

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2**  
**дисциплины «Программирование на Python»**

Выполнил:

Лейс Алексей Вячеславович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»

---

(подпись)

Руководитель практики: кандидат тех.  
наук доцент кафедры  
инфокоммуникаций: Воронкин Р.А

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

**Тема:** Исследование возможностей Git для работы с локальными репозиториями

**Цель работы:** исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Порядок выполнения работы:

```
Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ git tag -a v1 -m "version 1"

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ git push
Enumerating objects: 45, done.
Counting objects: 100% (45/45), done.
Delta compression using up to 4 threads
Compressing objects: 100% (31/31), done.
Writing objects: 100% (44/44), 33.97 KiB | 1.26 MiB/s, done.
Total 44 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/LAVgame/lab_1.2
   8954bc6..6ccd478  main -> main

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 6ccd478 (HEAD -> main, tag: v1, origin/main, origin/HEAD) one commit
* 8954bc6 Initial commit

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ |
```

Добавление тега с комментарием

```
modified: Lab_2/2/.vs/2/v16/.suo
modified: Lab_2/2/Program.cs
modified: Lab_2/2/bin/Debug/netcoreapp3.1/2.pdb
modified: Lab_2/2/obj/2.csproj.nuget.dgspec.json
modified: Lab_2/2/obj/Debug/netcoreapp3.1/2.GeneratedMSBuildEditorConfig.editorconfig
fig
modified: Lab_2/2/obj/Debug/netcoreapp3.1/2.assets.cache
modified: Lab_2/2/obj/Debug/netcoreapp3.1/2.csproj.AssemblyReference.cache
modified: Lab_2/2/obj/Debug/netcoreapp3.1/2.csproj.FileListAbsolute.txt
modified: Lab_2/2/obj/Debug/netcoreapp3.1/2.genruntimeconfig.cache
modified: Lab_2/2/obj/Debug/netcoreapp3.1/2.pdb
modified: Lab_2/2/obj/project.assets.json
modified: Lab_2/2/obj/project.nuget.cache

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ git tag -a v2 -m "two"

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ git commit -m "two commit"
[main 97f557f] two commit
13 files changed, 45 insertions(+), 18 deletions(-)
rewrite Lab_2/2/.vs/2/v16/.suo (90%)
rewrite Lab_2/2/obj/Debug/netcoreapp3.1/2.csproj.AssemblyReference.cache (100%)

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$ git push
Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 4 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (26/26), 4.48 KiB | 327.00 KiB/s, done.
Total 26 (delta 10), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (10/10), completed with 10 local objects.
To https://github.com/LAVgame/lab_1.2
   6ccd478..97f557f  main -> main

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/программирование на Python/2/Lab_1.2 (main)
$
```

```

student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/Lab_1.2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 97f557f (HEAD -> main, origin/main, origin/HEAD) two commit
* 6ccd478 (tag: v2, tag: v1) one commit
* 8954bc6 Initial commit

student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/Lab_1.2 (main)
$

```

Выполнение команды: `git log --graph --pretty=oneline --abbrev-commit` – которая просматривает журналы с выполненными действиями

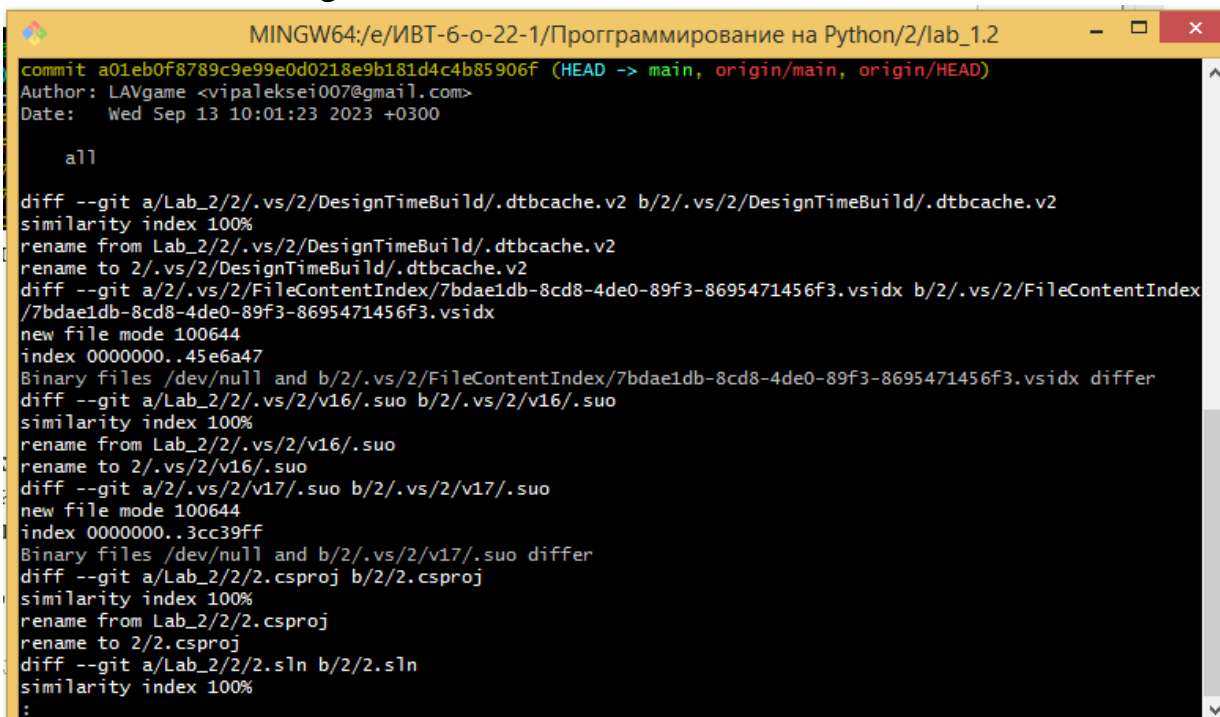
```

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/Lab_1.2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* e4f31fb (HEAD -> main, tag: v4, origin/main, origin/HEAD) 4 commit
* 0d676fa (tag: v3) 3 commit
* 97f557f two commit
* 6ccd478 (tag: v2, tag: v1) one commit
* 8954bc6 Initial commit

```

Журнал после нескольких тегов и коммитов

### 1. Выполнение: `git show HEAD`



```

commit a01eb0f8789c9e99e0d0218e9b181d4c4b85906f (HEAD -> main, origin/main, origin/HEAD)
Author: LAVgame <vipaleksei007@gmail.com>
Date:   Wed Sep 13 10:01:23 2023 +0300

    all

diff --git a/Lab_2/2/.vs/2/DesignTimeBuild/.dtbcache.v2 b/2/.vs/2/DesignTimeBuild/.dtbcache.v2
similarity index 100%
rename from Lab_2/2/.vs/2/DesignTimeBuild/.dtbcache.v2
rename to 2/.vs/2/DesignTimeBuild/.dtbcache.v2
diff --git a/2/.vs/2/FileContentIndex/7bdae1db-8cd8-4de0-89f3-8695471456f3.vsidx b/2/.vs/2/FileContentIndex/7bdae1db-8cd8-4de0-89f3-8695471456f3.vsidx
new file mode 100644
index 0000000..45e6a47
Binary files /dev/null and b/2/.vs/2/FileContentIndex/7bdae1db-8cd8-4de0-89f3-8695471456f3.vsidx differ
diff --git a/Lab_2/2/.vs/2/v16/.suo b/2/.vs/2/v16/.suo
similarity index 100%
rename from Lab_2/2/.vs/2/v16/.suo
rename to 2/.vs/2/v16/.suo
diff --git a/2/.vs/2/v17/.suo b/2/.vs/2/v17/.suo
new file mode 100644
index 0000000..3cc39ff
Binary files /dev/null and b/2/.vs/2/v17/.suo differ
diff --git a/Lab_2/2/2.csproj b/2/2.csproj
similarity index 100%
rename from Lab_2/2/2.csproj
rename to 2/2.csproj
diff --git a/Lab_2/2/2.sln b/2/2.sln
similarity index 100%
:

```

### 2. Выполнение `git show HEAD~1`

```

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/lab_
2 (main)
$ git show HEAD~1
commit e4f31fb2e02619fee3bf492021fb2a171f829833 (tag: v4)
Author: LAVgame <vipaleksei007@gmail.com>
Date: Mon Sep 11 12:33:19 2023 +0300

    4 commit

diff --git a/Lab_2/2/.vs/2/v16/.suo b/Lab_2/2/.vs/2/v16/.suo
index 9bbd5b1..3cc39ff 100644
Binary files a/Lab_2/2/.vs/2/v16/.suo and b/Lab_2/2/.vs/2/v16/.suo differ
diff --git a/Lab_2/2/Program.cs b/Lab_2/2/Program.cs
index 1b3760e..99fc9ab 100644
--- a/Lab_2/2/Program.cs
+++ b/Lab_2/2/Program.cs
@@ -15,37 +15,27 @@ namespace _2
     float v = Convert.ToSingle(Console.ReadLine());
     Console.WriteLine("Введите число C = ");
     float c = Convert.ToSingle(Console.ReadLine());
-    Console.WriteLine(" ");
-    Console.WriteLine($"A + B = {a+b}");
-    Console.WriteLine($"A - B = {a-b}");
-    Console.WriteLine($"A x B = {a*b}");
-    Console.WriteLine($"A : B = {a/b:0.00}");
-    Console.WriteLine(" ");
-    Console.WriteLine($"A + B = {a + v}");
-    Console.WriteLine($"A - B = {a - v}");
-    Console.WriteLine($"A x B = {a * v}");
-    Console.WriteLine($"A : B = {a / v:0.00}");

```

Откат:

Перед выполнением удалил из кода часть программы

```

Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/lab_1.
2 (main)
$ git checkout -- 2/Program.cs

```

Удаление несохранённых изменений

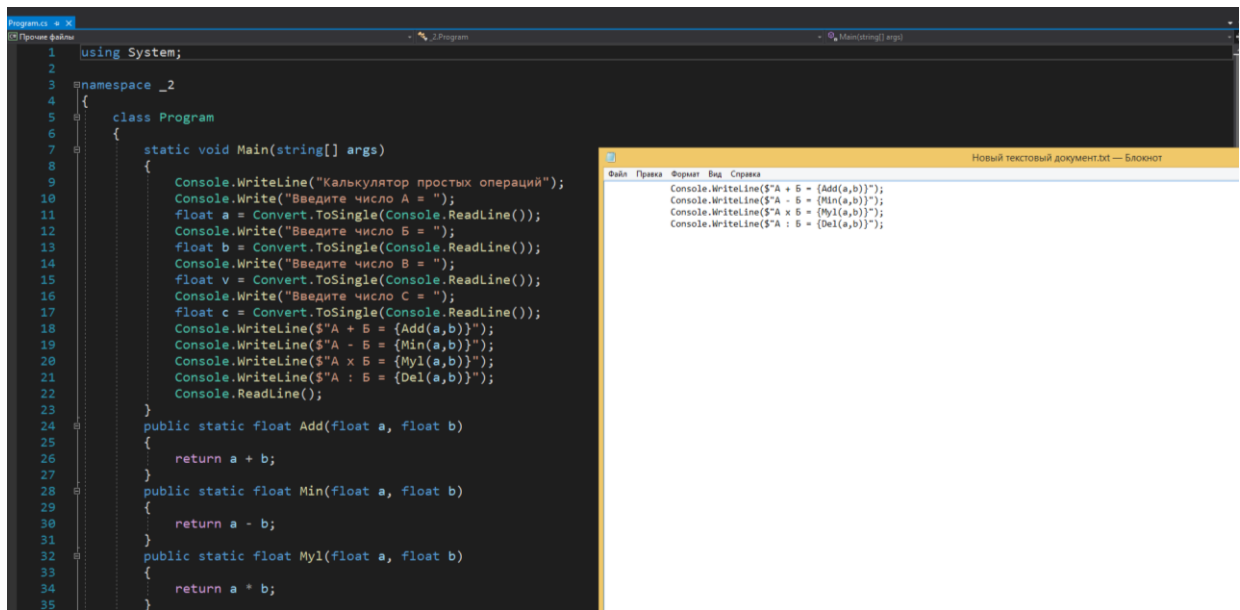
```

$ git add .
Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/lab_1.
2 (main)
$ git commit -m "delete files Program"
[main 5ab0325] delete files Program
15 files changed, 42 insertions(+), 77 deletions(-)
rewrite 2/.vs/2/v16/.suo (78%)
create mode 100644 2/.vs/2/v17/.futdcache.v2
rewrite 2/.vs/2/v17/.suo (85%)
create mode 100644 2/.vs/ProjectEvaluation/2.metadata.v7.bin
create mode 100644 2/.vs/ProjectEvaluation/2.projects.v7.bin
rewrite 2/obj/Debug/netcoreapp3.1/2.csproj.AssemblyReference.cache (100%)
rewrite 2/obj/project.nuget.cache (83%)

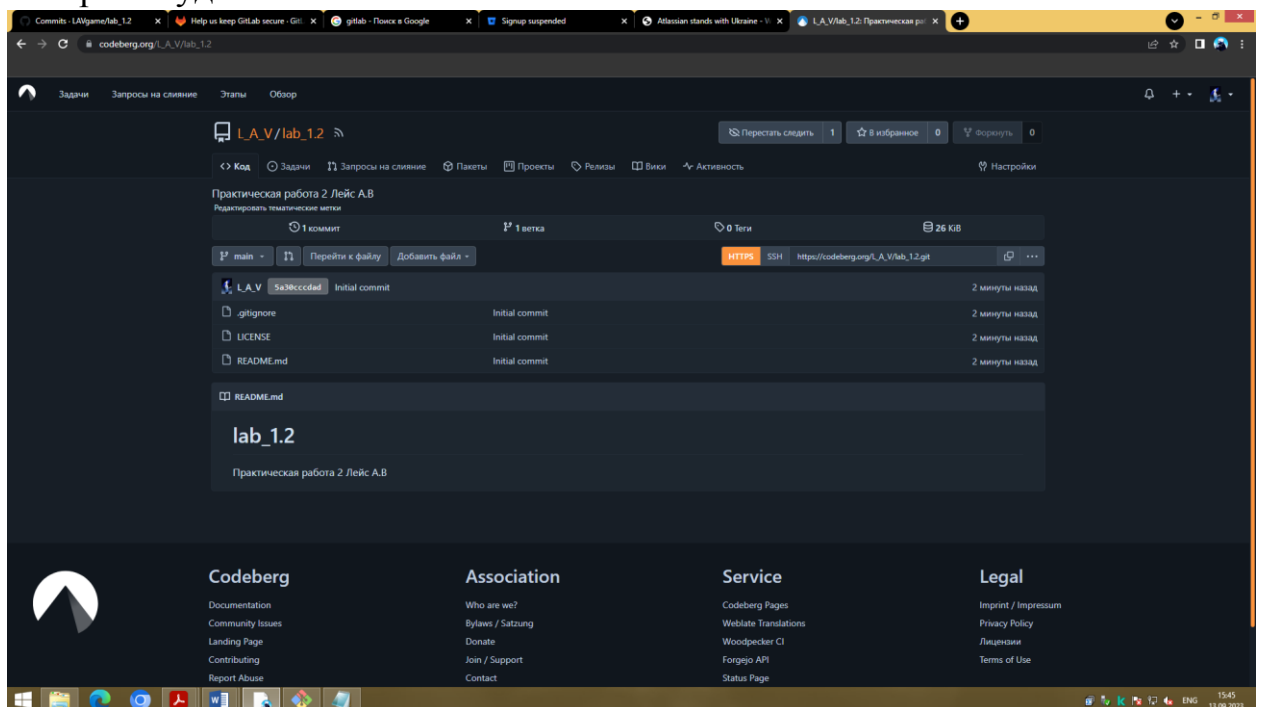
Student@PC-01-9_521 MINGW64 /e/ИВТ-6-о-22-1/Программирование на Python/2/lab_1.
2 (main)
$ git reset --hard HEAD~1
HEAD is now at a01eb0f all

```

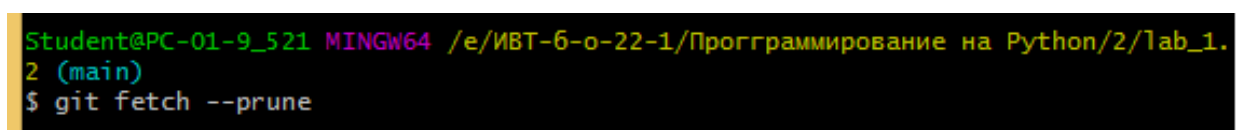
Откат состояние хранилища к предыдущей версии командой



Файл вернулся к изначальному состоянию, справа на рисунке строчки кода которые я удалял.



Создание учетной записи и репозитория на CodeBerg



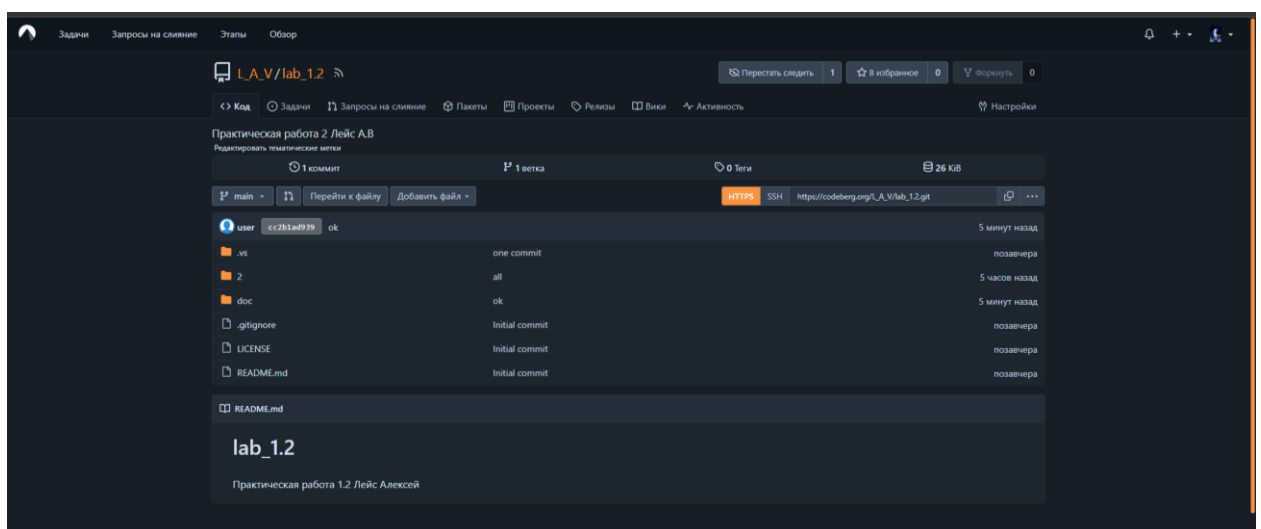
Выполнение команды git fetch --prune

```

Student@PC-01-9_521 MINGW64 /e/ИБТ-6-о-22-1/Программирование на Python/2/lab_1.
2 (main)
$ git push --prune https://codeberg.org/L_A_V/lab_1.2.git +refs/remotes/origin/*
:refs/heads/* +refs/tags/*:refs/tags/*
info: detecting host provider for 'https://codeberg.org/'...
info: detecting host provider for 'https://codeberg.org/'...
Enumerating objects: 127, done.
Counting objects: 100% (127/127), done.
Delta compression using up to 4 threads
Compressing objects: 100% (92/92), done.
Writing objects: 100% (127/127), 2.36 MiB | 636.00 KiB/s, done.
Total 127 (delta 36), reused 4 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (36/36), done.
remote:
remote: Create a new pull request for 'HEAD':
remote:   https://codeberg.org/L_A_V/lab_1.2/compare/main...HEAD
remote:
remote: ..... Processing 6 references
remote: Processed 6 references in total
To https://codeberg.org/L_A_V/lab_1.2.git
+ 5a30ccc...cc2b1ad origin/main -> main (forced update)
* [new branch]      origin/HEAD -> HEAD
* [new tag]         v1 -> v1
* [new tag]         v2 -> v2
* [new tag]         v3 -> v3
* [new tag]         v4 -> v4

```

Выполнение команды на перенос репозитория на другой сервис codeberg



Результат переноса

Ссылка: [https://codeberg.org/L\\_A\\_V/lab\\_1.2](https://codeberg.org/L_A_V/lab_1.2)

### Ответы на вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

После того, как вы создали несколько коммитов или же клонировали репозиторий с уже существующей историей коммитов, вероятно Вам понадобится возможность посмотреть что было сделано — историю

коммитов. Одним из основных и наиболее мощных инструментов для этого является команда `git log`

Следующей действительно полезной опцией является `--pretty`. Эта опция меняет формат вывода.

## 2. Как ограничить вывод при просмотре истории коммитов?

В дополнение к опциям форматирования вывода, команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. Одна из таких опций — это опция `-2`, которая показывает только последние два коммита. В действительности вы можете использовать `-<n>`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов.

## 3. Как внести изменения в уже сделанный коммит?

Можно восстановить даже коммиты из веток, которые были удалены, или коммиты, перезаписанные параметром `--amend`. Но всё, что не было включено в коммит и потеряно — скорее всего, потеряно навсегда.

## 4. Как отменить индексацию файла в Git?

Использовать `git reset HEAD <file>...` для исключения из индекса

## 5. Как отменить изменения в файле?

Использовать `"git reset HEAD <file>..."` для возвращения к версии из последнего коммита.

## 6. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

## 7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`.

## 8. Как добавить удаленный репозиторий для данного локального репозитория?

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), просто выполните команду `git remote add <shortname> <url>`.

## 9. Как выполнить отправку/получение изменений с удаленного репозитория?

Для получения данных из удалённых проектов, следует выполнить `$ git fetch [remote-name]`.

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push <remote-name> <branch-name>`.

## 10. Как выполнить просмотр удаленного репозитория?

Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`.

#### 11. Каково назначение тэгов Git?

Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.).

#### 12. Как осуществляется работа с тэгами Git?

Достаточно набрать команду `git tag` (параметры `-l` и `--list` опциональны).

#### 13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

**Вывод:** На основе выполненной работы я исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями.