



# Лекция 0

Контролируем версии

## **Цель курса:**

Сделать из группы новичков, специалистов, способных решать задачи различной сложности.

## Как проходит обучение:

- **24 часа** аудиторных занятий. Из них **8 часов** уделено практике;
- Не меньше **20 часов** домашней работы;
- Около **15 часов** на подготовку выпускного проекта;
- Выборка из **200 тестов**, охватывающих все разобранные темы.



PHOTODIARY

## The perfect weekend getaway

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciant. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

LEAVE A COMMENT



## Team

et, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris



PHOTODIARY

## A day exploring the Alps

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris

Sign up for our newsletter!



## Получаемые навыки:

- Научиться использовать HTML;
- Научиться использовать CSS;
- Научиться пользоваться системой контроля версий (Git);
- Изучить принципы работы с Photoshop CC и Avocode;
- Научиться создавать сайты с фиксированной шириной из .PSD макета.

## Особенности этого курса

- Опаздывать не стоит;
- Сменка;
- Процесс работы с Git и BASH проще, чем в следующих курсах;
- CSS код будет проверяться автоматически;
- Структуру каталогов менять нельзя;
- Личные файлы помещаются в папку Personal;
- В субботу проводятся дополнительные занятия, на которые нужно предварительно записываться.

## **Цели этого занятия:**

- Познакомиться с самыми основными командами BASH;
- Изучить основы Git и Github;
- Настроить рабочее окружение.

### Основные требования:

- HTML5/CSS3, SASS/LESS, Twitter Bootstrap, создание адаптивных UI
- Работа с макетами (Photoshop, Illustrator, Sketch)
- Хорошее знание Javascript (ES5, ES6), TypeScript будет плюсом
- Опыт работы со следующими фреймворками: Angular, React, Ember, Backbone
- Владение одним или несколькими инструментами сборки, такими как WebPack, Gulp, Grunt
- Опыт разработки мобильных приложений с помощью React Native, Ionic или Xamarin будет плюсом
- AJAX, Websockets, понимание RESTful
- Git/SVN
- Понимание основ SEO
- Базовые знания node.js или других backend-технологий будут плюсом, так же как и опыт работы с СУБД
- Хорошее знание английского



# Глоссарий

**HTML-файл** – документ, в котором содержится **контент** – текст, формы, изображения, пути загрузки скриптов и стилей.

**CSS-файл** – документ, в котором указаны **параметры отображения** элементов (цвет, размер, положение).

**JavaScript** – **язык программирования сценариев**, сейчас широко используется везде, в том числе и в браузере.

**SEO** – комплекс оптимизационных мер для поднятия позиций сайта в результатах выдачи поисковых систем.

# Основы работы с BASH



## **BASH –**

одна из наиболее популярных современных разновидностей командной оболочки UNIX. Особенно популярна в среде Linux, где она часто используется в качестве предустановленной командной оболочки.

## Немного обобщим

- **pwd** – возвращает путь к рабочей директории;
- **cd (change directory)** – путешествуем по директориям;
- **history** – показывает историю команд за все время;
- **help** – показывает версию терминала и доступные команды;
- **Tab** – подсказывает путь/имя файла/команду;
- **Стрелка вверх** – показывает предыдущие команды;
- **Ctrl+C** – прекратить выполнение команды;
- **ESC + :q** – выйти из редактора Vim.

**Git**

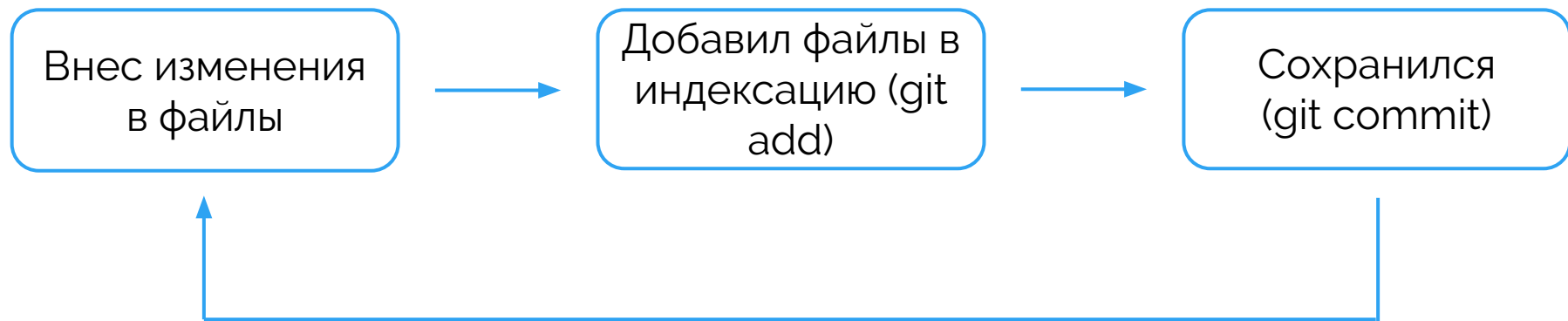


## **Git –**

распределенная система контроля версий, которая рассчитана на командную работу над проектом.

**Как работает Git?**

## Схема работы с git





**Начало работы**

Перед началом работы, необходимо указать в конфиге имя и email, которые потом будут указываться в каждом коммите. Делается это с помощью команд:

```
git config --global user.name "Your Name"
```

```
git config --global user.email "your_email@example.com"
```

# Локальный репозиторий

knyazevich@Luna: ~/my-project

```
knyazevich@Luna:~$ mkdir my-project ; cd my-project ; touch style.css index.html
```

```
knyazevich@Luna:~/my-project$ git init
```

```
Initialized empty Git repository in /home/knyazevich/my-project/.git/
```

```
knyazevich@Luna:~/my-project$ git add .
```

```
knyazevich@Luna:~/my-project$ git status
```

```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   index.html
```

```
new file:   style.css
```

```
knyazevich@Luna:~/my-project$ git commit -m 'Initial commit'
```

```
[master (root-commit) 39e3037] Initial commit
```

```
2 files changed, 24 insertions(+)
```

```
create mode 100644 index.html
```

```
create mode 100644 style.css
```

```
knyazevich@Luna:~/my-project$
```

knyazevich@Luna: ~/my-project

```
knyazevich@Luna:~/my-project$ git log --oneline
```

```
4407b5f Header logo -- bug fixed
```

```
c365da9 Logo in header w/ link added
```

```
fca3530 Article wrapper added
```

```
1ad5c55 Footer added
```

```
29eb1b5 Header added
```

```
39e3037 Initial commit
```

```
knyazevich@Luna:~/my-project$
```

## **.gitignore –**

позволяет указать файлы или папки, которые не нужно индексировать. То есть, эти файлы никогда не улетят на GitHub.

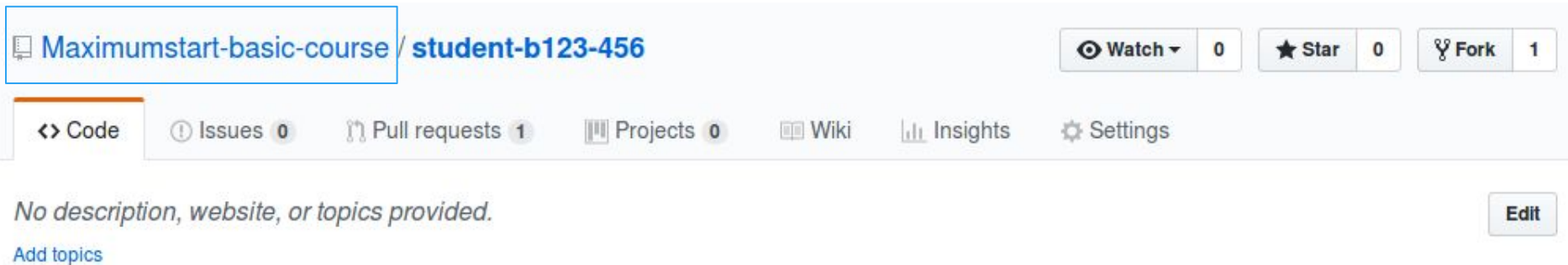
## Какие команды мы уже знаем?

- **git init** – инициализирует пустой репозиторий;
- **git add <file> / <.>** – добавляет файл / все файлы;
- **git commit -m 'Message'** – создает коммит;
- **git log --oneline (-1 --all --graph)** – выводит историю коммитов;
- **git remote add <name> <path>** – добавляет внешний путь;
- **git push** – отправляет коммиты на сервер;
- **git pull** – загружает коммиты с сервера;
- **git config ...** – вносит изменения в конфигурационный файл.

**Как мы будем работать?**



# Основной репозиторий



The screenshot shows the GitHub interface for a repository named 'Maximumstart-basic-course' owned by 'student-b123-456'. The repository name is highlighted with a blue box. The interface includes navigation tabs for 'Code', 'Issues' (0), 'Pull requests' (1), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. On the right, there are buttons for 'Watch' (0), 'Star' (0), and 'Fork' (1). Below the navigation tabs, a message states 'No description, website, or topics provided.' with an 'Edit' button. A link to 'Add topics' is also visible.

Maximumstart-basic-course / student-b123-456

Watch 0 Star 0 Fork 1

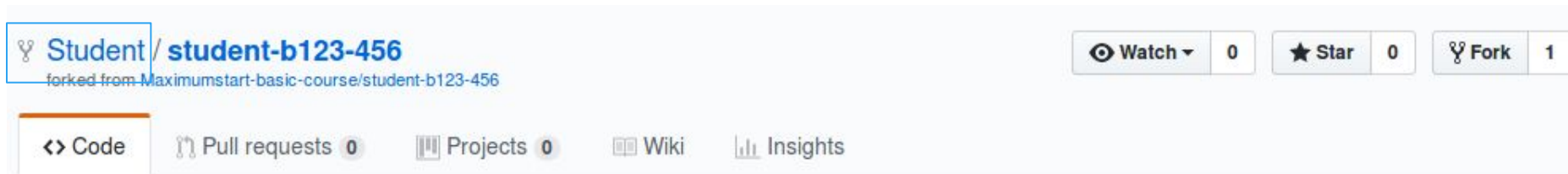
Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

[Add topics](#)

это основной репозиторий, в котором будет находиться чистый код, в который **вы не можете** пушить свои коммиты непосредственно. Изменения в него попадают через **pull request**.

## Личный репозиторий (форк)



*No description, website, or topics provided.*

это ваш личный репозиторий, которые является форком основного и в который вы **можете** пушить. Именно его вы клонируете на компьютер и в созданной папке будете работать.

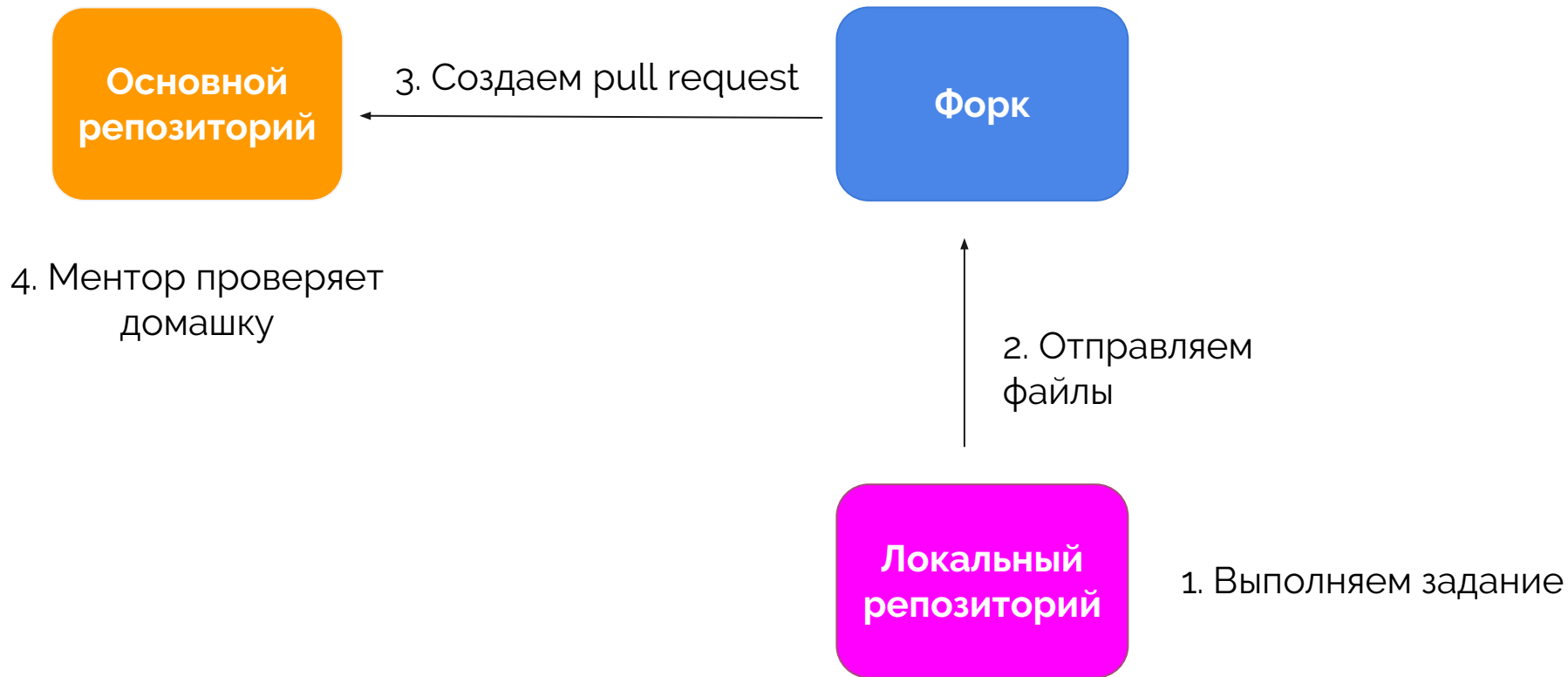
## Локальный репозиторий (клонированный)



student-b123-456

папка у вас на компьютерах – это клон личного репозитория, в нем вы будете **вести всю непосредственную работу**, которая затем будет отправляться в личный репозиторий.

# Как сдается домашка?



## Как создать pull request?

после того, как домашнее задание готово – необходимо создать **pull request**:



## Отправка задания на проверку

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



base fork: Maximumstart-basic-course/di... ▾

base: master ▾



head fork: DonProton/dima-e1-002 ▾

compare: first-task ▾

✓ **Able to merge.** These branches can be automatically merged.



**Create pull request**

Discuss and review the changes in this comparison with others.



Проверку проходят исключительно проекты, которые соответствуют всем базовым требованиям и сданы в срок. Каждый студент имеет 4 (четыре) попытки сдачи проекта и не более одной в день, поэтому рекомендуем грамотно расходовать свое время и не оставлять все на последний день

## Разобранные требования

- Коммиты должны иметь осмысленные названия, которые отражают общее содержание изменений. Чистоте истории коммитов уделяется особое внимание, чистые коммиты важны не меньше, чем чистый код.
- Вёрстка идентично отображается в последних версиях браузеров Chrome, Opera, Edge и Firefox.
- JavaScript\jQuery не используется.