

ОСНОВЫ Git

ОСНОВЫ Git

Основы Git

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](https://itvdn.com)

Доступ можно получить через руководство вашего учебного центра



Проверьте как Вы усвоили данный материал на [TestProvider.com](https://testprovider.com)

Что такое Git?

Основы Git

Определение

Git – это бесплатная и открытая система контроля версий (SCM). Он создан для отслеживания изменений кода в ваших проектах. Это предоставляет вам контроль на каждом этапе развития вашего приложения.



ОСНОВЫ Git

Установка

Чтобы начать работу с Git, его нужно установить. Скачайте установщик с официального сайта, запустите его и следуйте его указаниям. После этого в меню Пуск появятся две новые программы: Git Bash и Git GUI.



Основы Git

Начало работы

Выполните следующие команды, чтобы Git узнал Ваше имя и электронную почту (измените на свои данные):

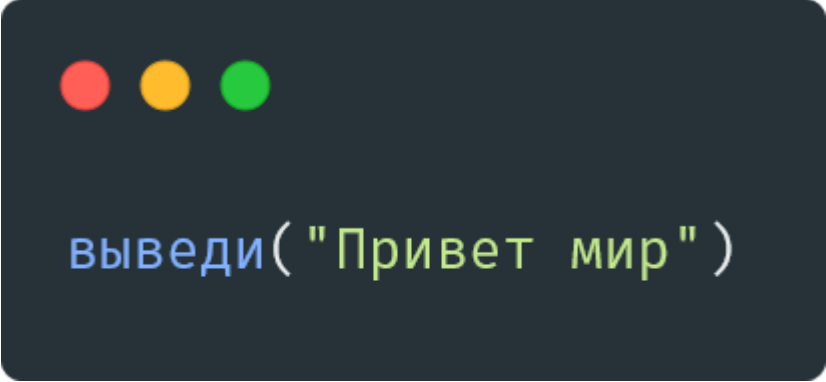
```
git config --global user.name "Leonid Podriz"  
git config --global user.email "leonidpodriz@gmail.com"
```

Создание пустого проекта

ОСНОВЫ Git

Подготовка рабочей папки

Начните работу с создания пустого каталога с именем **"code"**, затем войдите в него и создайте там файл с именем **main.code** с таким псевдокодом:



```
выведи("Привет мир")
```


Основы Git

Создание пустого репозитория


Теперь у вас есть каталог с одним файлом.
Чтобы создать git репозиторий из этого каталога, выполните команду:



ОСНОВЫ Git

Добавление файла в репозиторий

Создание репозитория не значит, что Git сразу же начал следить за всеми файлами в папке. Файлы необходимо добавить вручную. Это дает контроль на файловой структурой Git репозитория и гарантирует, что никакие конфиденциальные данные не попадут в открытый доступ без вашего согласия.



```
git add main.code  
git commit -m "Add my fisrt code file"
```

ОСНОВЫ Git

Проверка состояния репозитория

Команда **git status** возвращает информацию о репозитории:

- сколько файлов изменено с последнего коммита
- сколько файлов было добавлена файл, но не добавлено в репозиторий
- Какие файлы были удалены
- И так дальше.



```
$ git status
# On branch master
nothing to commit (working directory clean)
```

Оснoвы Git

Изменения

Изменим код в `main.code`:

```
выведи("Привет, Мир")
```

В код добавлена запятая и "Мир" написано с большой буквы. Как на это отреагирует Git?

```
> git status
On branch master
Changes not staged for commit:
  (use "git add <file> ..." to update what will be committed)
  (use "git restore <file> ..." to discard changes in working directory)
        modified:   main.code


no changes added to commit (use "git add" and/or "git commit -a")
```

Обработка изменений

Git

Добавьте изменения

Выполните команду git проиндексировать изменения. Проверьте состояние:



```
> git add main.code
> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file> ..." to unstage)
       modified:   main.code
```

ОСНОВЫ Git

КОММИТ

Предположим, что вы отредактировали три файла (**main.code**, **module.code**, и **math.code**). Теперь вы хотите «закоммитить» все изменения, при этом чтобы изменения в **main.code** и **module.code** были одним коммитом, в то время как изменения в **math.code** логически не связаны с первыми двумя файлами и должны идти отдельным коммитом.

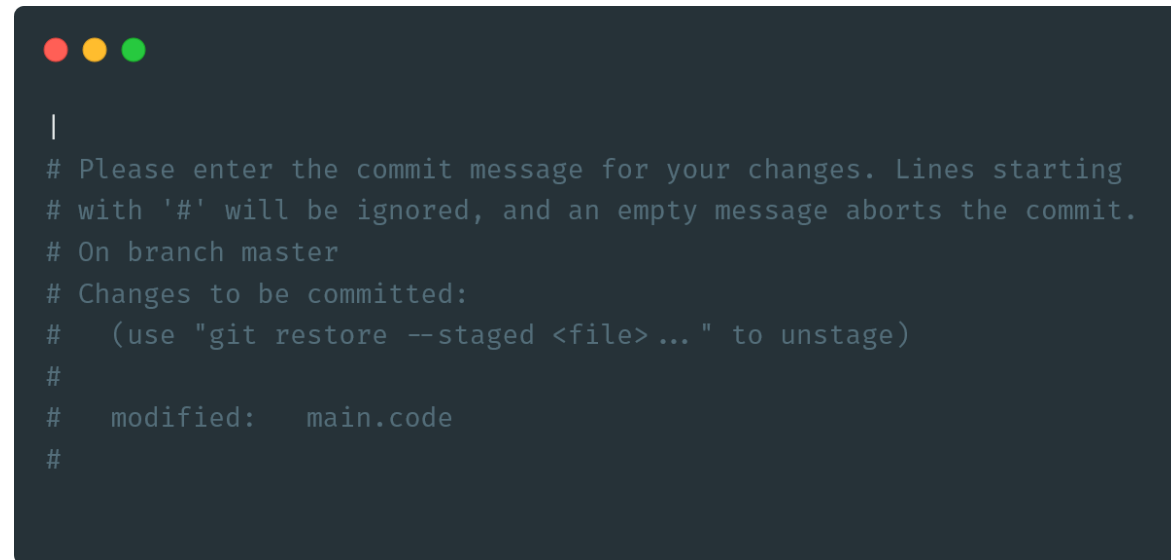
```
git add main.code
git add module.code
git commit -m "Changes for main and module"
```

```
git add math.code
git commit -m "Unrelated change to math"
```

ОСНОВЫ Git

Коммитим изменения

Команда **git commit** позволит вам интерактивно редактировать комментарии для коммита. Если вы опустите метку *-m* из командной строки, git перенесет вас в редактор по вашему выбору.

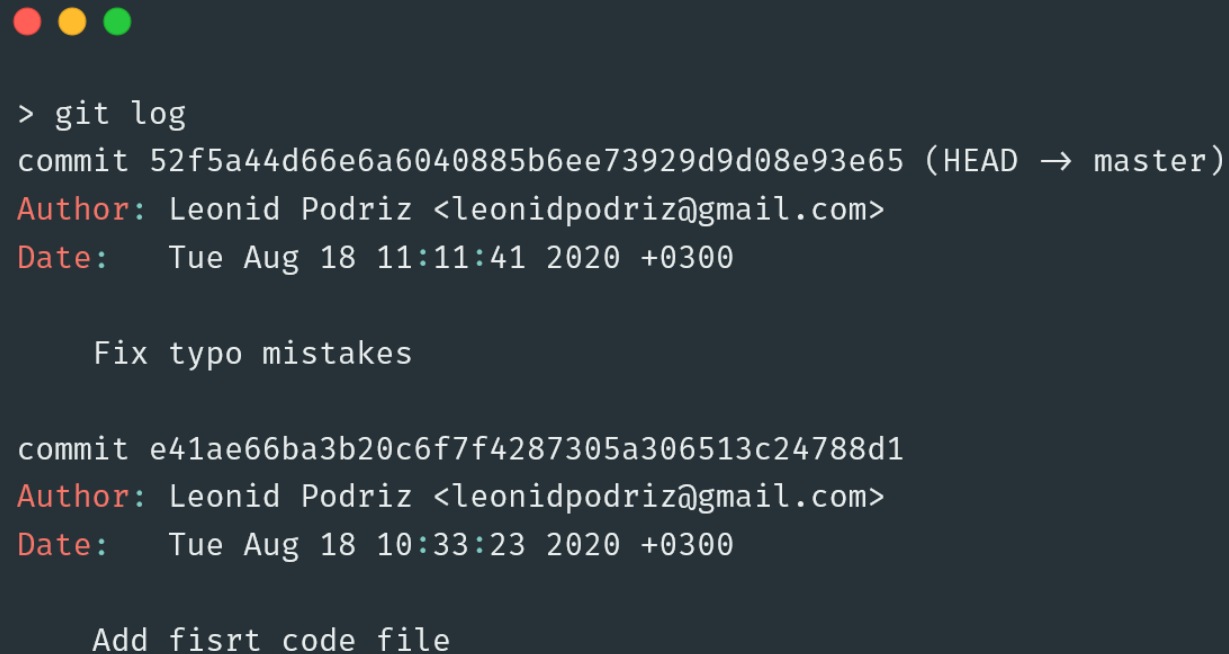
A dark-themed terminal window with three colored window control buttons (red, yellow, green) at the top. It displays the interactive commit message editor with the following text:

```
|
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git restore --staged <file> ..." to unstage)
#
#   modified:   main.code
#
```


ОСНОВЫ Git

История

Чтобы получить историю комитов в репозитории используйте команду: `git log`:



```
> git log
commit 52f5a44d66e6a6040885b6ee73929d9d08e93e65 (HEAD → master)
Author: Leonid Podriz <leonidpodriz@gmail.com>
Date: Tue Aug 18 11:11:41 2020 +0300

    Fix typo mistakes

commit e41ae66ba3b20c6f7f4287305a306513c24788d1
Author: Leonid Podriz <leonidpodriz@gmail.com>
Date: Tue Aug 18 10:33:23 2020 +0300

    Add fisrt code file
```

ОСНОВЫ Git

Что дальше?

Git имеет много функций. Чтобы изучить все, понадобится много времени. Мы знакомили вас с самым **БАЗОВЫМ** функционалом Git.

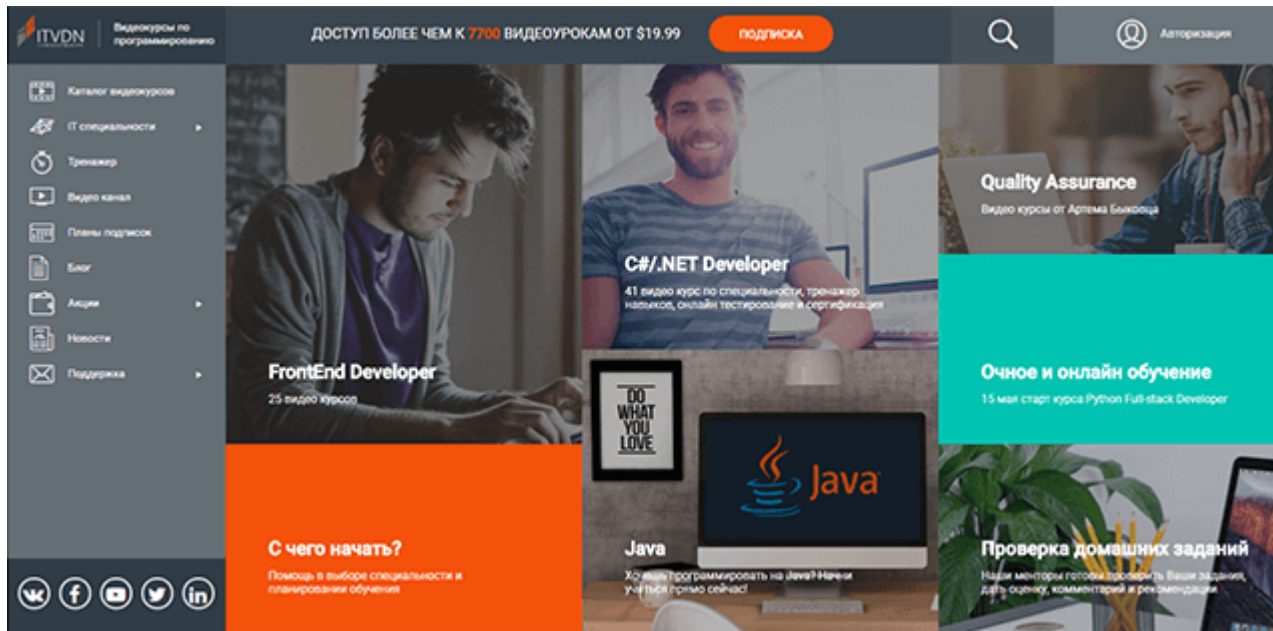
Некоторую часть функционала вам, возможно, никогда не придется использовать. Что-то, напротив, вы будете использовать каждый день.

Если вам понравилось работать с Git и вы желаете освоить его лучше, рекомендую изучить работу с ветками. Так же обратите внимание на рекомендации, где вы найдете материал для закрепления и новые темы, что могут вам пригодиться и облегчить работу.

Практическая часть

Смотрите наши уроки в видео формате

ITVDN.com



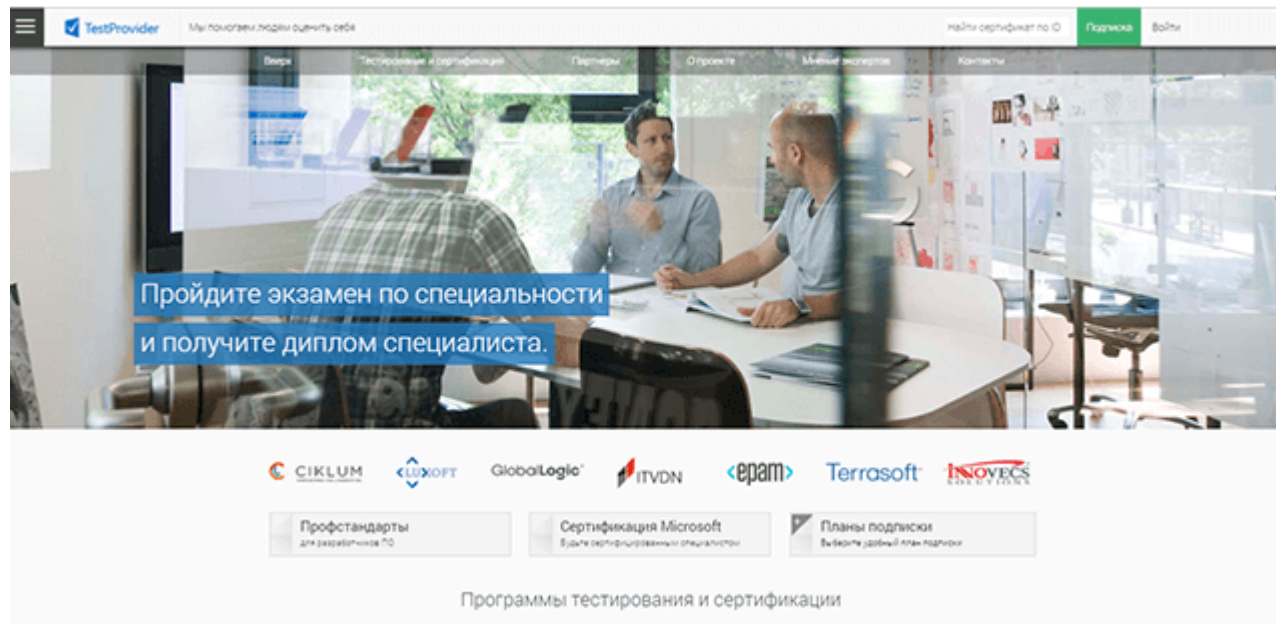
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](https://itvdn.com) для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

