

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 ЛАБОРАТОРНОЙ РАБОТЕ №2**  
**дисциплины «Программирование на Python»**  
**Вариант \_\_\_\_**

Выполнил:  
Дудкин Константин Александрович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»  
направление «Программное  
обеспечение средств вычислительной  
техники и автоматизированных  
систем»,  
очная форма обучения

---

(подпись)

Руководитель практики:  
Кандидат технических наук, доцент  
кафедры инфокоммуникаций, доцент  
Воронкин Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

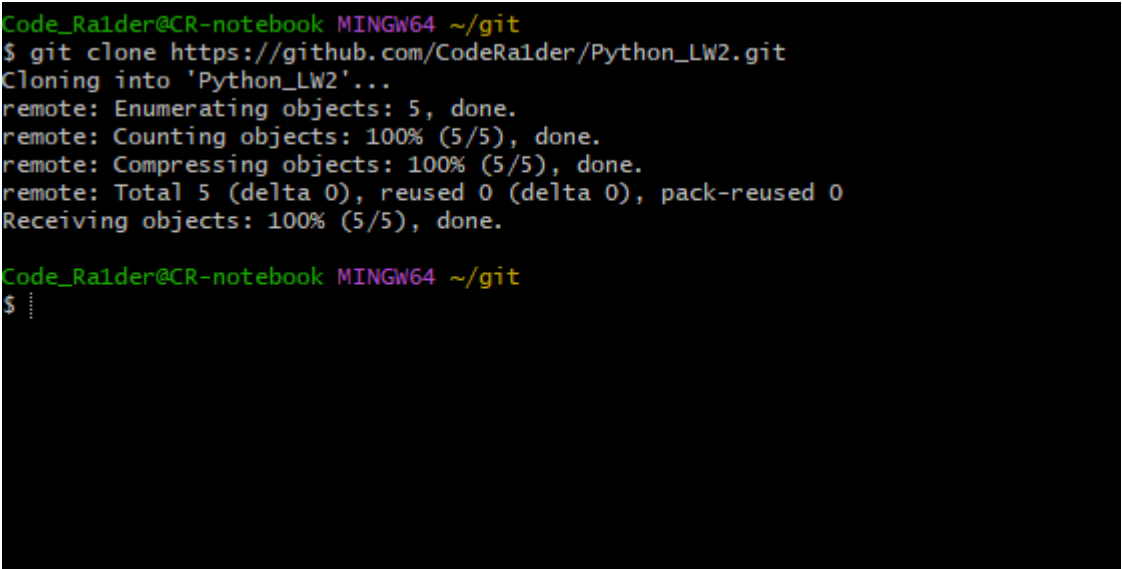
Тема: Исследование возможностей Git для работы с локальными репозиториями

Цель: Исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями

### Порядок выполнения работы

1. Изучил теоретический материал и создал общедоступный репозиторий с лицензией MIT

2. Клонировал свой репозиторий:

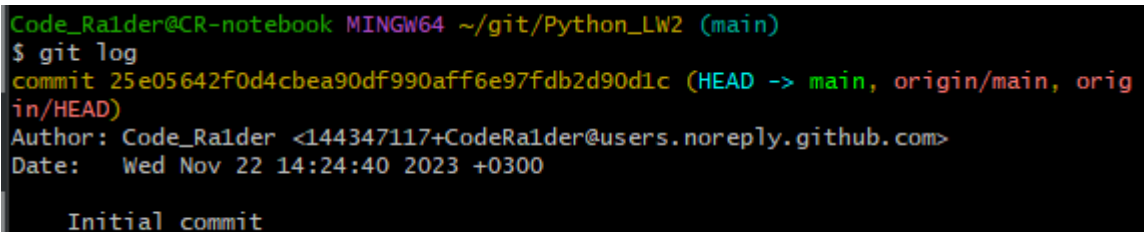


```
Code_Ra1der@CR-notebook MINGW64 ~/git
$ git clone https://github.com/CodeRa1der/Python_LW2.git
Cloning into 'Python_LW2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

Code_Ra1der@CR-notebook MINGW64 ~/git
$ .....
```

Рисунок 1. Клонирование созданного репозитория

3. Воспользовался командой git log и всеми представленными внутри методических материалов вариантами данной команды:



```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log
commit 25e05642f0d4cbea90df990aff6e97fdb2d90d1c (HEAD -> main, origin/main, origin/HEAD)
Author: Code_Ra1der <144347117+CodeRa1der@users.noreply.github.com>
Date:   Wed Nov 22 14:24:40 2023 +0300

    Initial commit
```

Рисунок 2. Команда git log

```

Code_Raider@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log -p -2
commit 25e05642f0d4cbea90df990aff6e97fdb2d90d1c (HEAD -> main, origin/main, origin/HEAD)
Author: Code_Raider <144347117+CodeRaider@users.noreply.github.com>
Date:   Wed Nov 22 14:24:40 2023 +0300

    Initial commit

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..68bc17f
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,160 @@
+# Byte-compiled / optimized / DLL files
+__pycache__/
+*.py[cod]
+*.py.class
+
+# C extensions
+*.so
+
+# Distribution / packaging
+*.Python
+build/
+develop-eggs/
+dist/
+downloads/
+eggs/
+*.egg
+lib/
+lib64/
+parts/
+sdist/
+var/
+wheels/
+share/python-wheels/
+*.egg-info/
+*.installed.cfg
+*.egg
+MANIFEST
+
+# PyInstaller
+# Usually these files are written by a python script from a template
+# before PyInstaller builds the exe, so as to inject date/other infos into it.
+*.manifest
+*.spec
+
+# Installer logs
+pip-log.txt
+pip-delete-this-directory.txt

```

Рисунок 3. Команда git log -p -2

```

Code_Raider@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log --stat
commit 25e05642f0d4cbea90df990aff6e97fdb2d90d1c (HEAD -> main, origin/main, origin/HEAD)
Author: Code_Raider <144347117+CodeRaider@users.noreply.github.com>
Date:   Wed Nov 22 14:24:40 2023 +0300

    Initial commit

 .gitignore | 160 +++++
 LICENSE    |  21 +++++
 README.md  |   2 +
 3 files changed, 183 insertions(+)

```

Рисунок 4. Команда git log --stat

```

Code_Raider@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log --pretty=oneline
25e05642f0d4cbea90df990aff6e97fdb2d90d1c (HEAD -> main, origin/main, origin/HEAD) Initial commit

```

Рисунок 5. Команда git log --pretty=oneline

```

Code_Raider@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log --pretty=format:"%h - %an, %ar : %s"
25e0564 - Code_Raider, 66 minutes ago : Initial commit

```

Рисунок 6. Команда git log --pretty=format:"%h - %an, %ar : %s"

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log --pretty=format:"%h %s" --graph
* 25e0564 Initial commit
```

Рисунок 7. Команда git log --pretty=format:"%h %s" --graph

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log --since=2.weeks
commit 25e05642f0d4cbea90df990aff6e97fdb2d90d1c (HEAD -> main, origin/main, origin/HEAD)
Author: Code_Ra1der <144347117+CodeRa1der@users.noreply.github.com>
Date: Wed Nov 22 14:24:40 2023 +0300

Initial commit
```

Рисунок 8. Команда git log --since=2.weeks

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 25e0564 (HEAD -> main, origin/main, origin/HEAD) Initial commit
```

Рисунок 9. Команда git log --graph --pretty=oneline --abbrev-commit

4. Написал простенькую программу для коммита и просмотрел коммиты через git show:

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git show HEAD
commit 81582b0fcfd7396067307cfffbb919f29efed2fd9 (HEAD -> main, origin/main, origin/HEAD)
Author: CodeRa1der <kostydu2342@gmail.com>
Date: Wed Nov 22 15:37:43 2023 +0300

Renamed program

diff --git a/1 or 2.py b/2 or 1.py
similarity index 100%
rename from 1 or 2.py
rename to 2 or 1.py
```

Рисунок 10. Команда git show HEAD

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git show HEAD~1
commit 2a06016d235499d0584d885169f58258150c220f
Author: CodeRa1der <kostydu2342@gmail.com>
Date: Wed Nov 22 15:37:12 2023 +0300

Added program

diff --git a/1 or 2.py b/1 or 2.py
new file mode 100644
index 0000000..a700895
--- /dev/null
+++ b/1 or 2.py
@@ -0,0 +1,5 @@
+num = int(input("1 or 2? "))
+if num == 1:
+    print("This is 1!!!")
+else:
+    print("This is 2!!!")
```

Рисунок 11. Команда git show HEAD~1

5. Освоил возможность отката репозитория к заданной версии: воспользовался командой `git checkout`, чтобы убрать все несохранённые изменения, и командой `git reset`, чтобы выполнить откат к предыдущей версии репозитория

```
Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git checkout -- README.md

Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git reset --hard HEAD~1
HEAD is now at 81582b0 Renamed program

Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$
```

Рисунок 12. `git checkout` и `git reset`

6. Изучил команды для работы с удалёнными репозиториями:

```
Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote
origin

Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote -v
\origin https://github.com/CodeRalder/Python_LW2.git (fetch)
origin https://github.com/CodeRalder/Python_LW2.git (push)

Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote add pb https://github.com/CodeRalder/Python_LW2.git

Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git fetch pb
From https://github.com/CodeRalder/Python_LW2
* [new branch]      main      -> pb/main

Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ |
```

Рисунок 13. Использование команд `git remote`, `git remote -v`, `git remote add`, `git fetch`

```
Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote rename pb pablo
Renaming remote references: 100% (1/1), done.
```

Рисунок 14. `git remote rename`

```
Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote
origin
pablo
```

Рисунок 15. Результат работы прошлой команды

```
Code_Ralder@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote remove pablo
```

Рисунок 16. Удаление репозитория

7. Изучил команды для работы с тегами:

```
CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git tag

CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git tag -l

CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git tag

CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git tag -a v1.0 -m "First version"

CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git show v1.0
tag v1.0
Tagger: CodeRa1der <kostydu2342@gmail.com>
Date: Thu Dec 14 10:33:14 2023 +0300

First version

commit 81582b0fcfd7396067307cfffbb919f29efed2fd9 (HEAD -> main, tag: v1.0)
Author: CodeRa1der <kostydu2342@gmail.com>
Date: Wed Nov 22 15:37:43 2023 +0300

    Renamed program

diff --git a/1 or 2.py b/2 or 1.py
similarity index 100%
rename from 1 or 2.py
rename to 2 or 1.py
```

Рисунок 17. git tag, git tag -a и git show <tag>

```
CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git push origin --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 164 bytes | 164.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/CodeRa1der/Python_LW2.git
 * [new tag]          v1.0 -> v1.0

CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git tag -d v1.0
Deleted tag 'v1.0' (was 6b32a7a)
```

Рисунок 18. Отправка тега в удалённый репозиторий и удаление тега в локальном репозитории

```
CodeRa1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git push origin --delete v1.0
To https://github.com/CodeRa1der/Python_LW2.git
- [deleted]          v1.0
```

Рисунок 19. Удаление тега в удалённом репозитории

8. Зарегистрировался на BitBucket и создал в нём репозиторий. Скопировал репозиторий GitHub в созданный репозиторий

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote add bit https://Code_Ra1der@bitbucket.org/code-ra1der/lw2.git

Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git remote
bit
origin
```

Рисунок 20. Добавление репозитория

```
Code_Ra1der@CR-notebook MINGW64 ~/git/Python_LW2 (main)
$ git push -f bit main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 3.27 KiB | 3.27 MiB/s, done.
Total 10 (delta 2), reused 4 (delta 0), pack-reused 0
To https://bitbucket.org/code-ra1der/lw2.git
+ b2c369f...81582b0 main -> main (forced update)
```

Рисунок 21. Копирование репозитория на BitBucket

Code\_Ra1der / LW2

**LW2**

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

main

Name	Size	Last commit	Message
.gitign...	3.01 KB	2023-11-22	Initial commit
2 or 1...	100 B	2023-11-22	Renamed program
LICENSE	1.04 KB	2023-11-22	Initial commit
READ...	102 B	2023-11-22	Initial commit

README.md

## Python\_LW2

Программирование на Python. Лабораторная работа №2

## Рисунок 22. Копированный репозиторий на BitBucket

### Контрольные вопросы

1. Для просмотра истории коммитов используют команду `git log`. Для данной команды применимы следующие опции:

- p - показывает патч для каждого коммита
- stat - показывает статистику изменённых файлов
- shortstat - показывает только строку с кол-вом изменённых файлов
- name-only - показывает список изменённых файлов после информации о коммите
- name-status - показывает файлы, которые были изменены
- abbrev-ccommit - показывает часть символов чек-суммы SHA-1
- graph - отображает ASCII граф с ветвлениями и слияниями репозитория
- pretty и --oneline - показывает коммиты в альтернативном формате

2. Для ограничения вывода `git log` тоже используются опции. Например. опция `-(n)` показывает последние n коммиты, `--author` - коммиты, где запись `author` совпадает с указанной строкой после опции и т.д.

3. Для изменения уже сделанного коммита можно отменить коммит командой `git commit --amend` и создать новый с новыми изменениями

4. Для отмены индексации файла используется команда `git reset <имя файла>`

5. Для отмены изменений в файле используют команду `git checkout -- <имя файла>`

6. Удалённый репозиторий Git - это репозиторий, сохранённый в интернете на каком-либо сервере. Преимущество удалённого репозитория - возможность его использования другими пользователями (если он публичный)

7. Для просмотра удалённых репозиториях используют команду `git remote`



8. Для добавления удалённых репозиториях используют команду `git remote add <имя репозитория> <ссылка на репозиторий>`

9. Для получения изменений, сделанных на удалённом репозитории, используется команда `git fetch` и `git pull`. Отличие данных команд - `git pull` выполняет слияние полученных наработок с локальным репозиторием, в то время как `git fetch` - нет

Для отправки изменений на удалённый репозиторий используется команда `git push`

10. Для просмотра удалённого репозитория используется команда `git remote show`

11. Тэг - метка, которая указывает на один из важных моментов в истории изменения репозитория. Они используются для упрощения

12. Работа с тэгами осуществляется посредством команды `git tag` с указанием в них определённых параметров:

`git tag -a <имя тэга>` - создание тэга

`git push origin <тэг>` - обмен тэгами

`git tag -d <тэг>` - удаление тэга

`git checkout <тэг>` - переход на тэг

13. Флаг `--prune` применяется в командах `git push` и `git fetch` следующим образом:

В `git push --prune` выполняет функцию удаления ветки из удалённого репозитория, которой больше нет в локальном репозитории

В `git fetch --prune` выполняет ту же функцию, что и в `git push`, но очистка происходит в сторону локального репозитория

Данный флаг необходим, если пользователю требуется очистить репозиторий от лишних веток и поддерживать репозиторий в чистом состоянии

Вывод: я изучил возможности Git в отношении работы с удалёнными репозиториями и репозиториями на сторонних ресурсах