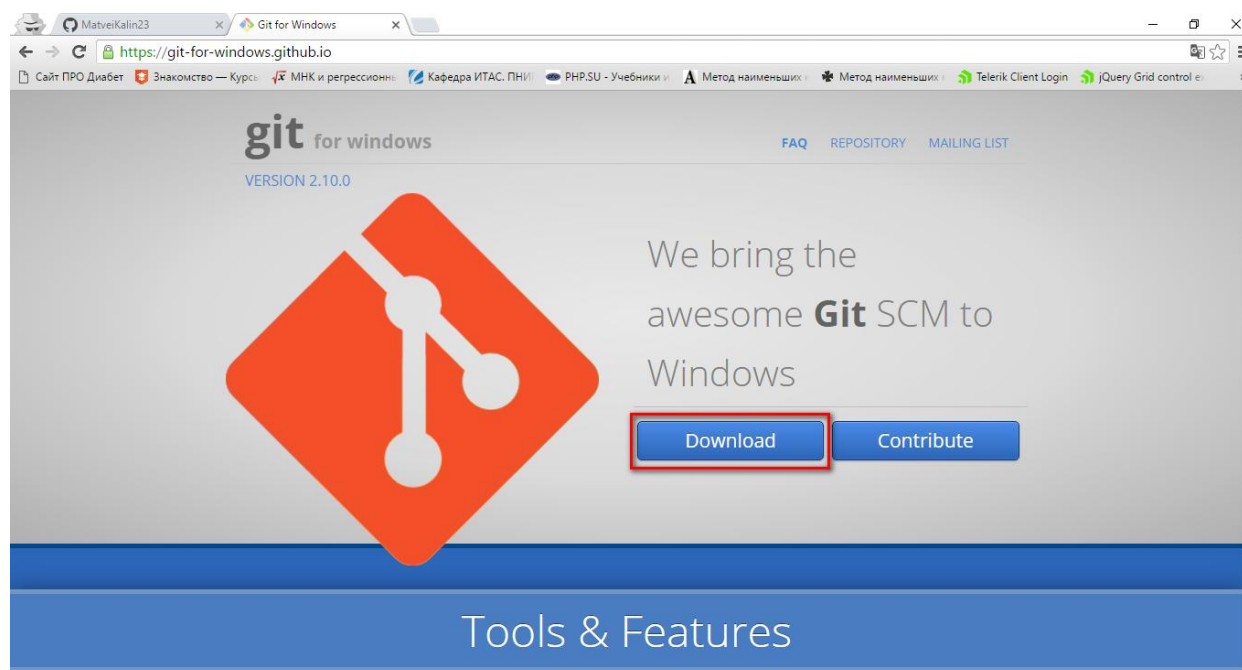




3). Как добавлять файлы на GitHub.

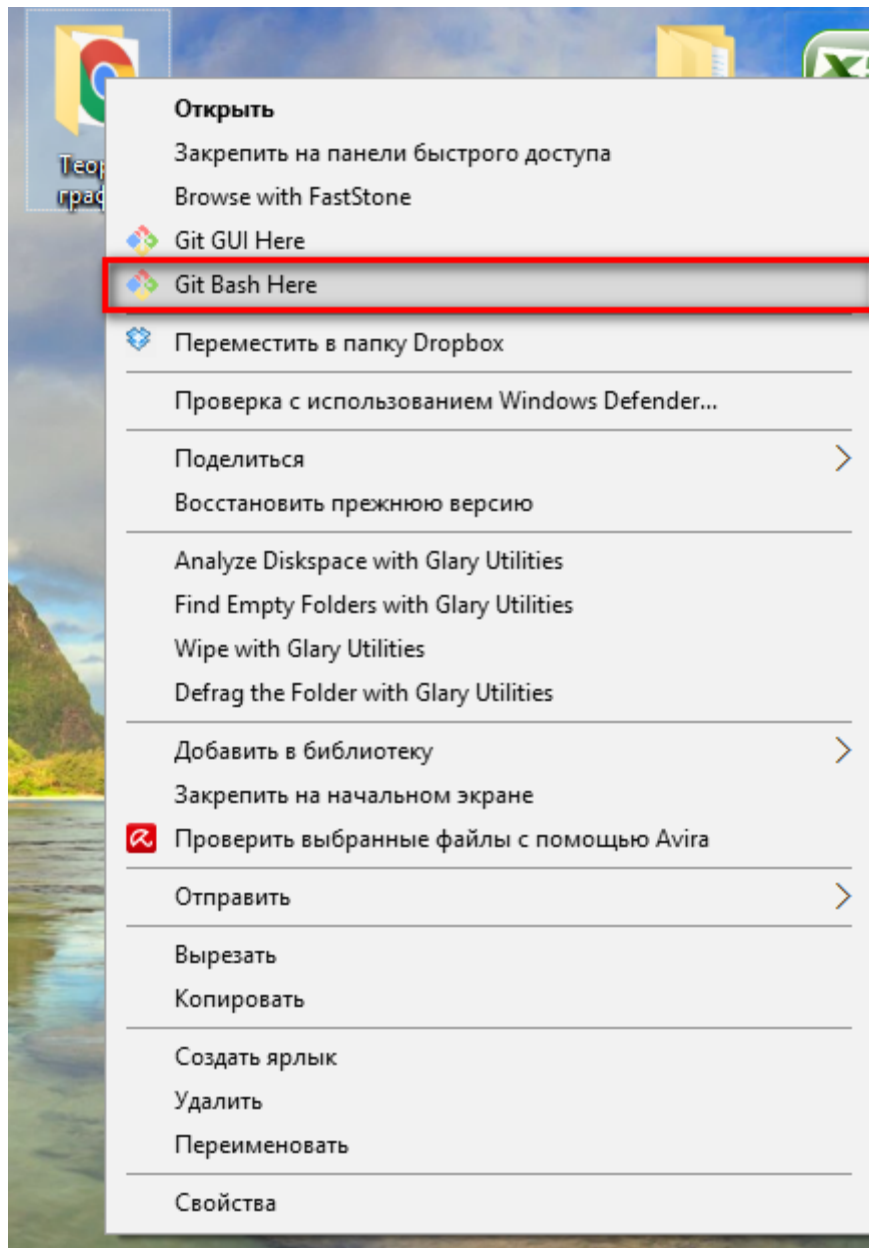
Этот пункт для всех участников подгрупп.

- Необходимо установить командную строку «Git» (для Windows), для этого нужно скачать установочный файл с сайта: <https://git-for-windows.github.io/>

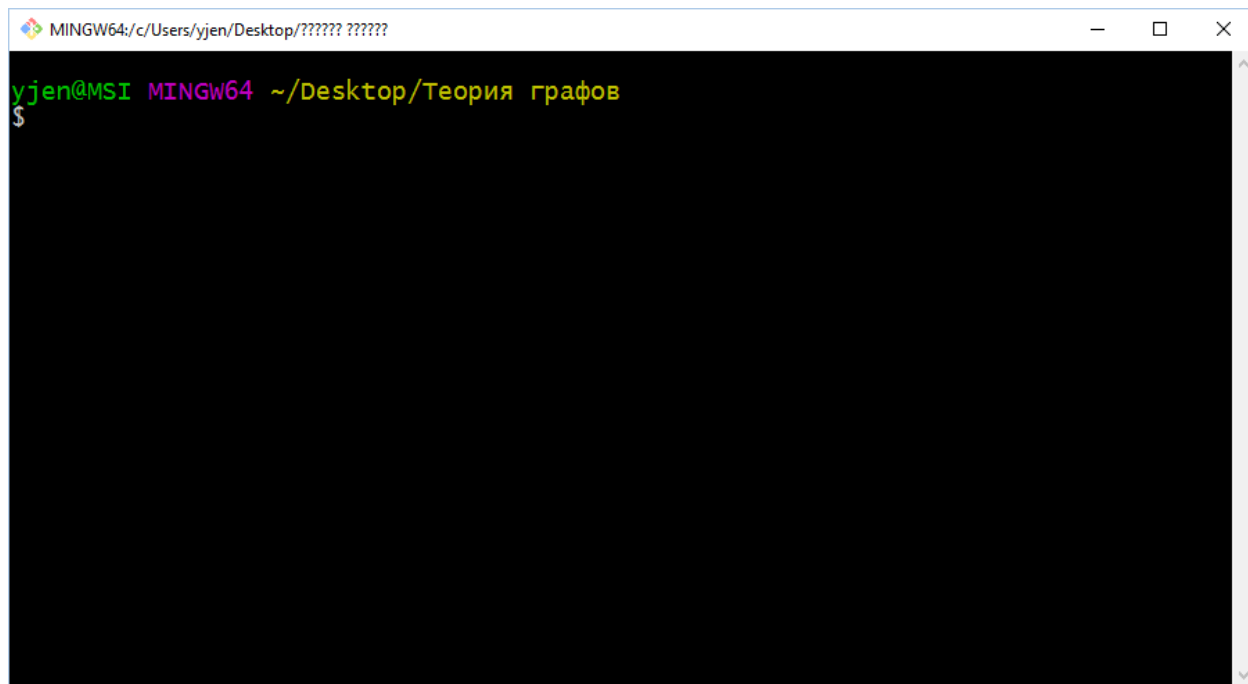


- Предположим, что у Вас на рабочем столе имеется папка с проектом, ее необходимо отправить в удаленный репозиторий.

- Чтобы отправить данные в удаленный репозиторий нужно щелкнуть правой клавишей мыши на папке с проектом и выбрать пункт «Git Bash Here».



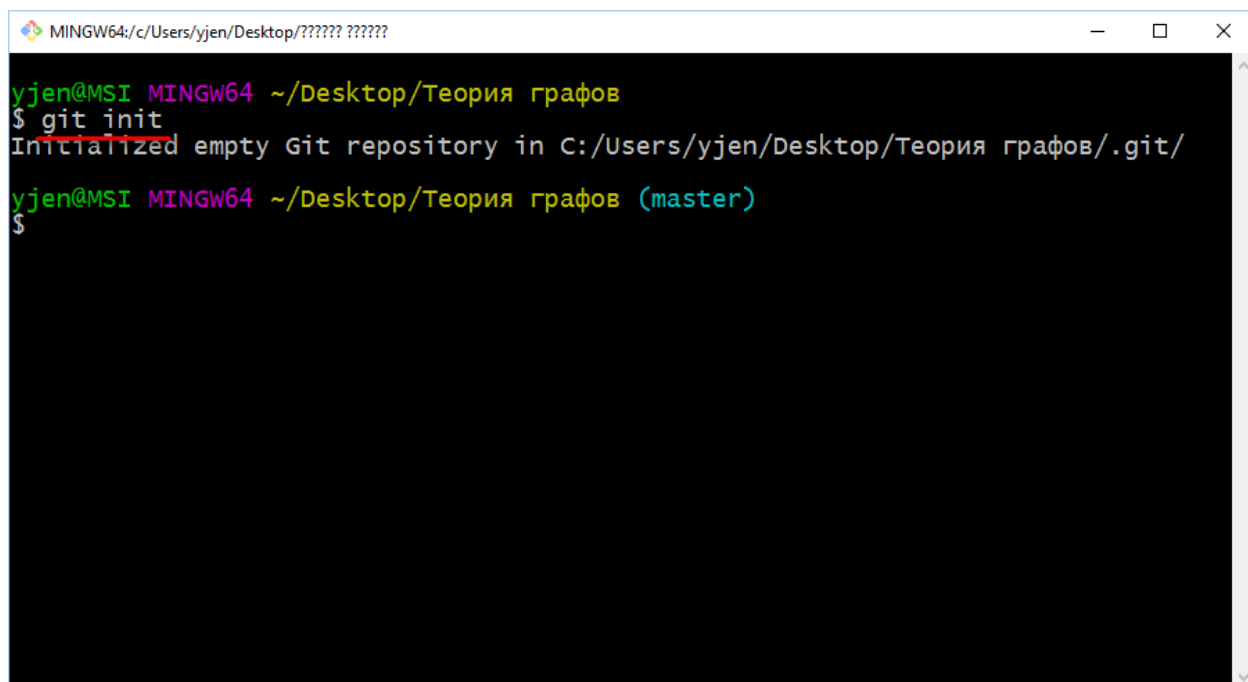
- После чего откроется консоль «GitBash».



```
MINGW64: c:/Users/yjen/Desktop/????? ????
yjen@MSI MINGW64 ~/Desktop/Теория графов
$
```

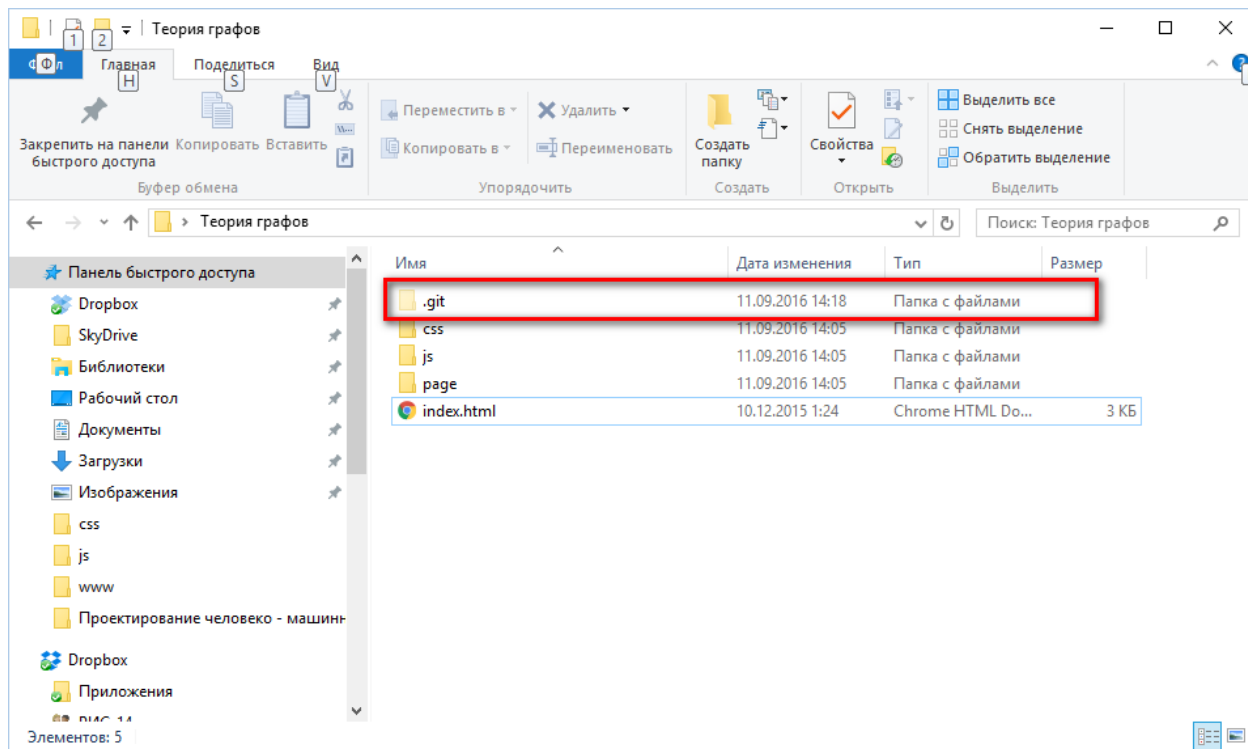
- В этой консоли необходимо ввести команду, эта команда создает основу git репозитория:

`git init`



```
MINGW64: c:/Users/yjen/Desktop/????? ????
yjen@MSI MINGW64 ~/Desktop/Теория графов
$ git init
Initialized empty Git repository in C:/Users/yjen/Desktop/Теория графов/.git/
yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$
```

- И если Вы перейдете в папку проекта, а также, если у Вас показываются скрытые папки, то можно увидеть папку «.git», содержащую все необходимые файлы репозитория.



- После чего необходимо добавить файлы под версионный контроль командой в консоль:

`git add .`

```
MINGW64: c:/Users/yjen/Desktop/????? ????
yjen@MSI MINGW64 ~/Desktop/Теория графов
$ git init
Initialized empty Git repository in C:/Users/yjen/Desktop/Теория графов/.git/
yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git add .
warning: LF will be replaced by CRLF in js/jquery-1.11.2.min.js.
The file will have its original line endings in your working directory.
yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ |
```

- Чтобы просмотреть какие файлы готовы к коммиту (к созданию версии) необходимо воспользоваться командой:

`git status`

```
MINGW64/c:/Users/yjen/Desktop/????? ????
$ git init
Initialized empty Git repository in C:/Users/yjen/Desktop/Теория графов/.git/
yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git add .
warning: LF will be replaced by CRLF in js/jquery-1.11.2.min.js.
The file will have its original line endings in your working directory.
yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   css/style.css
        new file:   index.html
        new file:   js/bfs.js
        new file:   js/clickesGraph.js
        new file:   js/coreGraph.js
        new file:   js/countNumberRoutes.js
        new file:   js/dfs.js
```

Зеленым цветом показаны файлы, которые готовы к коммиту.

Красным цветом показаны файлы, которые были изменены и не отслеживаются версионным контролем.

В данном случае все файлы готовы к коммиту (все зеленого цвета).

Если у Вас код изменился и появились файлы красным цветом, значит нужно снова воспользоваться командой:

```
git add .
```

- Чтобы отправить коммит в удаленный репозиторий (хранилище), нужно воспользоваться командой (название коммита может быть – Имя Фамилия и какой по счету коммит):

```
git commit -m "Название коммита"
```

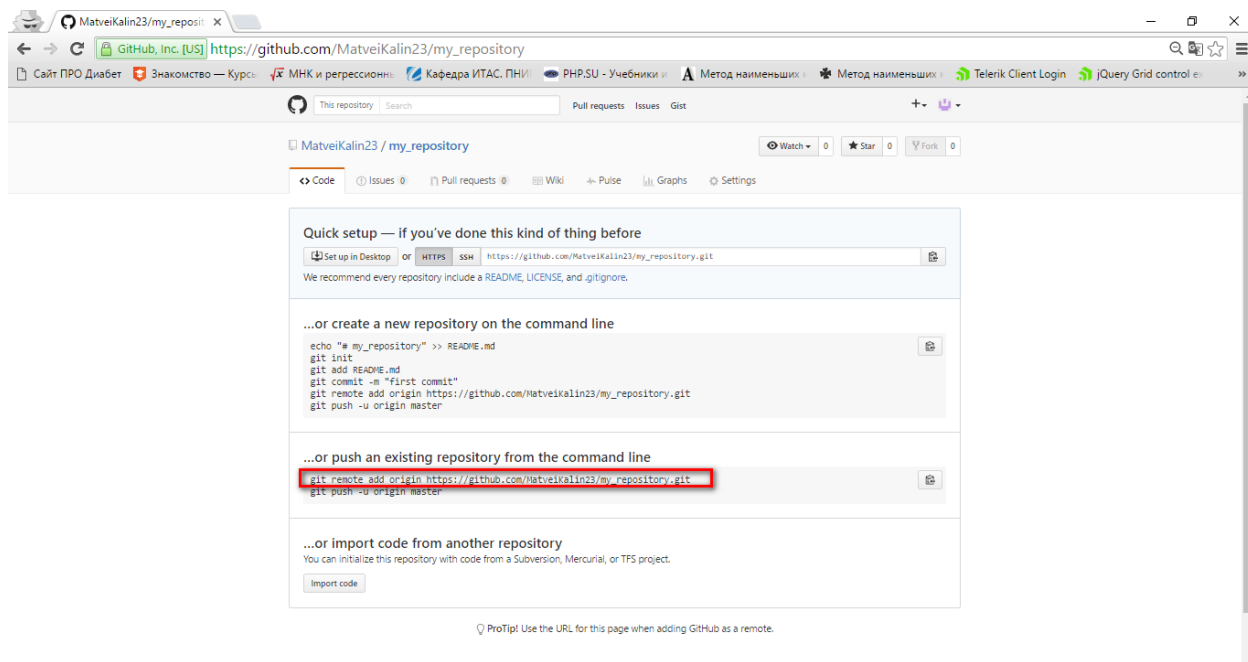
```
MINGW64/c/Users/yjen/Desktop/????? ????
new file:   page/dfs.html
new file:   page/dijkstra.html
new file:   page/floyd_Worshell.html
new file:   page/levelParallel.html
new file:   page/methodShimbell.html
new file:   page/minSpanningTree.html

yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git commit -m "Название коммита"
[master (root-commit) 894a396] Название коммита
24 files changed, 7496 insertions(+)
create mode 100644 css/style.css
create mode 100644 index.html
create mode 100644 js/bfs.js
create mode 100644 js/clickesGraph.js
create mode 100644 js/coreGraph.js
create mode 100644 js/countNumberRoutes.js
create mode 100644 js/dfs.js
create mode 100644 js/dijkstra.js
create mode 100644 js/floyd_Worshell.js
create mode 100644 js/jointFunction.js
create mode 100644 js/jquery-1.11.2.min.js
create mode 100644 js/levelParallel.js
```

- После чего переходим в наш репозиторий в браузере. Мой тестовый имеет адрес:

https://github.com/MatveiKalin23/my_repository

И внизу, с сайта, копируем строку и вставляем ее в консоль GitBash.

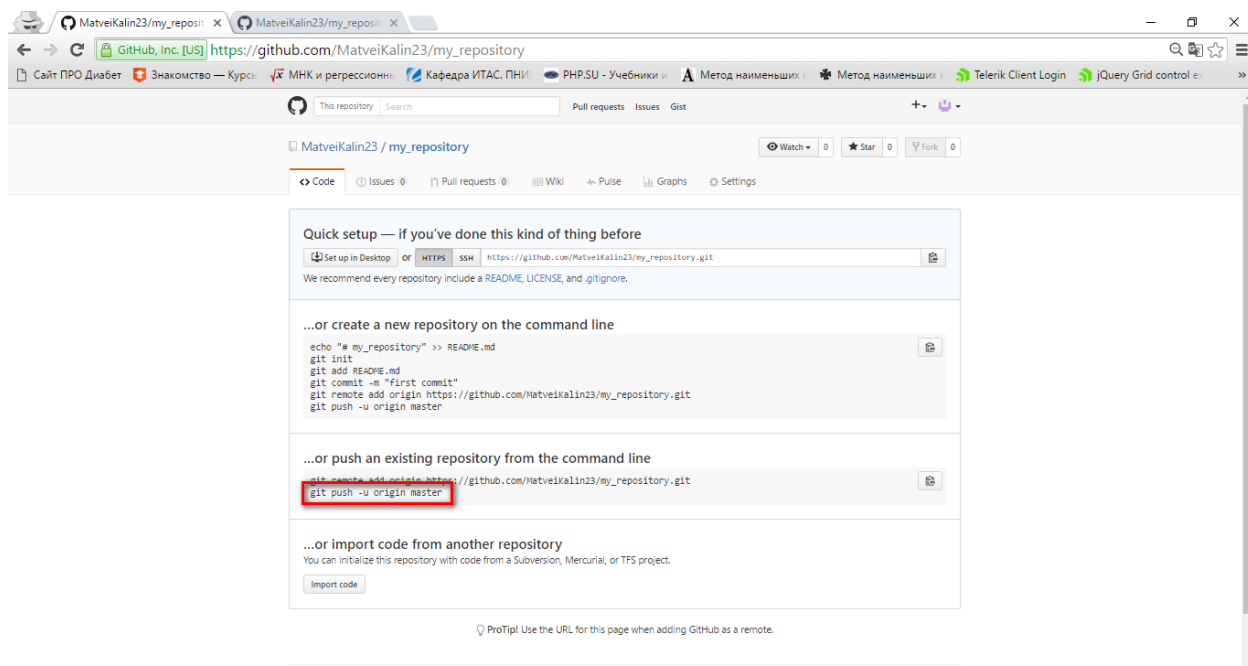


```
MINGW64/c/Users/yjen/Desktop/????? ????
create mode 100644 js/dfs.js
create mode 100644 js/dijkstra.js
create mode 100644 js/floyd_warshall.js
create mode 100644 js/jointFunction.js
create mode 100644 js/jquery-1.11.2.min.js
create mode 100644 js/levelParallel.js
create mode 100644 js/methodShimbell.js
create mode 100644 js/minSpanningTree.js
create mode 100644 page/bfs.html
create mode 100644 page/clicksGraph.html
create mode 100644 page/coreGraph.html
create mode 100644 page/countNumberRoutes.html
create mode 100644 page/dfs.html
create mode 100644 page/dijkstra.html
create mode 100644 page/floyd_warshall.html
create mode 100644 page/levelParallel.html
create mode 100644 page/methodShimbell.html
create mode 100644 page/minSpanningTree.html

yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git remote add origin https://github.com/MatveiKalin23/my_repository.git

yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$
```

- После чего копируем еще одну строку с сайта и вставляем ее в консоль GitBash:



```
MINGW64/c/Users/yjen/Desktop/????? ????
create mode 100644 js/dijkstra.js
create mode 100644 js/floyd_Uorshell.js
create mode 100644 js/jointFunction.js
create mode 100644 js/jquery-1.11.2.min.js
create mode 100644 js/levelParallel.js
create mode 100644 js/methodShimbell.js
create mode 100644 js/minSpanningTree.js
create mode 100644 page/bfs.html
create mode 100644 page/clickesGraph.html
create mode 100644 page/coreGraph.html
create mode 100644 page/countNumberRoutes.html
create mode 100644 page/dfs.html
create mode 100644 page/dijkstra.html
create mode 100644 page/floyd_Uorshell.html
create mode 100644 page/levelParallel.html
create mode 100644 page/methodShimbell.html
create mode 100644 page/minSpanningTree.html

yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git remote add origin https://github.com/MatveiKalin23/my_repository.git

yjen@MSI MINGW64 ~/Desktop/Теория графов (master)
$ git push -u origin master
```

- После этой команды нужно ввести логин и пароль того, кто создал репозиторий.
- После чего можно посмотреть на сайте GitHub. Свой коммит.
- Если у Вас что-то изменилось в проекте, то необходимо снова использовать команды:

```
git add .
git commit -m "Название коммита"
git push -u origin master
```

4). Как сделать откат коммитов.

Иногда требуется удалить коммит с локального и удаленного сервера.

- Для этого в консоли необходимо воспользоваться командой:

```
git log
```

```
MINGW64/c/Users/yjen/Desktop/Graph
yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git add .

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git commit -m "2 commit"
[master 673b0cf] 2 commit
1 file changed, 1 insertion(+)

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git log
commit 673b0cfdeab9f7f4e98d5a0704a06f/edfed50ab
Author: matvei <matthewk@mail.ru>
Date: Sun Sep 11 16:29:11 2016 +0500

    2 commit

commit 5478675c1134195c0ea84921b3169f6bd75bb08e
Author: matvei <matthewk@mail.ru>
Date: Sun Sep 11 16:17:07 2016 +0500

    1 commit

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$
```

Эта команда позволяет вывести историю коммитов. Самые новые находятся в верху. В данном случае находится всего 2-а коммита.

- К примеру, нам необходимо вернуться к первому коммиту (При этом все изменения удалятся, так что нужно быть осторожней).

Чтобы вернуться к первому коммиту, мы должны скопировать хэш-код первого коммита.

```

MINGW64:/c/Users/yjen/Desktop/Graph
yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git add .

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git commit -m "2 commit"
[master 673b0cf] 2 commit
1 file changed, 1 insertion(+)

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git log
commit 673b0cfdeab9f7f4e98d5a0704a06f7edfed50a6
Author: matvei <matthewk@mail.ru>
Date: Sun Sep 11 16:29:11 2016 +0500

    2 commit

commit 5478675c1134195c0ea84921b3169f6bd75bb08e
Author: matvei <matthewk@mail.ru>
Date: Sun Sep 11 16:17:07 2016 +0500

    1 commit

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$

```

И приписать его к команде:

```
git reset --hard 5478675c1134195c0ea84921b3169f6bd75bb08e
```

```

MINGW64:/c/Users/yjen/Desktop/Graph
Author: matvei <matthewk@mail.ru>
Date: Sun Sep 11 16:29:11 2016 +0500

    2 commit

commit 5478675c1134195c0ea84921b3169f6bd75bb08e
Author: matvei <matthewk@mail.ru>
Date: Sun Sep 11 16:17:07 2016 +0500

    1 commit

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git reset --hard 5478675c1134195c0ea84921b3169f6bd75bb08e
fatal: ambiguous argument '-hard': unknown revision or path not in the working
ree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git reset --hard 5478675c1134195c0ea84921b3169f6bd75bb08e
HEAD is now at 5478675 1 commit

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$

```

- Затем, чтобы синхронизировать локальный репозиторий с удаленным необходимо воспользоваться командой:

```
git push -f
```


- После этого весь проект вернется к состоянию, когда был первый коммит.
- Даже, если мы выведем историю коммитов, мы увидим, что второй коммит исчез.

```
MINGW64:/c/Users/yjen/Desktop/Graph
'git <command> [<revision>...] -- [<file>...]'

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git reset --hard 5478675c1134195c0ea84921b3169f6bd75bb08e
HEAD is now at 5478675 1 commit

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git push -f
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ git log
commit 5478675c1134195c0ea84921b3169f6bd75bb08e
Author: matvei <matthewk@mail.ru>
Date:   Sun Sep 11 16:17:07 2016 +0500

    1 commit

yjen@MSI MINGW64 ~/Desktop/Graph (master)
$ |
```