

GIT

Урок 2

Введение в GIT

НЕМНОГО ПОВТОРЕНИЯ

Создадим папку, репозиторий и файл index.html

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (master)
$ mkdir dev

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (master)
$ cd dev

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа/dev (master)
$ git init
Initialized empty Git repository in C:/Users/Regina/Desktop/4 группа/dev/.git/

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа/dev (master)
$ touch index.html
```

Проиндексируем, сделаем коммит и посмотрим историю



Search or jump to...

7

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



Set your status

ReginaZakirova

🕒 Joined 20 minutes ago

Edit

ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

[Edit profile](#)



[Overview](#) [Repositories 1](#) [Projects 0](#) [Stars 0](#) [Followers 0](#) [Following 0](#)

Find a repository...

Type: All ▾

Language: All ▾

[New](#)

newgit

★ Star

Updated 7 minutes ago



РАБОТА С SSH-КЛЮЧЕМ

Создаем SSH-ключ

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа/dev (master)
$ ssh-keygen
```

Скопируем сгенерированный ключ

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа/dev (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDdc+qV5QMUGypre93dEOjQCdCafJFz5chrvr7gN7bn
ZFuDvbCWeS0G2ZhwG5+kUHE4cG1yp713AkghDfpgWPLbcJwLGfxOLi0INpU1IrRGizkVMSDXyadiPGaa
re9SBT3Lv0601wgouhyMSRsVXHRiPoFo0nOi8/QTpGNTyi1hwYI6AQ/t88Nr59A9UcoCKT+jcUTsSSI+
fRdfx3eQ2iGnjykZxTjA2mHXmVzFshvn7QgpXKXHuVYgUpcdHJkKIXT5e9tnbtw/o637agfnwgnSJa1Y
MwqVtU+Qsdy1HfXaaqDELker7fWpi+Pu5Jv+3l9KbYxNwZGghpfsX8h/apn Regina@LAPTOP-JM4R8
740
```

ТЕКСТОВЫЙ РЕДАКТОР

Откроем страницу index.html при помощи редактора vi.

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа/dev (main)  
$ vi index.html
```

Для перехода в режим ввода текста нажимаем клавишу **i**.

Для перехода в командный режим нажимаем **Esc**

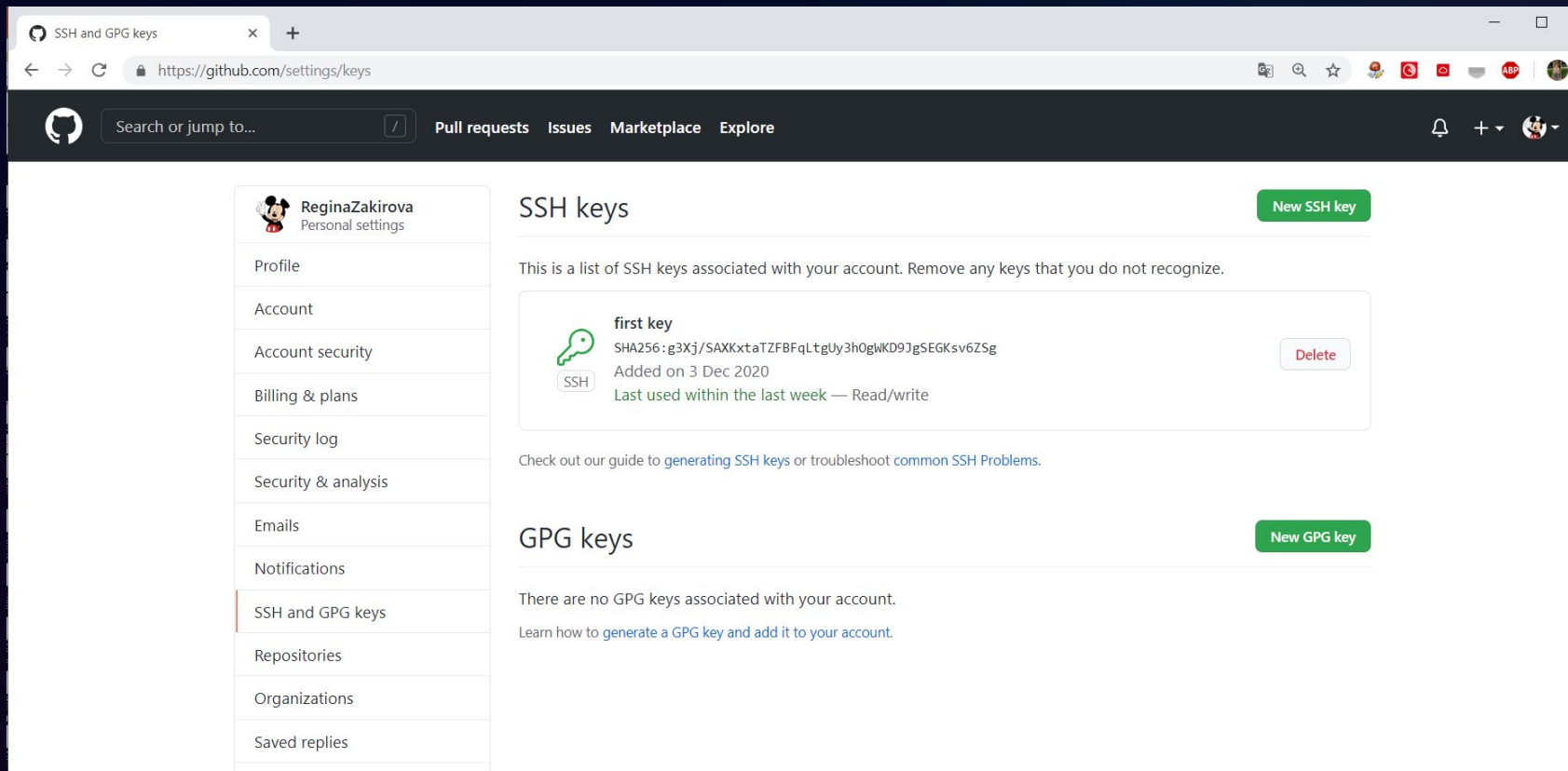
Для выхода из редактора необходимо нажать:

:q <Entert> или **Esc+Shift+Z+Shift+Z**

Для выхода без сохранения нажимаем **!:q**

Для выхода с сохранением нажимаем **:wq**

РАБОТА С SSH-КЛЮЧЕМ



The screenshot shows the GitHub 'SSH and GPG keys' settings page for user ReginaZakirova. The page is divided into two main sections: SSH keys and GPG keys. The SSH keys section contains one key named 'first key' with a SHA256 fingerprint and a 'Delete' button. The GPG keys section is currently empty.

SSH and GPG keys

https://github.com/settings/keys

Search or jump to... Pull requests Issues Marketplace Explore

ReginaZakirova
Personal settings

- Profile
- Account
- Account security
- Billing & plans
- Security log
- Security & analysis
- Emails
- Notifications
- SSH and GPG keys
- Repositories
- Organizations
- Saved replies

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Key Name	Fingerprint	Added	Last Used	Permissions	Action
first key	SHA256:g3Xj/SAXKxtaTZFBFqLtGy3h0gWkD9JgSEGKsv6ZSg	Added on 3 Dec 2020	Last used within the last week	Read/write	Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

ОТПРАВЛЯЕМ РЕПОЗИТОРИЙ НА GITHUB



Read the guide

ReginaZakirova / dev

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

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

Set up in Desktop or **HTTPS** SSH git@github.com:ReginaZakirova/dev.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a README, LICENSE, and .gitignore.

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

```
echo "# dev" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:ReginaZakirova/dev.git
git push -u origin main
```

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

```
git remote add origin git@github.com:ReginaZakirova/dev.git
git branch -M main
git push -u origin main
```

PUSH

Для того, чтобы отправить работу на удаленный сервер, существует команда :

```
git push [удал. сервер] [ветка]
```

Чтобы отправить ветку master на сервер origin (имена выставляются автоматически), необходимо выполнить следующую команду

```
$ git push origin master
```


РАБОТА С УДАЛЕННЫМИ СЕРВЕРАМИ

1. Подключились к удалённому репозиторию.
2. Переименуем ветку.
3. Отправляем данные.

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 грынна (master)
$ git remote add origin https://github.com/ReginaZakirova/hello.git

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 грынна (master)
$ git branch -M main

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 грынна (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ReginaZakirova/hello.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```



РАБОТА С УДАЛЕННЫМИ СЕРВЕРАМИ

Чтобы просмотреть, какие удалённые серверы у вас уже настроены, следует выполнить команду `git remote`. Если вы клонировали репозиторий, то увидите как минимум `origin` — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование:

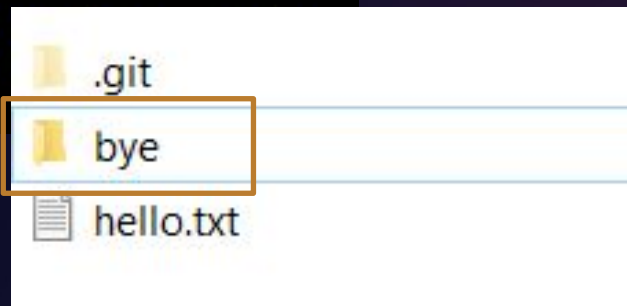
```
Regina@LAPTOP-JM4R874O MINGW64 ~/Desktop/4 группа (main)
$ git remote
origin

Regina@LAPTOP-JM4R874O MINGW64 ~/Desktop/4 группа (main)
$ git remote -v
origin https://github.com/ReginaZakirova/hello.git (fetch)
origin https://github.com/ReginaZakirova/hello.git (push)
```

КЛОНИРОВАНИЕ

Для того, чтобы клонировать репозиторий на локальную машину необходимо:

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (main)
$ git clone https://github.com/ReginaZakirova/bye.git
Cloning into 'bye'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0),
Unpacking objects: 100% (3/3), done.
```



FETCH

```
$ git fetch [имя удал. сервера]
```

Команда предназначена для получения данных из удалённых проектов.

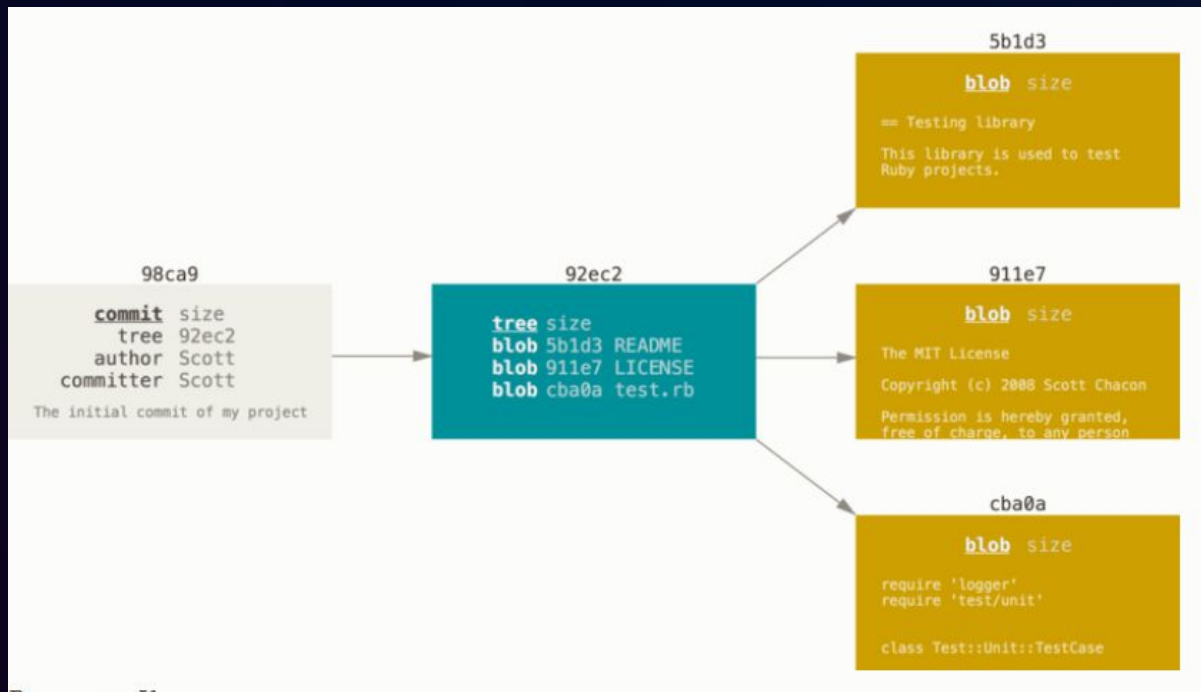
Команда `fetch` забирает данные в локальный репозиторий, но не сливает их с какими-либо другими работами и не модифицирует работу в данный момент. Необходимо вручную слить эти данные с пользовательскими проектами.

PULL

```
$ git pull
```

Команда автоматически извлекает и затем сливает данные из удалённой ветки в пользовательскую текущую ветку. Выполнение `git pull` извлекает данные с сервера, с которого изначально клонировали, и автоматически пытается слить (`merge`) их с кодом, над которым происходит в данный момент работа.

ВЕТВЛЕНИЕ



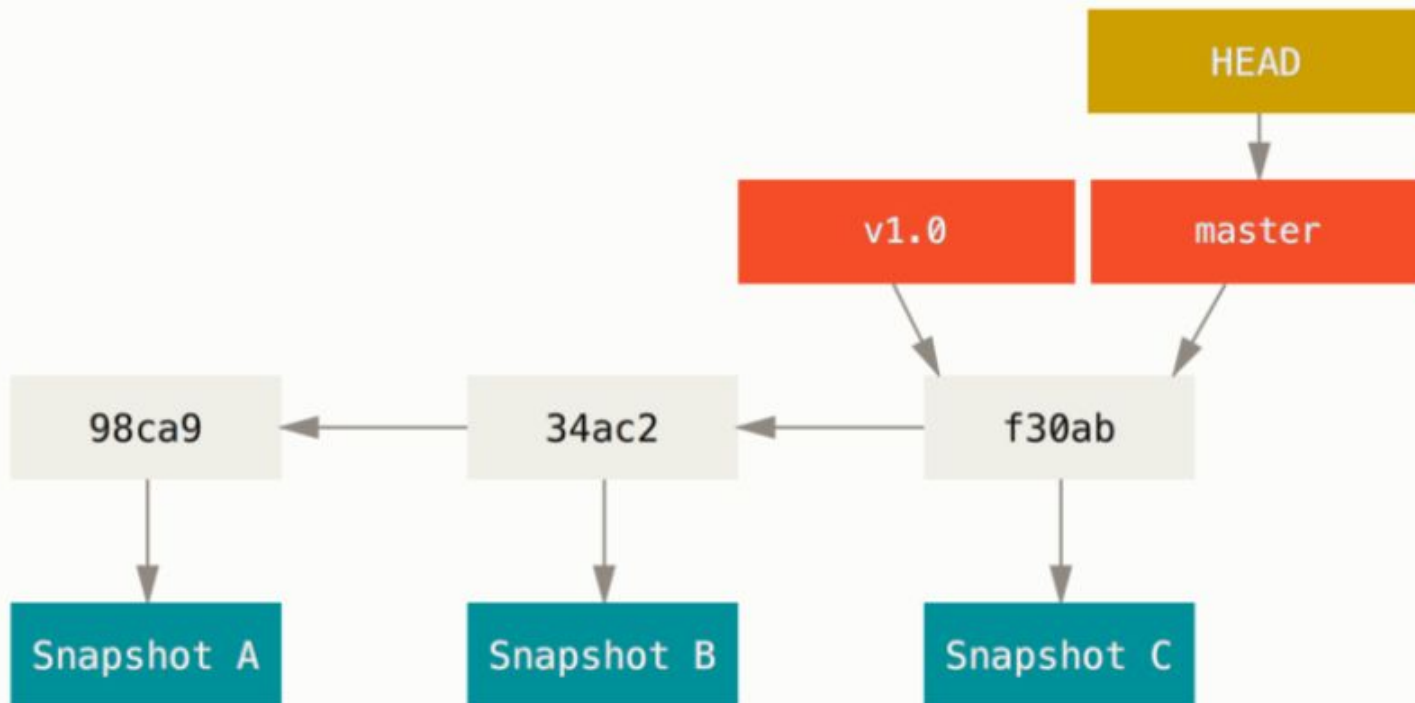
- три объекта;
- объект дерева каталогов;
- объект коммита.

ВЕТВЛЕНИЕ



По умолчанию, имя основной ветки в Git — **master**. Указатель ветки **master** будет передвигаться на следующий коммит автоматически

ВЕТВЛЕНИЕ



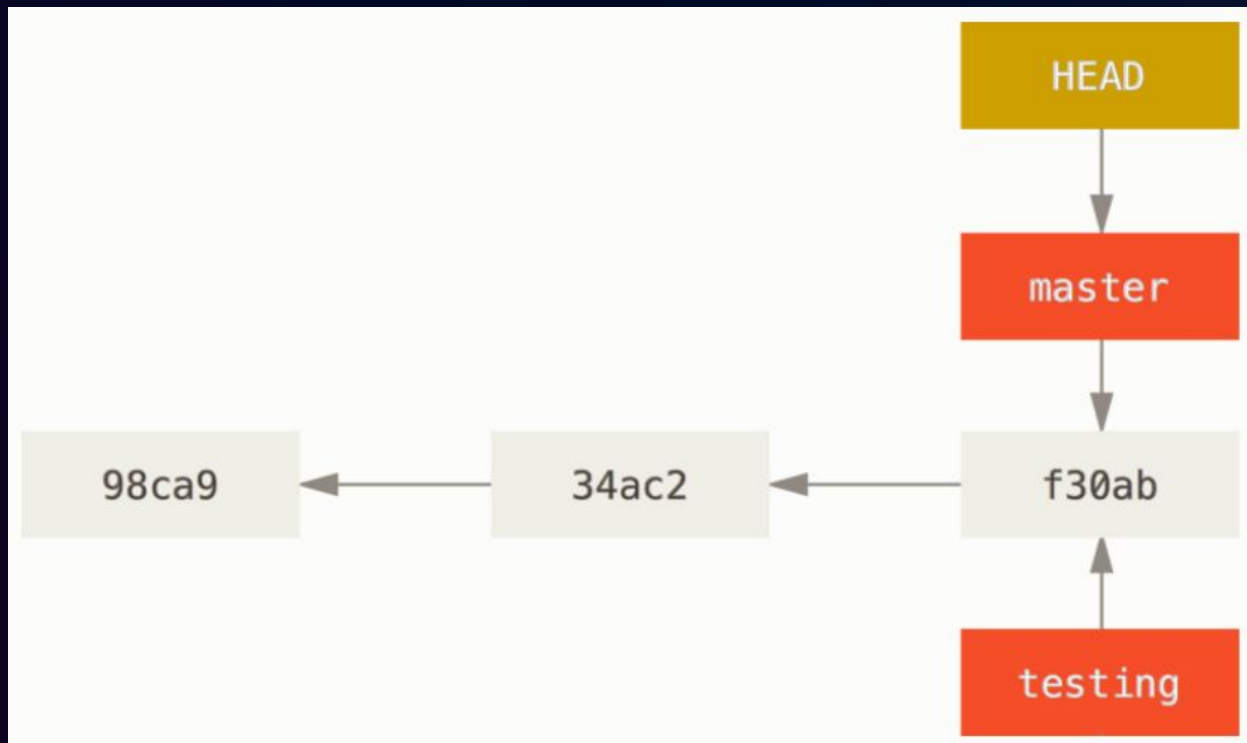
СОЗДАНИЕ НОВОЙ ВЕТКИ

Для создания новой ветки необходимо прописать команду:

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (main)  
$ git branch testing
```

Команда `git branch` только создаёт новую ветку, но не переключает на неё.

СОЗДАНИЕ НОВОЙ ВЕТКИ



Git хранит специальный указатель HEAD (это указатель на текущую локальную ветку).

ПЕРЕКЛЮЧЕНИЕ ВЕТОК



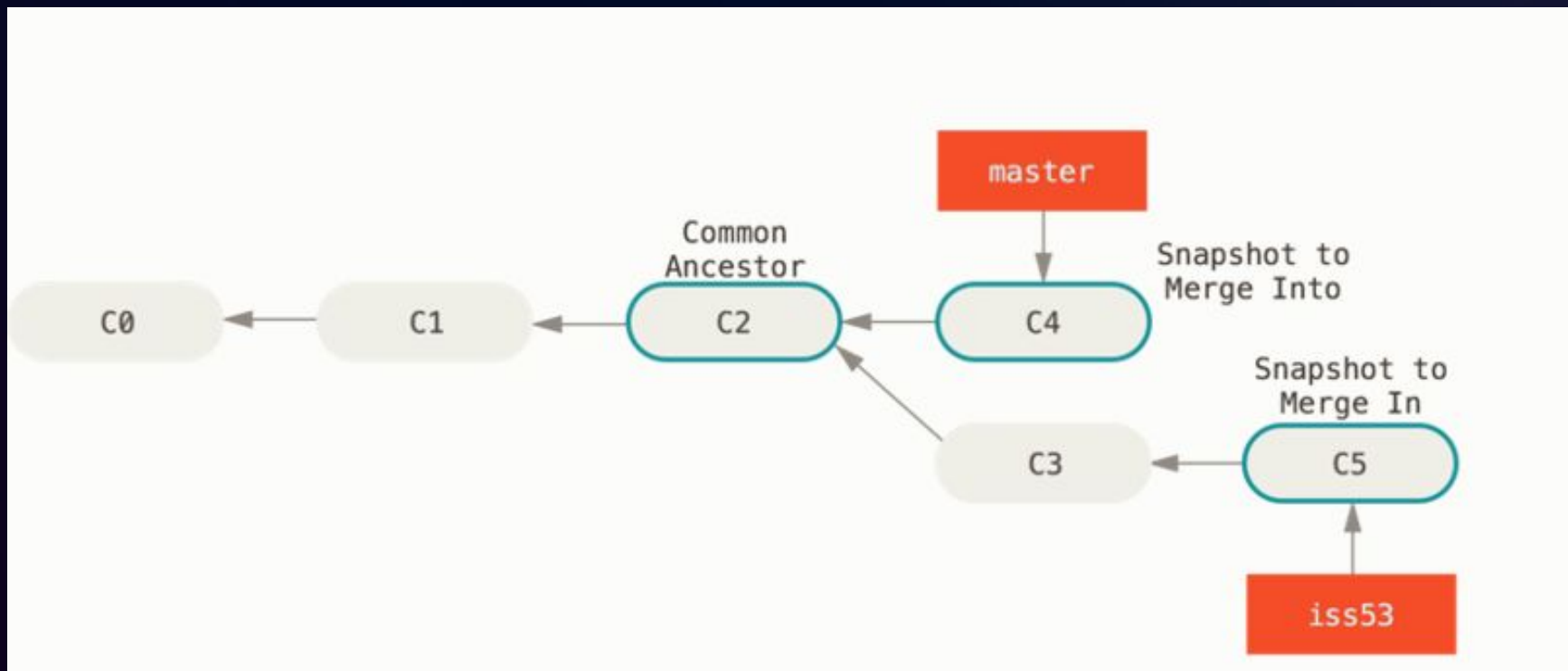
Для переключения веток необходимо
выполнить команду:

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (main)
$ git checkout testing
Switched to branch 'testing'
```

РАБОТА С ВЕТКАМИ



СЛИЯНИЕ ВЕТОК



СЛИЯНИЕ ВЕТОК

Для того, чтобы влить в ветку testing ветку testing2, необходимо переключиться на ветку, в которую вы хотите включить изменения, и выполнить команду git merge:

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 групна (testing2)
$ git checkout testing
Switched to branch 'testing'

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 групна (testing)
$ git merge testing2
Updating 3068422..02fc9cd
Fast-forward
 testing2.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 testing2.txt
```



ВОЗВРАЩЕНИЕ К ПРЕДЫДУЩЕЙ ВЕРСИИ



Команда `git checkout` позволяет переключаться между версиями

КОММИТОВ

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 гpyнна (master)
$ git log
commit 9349bcef0832da9814928ac6603e5a598f47bd43 (HEAD -> master)
Author: Regina Zakirova <Regina_174@mail.ru>
Date: Wed Dec 2 17:11:33 2020 +0500
```

second

```
commit 0532fdc027dd7403dc8559795444a498981ed92c
Author: Regina Zakirova <Regina_174@mail.ru>
Date: Wed Dec 2 17:10:01 2020 +0500
```

first

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 гpyнна (master)
```

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 гpyнна (master)
$ git checkout 0532fdc027dd7403dc8559795444a498981ed92c
Note: checking out '0532fdc027dd7403dc8559795444a498981ed92c'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-b` with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

```
HEAD is now at 0532fdc first
```

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 гpyнна ((0532fdc...))
$ _
```

ОТМЕНА ИНДЕКСАЦИИ

С помощью команды `git reset` мы можем вернуть в исходное состояние ветку, отдельный коммита, либо файл к тому состоянию, которому мы хотим. В нашем случае это HEAD. HEAD – это то последнее состояние, в котором был файл до

ИЗМЕНЕНИЙ

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Deskto
$ git reset HEAD hello.txt
Unstaged changes after reset:
M      hello.txt
```

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   hello.txt

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

Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа (master)
$
```


ОТМЕНА КОММИТА

Revert – это команда возврата. Git создаст новый коммит с отменой последнего коммита. Эта команда добавит новый коммит в историю текущей ветки, после чего та будет выглядеть следующим образом

```
Regina@LAPTOP-JM4R8740 MINGW64 ~/Desktop/4 группа/dev (main)
$ git revert HEAD --no-edit
[main 5f40ced] Revert "5"
Date: Thu Dec 3 16:36:27 2020 +0500
1 file changed, 0 insertions(+), 0 deletions(-)
```

ПОЛЕЗНЫЕ ССЫЛКИ

- <https://githowto.com/ru>
- <https://git-scm.com/book/ru/v2>