

Лабораторная работа №5.

История

Цель: научиться просматривать историю изменений в различном виде.

1. За историю коммитов отвечает команда

`git log`

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log
commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Sat Oct 15 13:38:18 2022 +0300
```

Second Commit

```
commit 55660690da52e29538c3ada52cda69cb9095b726
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Wed Oct 5 19:33:11 2022 +0300
```

my first commit

```
commit 69664c156809ec00eb2d0c0624b88777355d0162
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Wed Oct 5 14:44:49 2022 +0300
```

First Commit

Примечание: Когда весь текст результата команды не убирается в окно терминала, результат выводится в несколько строк. Чтобы увидеть следующие строки необходимо нажать Enter. Когда выводить уже нечего, появляется метка конца данных END. Чтобы покинуть программу вывода, нужно нажать клавишу q (quit - покинуть).

```
+Hello,world!!!
+</body>
+</html>
\ No newline at end of file
(END)
```

```
+</html>
\ No newline at end of file
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ |
```

2. Чтобы подробно изучить изменения, внесенные в каждый файл, следует добавить к команде флаг `-p`:

```
git log -p
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log -p
commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Sat Oct 15 13:38:18 2022 +0300

    Second Commit

diff --git a/hello.html b/hello.html
index 4a2abd2..ffd18b2 100644
--- a/hello.html
+++ b/hello.html
@@ -6,5 +6,15 @@
 <body>
 <h1>Hello,world!!!</h1>
 Я учусь работать с Git
+<table>
+  <tr>
+    <td>Git</td>
+    <td> - система управления версиями с распределенной архитектурой</td>
+  </tr>
+  <tr>
+    <td>GitHub</td>
+    <td> - служба размещения в Интернете репозитория Git</td>
+  </tr>
+...skipping...
commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Sat Oct 15 13:38:18 2022 +0300

    Second Commit
```

3. Сделаем вывод более удобным для чтения, с помощью команды

```
git log --oneline
```

выведем только id коммита и сообщение.

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --oneline
be72de3 (HEAD -> main, origin/main) Second Commit
5566069 my first commit
69664c1 First Commit
```

HEAD -> main означает, что мы в ветке main.

4. Следующая команда позволяет увидеть сообщения конкретного автора:

```
git log --author="user_name"
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --author="for-you-nastya"
commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Sat Oct 15 13:38:18 2022 +0300
```

Second Commit

```
commit 55660690da52e29538c3ada52cda69cb9095b726
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Wed Oct 5 19:33:11 2022 +0300
```

my first commit

```
commit 69664c156809ec00eb2d0c0624b88777355d0162
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Wed Oct 5 14:44:49 2022 +0300
```

First Commit

5. Просмотреть историю коммитов в виде графика для текущей ветки можно с помощью флагов `--graph`.

```
git log --graph
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --graph
* commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
| Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
| Date: Sat Oct 15 13:38:18 2022 +0300
|
| Second Commit
|
* commit 55660690da52e29538c3ada52cda69cb9095b726
| Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
| Date: Wed Oct 5 19:33:11 2022 +0300
|
| my first commit
|
* commit 69664c156809ec00eb2d0c0624b88777355d0162
| Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
| Date: Wed Oct 5 14:44:49 2022 +0300
|
| First Commit
```

Опция `--graph` выведет график в формате ASCII, отражающий структуру ветвления истории коммитов. В связке с флагами `--oneline` и `--decorate`, этот флаг упрощает понимание того, к какой ветке относится каждый коммит:

```
git log --graph --oneline --decorate
```

Для просмотра истории коммитов по всем веткам используется флаг `--all`.

```
git log --all --graph --oneline --decorate
```

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --all --graph --oneline --decorate
* be72de3 (HEAD -> main, origin/main) Second Commit
* 5566069 my first commit
* 69664c1 First Commit

```

6. Если вы хотите получить заданное количество записей, то следует ввести:

```
git log -n количество
```

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log -n 2
commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Sat Oct 15 13:38:18 2022 +0300

    Second Commit

commit 55660690da52e29538c3ada52cda69cb9095b726
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Wed Oct 5 19:33:11 2022 +0300

    my first commit

```

7. Далее возможен поиск по сообщению комментария:

```
git log --grep="комментарий"
```

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --grep="Second Commit"
commit be72de3e6caa8286e7f9ebe11575900449f00827 (HEAD -> main, origin/main)
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Sat Oct 15 13:38:18 2022 +0300

    Second Commit

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --grep="my first commit"
commit 55660690da52e29538c3ada52cda69cb9095b726
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date: Wed Oct 5 19:33:11 2022 +0300

    my first commit

```

8. И часто бывает полезен поиск по дате:

```
git log --since=1.month.ago --until=1.weeks.ago
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --since=1.month.ago --until=1.weeks.ago
commit 55660690da52e29538c3ada52cda69cb9095b726
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date:   Wed Oct 5 19:33:11 2022 +0300
```

my first commit

```
commit 69664c156809ec00eb2d0c0624b88777355d0162
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date:   Wed Oct 5 14:44:49 2022 +0300
```

First Commit

9. Параметры поиска можно сочетать, например, сделать поиск по дате и автору.

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (main)
$ git log --since=1.month.ago --until=1.day.ago --author="for-you-nastya"
commit 55660690da52e29538c3ada52cda69cb9095b726
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date:   Wed Oct 5 19:33:11 2022 +0300
```

my first commit

```
commit 69664c156809ec00eb2d0c0624b88777355d0162
Author: for-you-nastya <nastenka.kravchenko.01@mail.ru>
Date:   Wed Oct 5 14:44:49 2022 +0300
```

First Commit

10. При помощи команды `git diff <имя коммита/файла> <имя коммита/файла>` можно посмотреть различия между двумя коммитами или файлами.

```

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git log --oneline
47e6e48 (HEAD -> vetka, origin/vetka) Commit new vetka
be72de3 (origin/main, main) Second Commit
5566069 my first commit
69664c1 First Commit

пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit/hello (vetka)
$ git diff be72de3 47e6e48
diff --git a/hellogit.html b/hellogit.html
new file mode 100644
index 0000000..4d47978
--- /dev/null
+++ b/hellogit.html
@@ -0,0 +1,18 @@
+<!DOCTYPE html>
+<html>
+<head>
+<title>helloworld</title>
+</head>
+<body>
+Git используется в большинстве компаний, где над проектом работает хотя бы два
разработчика:
+<ol>
+ <li>Новый человек приходит в компанию и клонирует репозиторий проекта на ПК.<
/li>
+ <li>Новый человек приходит в компанию и клонирует репозиторий проекта на ПК.<
/li>
+ <li>Получает задачу, создаёт новую ветку и пишет код.</li>
+ <li>Когда всё готово – отправляет запрос на добавление кода в master-ветку.</
li>
+<li>Когда всё готово – отправляет запрос на добавление кода в master-ветку.</li>
+...skipping...
diff --git a/hellogit.html b/hellogit.html
new file mode 100644
index 0000000..4d47978
--- /dev/null
+++ b/hellogit.html
@@ -0,0 +1,18 @@
+<!DOCTYPE html>
+<html>
+<head>

```

`git diff HEAD` — показать, что изменилось с последнего коммита.

`git diff HEAD^` — показать, что изменилось с предпоследнего коммита.

`git diff [branch]` — сравнить текущую ветку с заданной.

`git diff --stat` — показать статистику какие файлы были изменены и как.

Другой полезной командой является `git blame <файл>`, которая для каждой строки файла показывает автора и контрольную сумму последнего коммита, который изменил эту строку. `-L <начало>, <конец>` позволяет ограничить эту команду заданными строками. Это можно использовать, например, для выяснения того, какой коммит привёл к определённому багу (чтобы можно было его откатить).

11. Следующие команды выполните самостоятельно и посмотрите, что:

`git log --follow [file]` — список изменения текущего файла, включая переименования

Опция `--pretty` меняет формат вывода:

`git log --pretty=oneline` — выводит каждый коммит в одну строку

`git log --pretty=format:"%h %s" --graph` — изменение вида отображения истории изменений

`git log --author='Name' --after={ 1.week.ago } --pretty=oneline --abbrev-commit` — посмотреть над чем работал заданный пользователь последнюю неделю

`git log --no-merges master..` — посмотреть историю изменений только для текущей ветки

`git show [commit]` — показать метадату и изменения в заданном коммите

`git show [branch]:[file]` — посмотреть на файл в другой ветке, не переключаясь на неё

`git log --stat` — показывает сокращённую статистику для коммитов, например изменённые файлы и количество добавленных/удалённых строк в каждом из них

`git log -S имя_функции` — позволяет посмотреть добавление/удаление функции