

Дорожная карта

- 0. Приступаем
- 1. Введение в Git
- 2. Начало работы с Git
- 3. Просмотр истории 🖱️
- 4. Ветвление
- 5. Слияние
- 6. Отмена изменений
- 7. Рабочий процесс

Дорожная карта (продолжение)

- 8. Работа в команде
- 9. Метки
- 10. Последние штрихи
- 11. Завершаем

Что такое коммит

Коммит - это единица изменения исходного кода в репозитории.

Фрагмент заголовка коммита:

```
$ git log -1  
commit a31fd4...  
Author: Maxim Suslov <msuslov@luxoft.com>  
Date: Mon Jul 29 2019 ...
```

Как вычисляется ID :

```
ID = content + author + date + log + previous commit
```

👉 Коммиты никогда не меняются

🤔 Почему для номеров коммитов не используются порядковые номера?

Команда `git log`

Команда `git log` показывает историю коммитов.

Имеются разные опции:

- для поиска коммитов по разным критериям
- для разных способов отображения

```
$ git log [refspec]
```

`refspec` - имя ветки, хэш коммита или тег

Команда `git log` → Отображение

Опция `-p` показывает изменения для каждого коммита:

```
commit 1224679011cf618c0ed51fc16f06334aad3c19d3
Author: Scott Chacon <schacon@gmail.com>
Date:   Thu Sep 22 08:05:09 2011 -0700

    oops. cname goes here

diff --git a/script/publish.rb b/script/publish.rb
index 827626d..4117629 100644
--- a/script/publish.rb
+++ b/script/publish.rb
@@ -1,5 +1,5 @@
 task :publish do
-  File.open("output/CNAME", 'w+') do |f|
+  File.open("output/book/CNAME", 'w+') do |f|
    f.puts("book.git-scm.com")
  end
```

Команда `git log` → Отображение

Опция `--stat` показывает статистику измененных файлов:

```
commit 90634acfcf9b106c9f5ac12bae5d92307cd577b0 (tag: 2019.07.27)
```

```
Author: Sergey M. <dstftw@gmail.com>
```

```
Date: Sat Jul 27 03:44:55 2019 +0700
```

```
release 2019.07.27
```

<code>.github/ISSUE_TEMPLATE/1_broken_site.md</code>	6	+++---
<code>.github/ISSUE_TEMPLATE/2_site_support_request.md</code>	4	++--
<code>.github/ISSUE_TEMPLATE/3_site_feature_request.md</code>	4	++--
<code>.github/ISSUE_TEMPLATE/4_bug_report.md</code>	6	+++---
<code>.github/ISSUE_TEMPLATE/5_feature_request.md</code>	4	++--
<code>ChangeLog</code>	2	+-
<code>docs/supportedsites.md</code>	1	+
<code>youtube_dl/version.py</code>	2	+-
8 files changed, 15 insertions(+), 14 deletions(-)		

Команда `git log` → Отображение

Опция `--shortstat` отображает только строку с количеством изменений/вставок/удалений для команды `--stat` :

```
commit 90634acfcf9b106c9f5ac12bae5d92307cd577b0 (tag: 2019.07.27)
Author: Sergey M. <dstftw@gmail.com>
Date:   Sat Jul 27 03:44:55 2019 +0700

    release 2019.07.27

8 files changed, 15 insertions(+), 14 deletions(-)
```


Команда `git log` → Отображение

Опция `--name-only` показывает список измененных файлов:

```
commit 90634acfcf9b106c9f5ac12bae5d92307cd577b0 (tag: 2019.07.27)
Author: Sergey M. <dstftw@gmail.com>
Date:   Sat Jul 27 03:44:55 2019 +0700

    release 2019.07.27

.github/ISSUE_TEMPLATE/1_broken_site.md
.github/ISSUE_TEMPLATE/2_site_support_request.md
.github/ISSUE_TEMPLATE/3_site_feature_request.md
.github/ISSUE_TEMPLATE/4_bug_report.md
.github/ISSUE_TEMPLATE/5_feature_request.md
ChangeLog
docs/supportedsites.md
youtube_dl/version.py
```

Команда `git log` → Отображение

Опция `--name-status` показывает список файлов, которые добавлены/изменены/удалены:

```
commit c68a03855ba01be97b5bc04179e2dff78b3d4796
Author: Ben Straub <bs@github.com>
Date:   Mon Jan 6 14:17:35 2014 -0800

    Fix markdown H4's (convert into asciidoc H5's)

A       en/book/images/bitnami.png
M       en/book/05-distributed-git/chapter5.asc
D       README.asciidoc
```

Команда `git log` → Отображение

Опция `--abbrev-commit` показывает только несколько символов SHA-1 чек-суммы вместо всех 40:

```
commit 90634acfc (tag: 2019.07.27)
Author: Sergey M. <dstftw@gmail.com>
Date:   Sat Jul 27 03:44:55 2019 +0700
```

```
    release 2019.07.27
```

Команда `git log` → Отображение

Опция `--relative-date` отображает дату в относительном формате (например, `2 weeks ago`) вместо стандартного формата даты:

```
commit 90634acfcf9b106c9f5ac12bae5d92307cd577b0 (tag: 2019.07.27)
Author: Sergey M. <dstftw@gmail.com>
Date:   4 days ago

    release 2019.07.27
```

Команда `git log` → Отображение

Опция `--oneline` показывает только первую строку из сообщения:

```
0e25143 Merge branch 'feature'  
ad8621a Fix a bug in the feature  
16b36c6 Add a new feature  
23ad9ad Add the initial code base
```

Команда `git log` → Отображение

Опция `--decorate` позволяет отображать все ссылки (ветки, метки):

```
0e25143 (HEAD, master) Merge branch 'feature'  
ad8621a (feature) Fix a bug in the feature  
16b36c6 Add a new feature  
23ad9ad (tag: v0.9) Add the initial code base
```

Команда `git log` → Отображение

Опция `--pretty` позволяет определить формат вывода информации о коммитах:

```
$ git log --pretty=format:"%cn committed %h on %cd"  
John committed 400e4b7 on Fri Jun 24 12:30:04 2014 -0500  
John committed 89ab2cf on Thu Jun 23 17:09:42 2014 -0500  
Mary committed 180e223 on Wed Jun 22 17:21:19 2014 -0500  
John committed f12ca28 on Wed Jun 22 13:50:31 2014 -0500
```

Команда `git log` → Построение графа

`--graph` отображает ASCII граф с ветвлениями и историей слияний.

```
$ git log --graph --oneline --decorate
- 0e25143 (HEAD, master) Merge branch 'feature'
| \
| * 16b36c6 Fix a bug in the new feature
| * 23ad9ad Start a new feature
- | ad8621a Fix a critical security issue
| /
- 400e4b7 Fix typos in the documentation
- 160e224 Add the initial code base
```


Практика

1. Переключитесь в репозиторий хотя бы с несколькими коммитами.
2. Выполните команду `git log`.
3. Добавьте следующие опции и посмотрите, как изменится вывод:

- `--stat`
- `--name-only`
- `--shortstat`
- `--name-status`
- `--abbrev-commit`
- `--relative-date`

Выпишите себе команду, которая показывает информацию в наиболее удобном для вас формате.

4. Постройте вывод истории в виде графа `git log --graph --oneline --decorate`

Команда `git log` → Ограничение вывода

Команда `git log` позволяет вывести подмножество коммитов:

- параметр `-n`
- параметры `--since` и `--until` (или `--after` и `--before`)
- параметр `--author`
- параметр `--grep`
- параметр `-S`
- указание диапазона `<from>` до `<to>`

Команда `git log` → Ограничение вывода

Параметр `-<n>` ограничивает количество коммитов на заданное число `n`.

Команда `git log` → Ограничение вывода

Параметры `--since` и `--until` (или их синонимы `--after` и `--before`) ограничивают вывод по времени:

```
$ git log --after=yesterday
$ git log --after=2019-7-1 --before=2019-7-4
$ git log --since=2.weeks.4.days
$ git log --since="2 years 1 day 3 minutes ago"
```

Команда `git log` → Ограничение вывода

Параметр `--author` даёт возможность фильтровать по автору коммита:

```
$ git log --author="John"  
$ git log --author="John\|Mary"  
$ git log --committer="John"
```

Команда `git log` → Ограничение вывода

Параметр `--grep` позволяет искать по словам в сообщении коммита:

```
$ git log --grep="JIRA-224:"  
$ git log --grep=release -i
```

Команда `git log` → Ограничение вывода

Параметр `-S` принимает строку и показывает только те коммиты, в которых изменение в коде повлекло за собой добавление или удаление заданной строки:

```
$ git log -S"Hello, World!"  
$ git log -Sfunction_name  
$ git log -G"<regex>"
```

Команда `git log` → Ограничение вывода

Фильтрация по имени файла или папки:

```
git log -- foo.py bar.py
```


Команда `git log` → Ограничение вывода

Ограничение по диапазону от `<from>` до `<to>`:

```
$ git log <from>..<to>
```

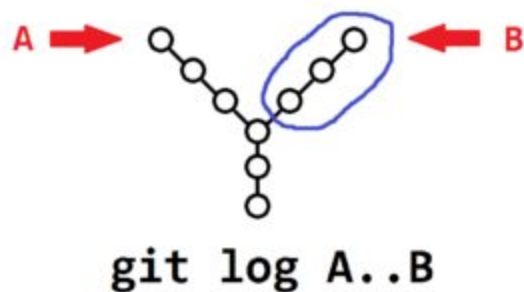
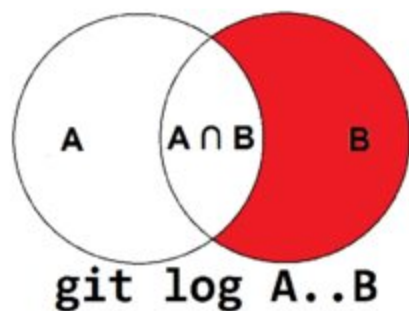
Практика

1. Передейдите в репозиторий хотя бы с несколькими коммитами.
2. Совместите и используйте опции `-n` и `--oneline` .
3. Посмотрите коммиты за последние три недели, используя `--since` .
4. Выберите среди них любого автора и посмотрите его коммиты, используя `--author` .
5. Сделайте поиск коммитов, используя `--grep` .

Где начинается ветка?

🤔 Мы знаем, где ветка заканчивается, но не где она была создана. Почему?

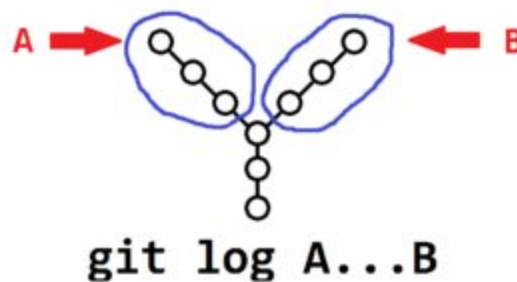
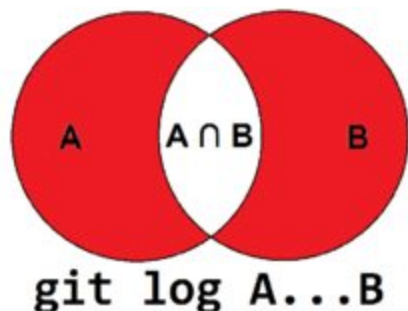
Показ коммитов только в ветке



Если в качестве **A** или **B** используется текущая ветка, то имя можно опустить:

```
$ git log master..origin/master  
$ git log ..origin/master # the same
```

Показ коммитов только в ветке



```
$ git log --oneline --decorate --left-right --graph master...origin/master
< 1794bee (HEAD, master) Derp some more
> 6e6ce69 (origin/master, origin/HEAD) Add hello.txt
```

Практика

1. Сделайте новую ветку, которая находится на 2 коммита назад от верхушки `master` :

```
$ git branch oldmaster master~2  
$ git checkout oldmaster
```

2. Создайте на ней несколько коммитов
3. Посмотрите на вывод команды `git log` с двумя (`..`) точками
4. Посмотрите на вывод команды `git log` с тремя (`...`) точками

Псевдонимы (alias)

Git позволяет определять псевдонимы (alias), чтобы сократить вводимые команды:

```
$ git config --global alias.<name> "<command>"
$ cat ~/.gitconfig
[alias]
  <name> = <command>
```

Например:

```
$ git config --global alias.l "log --pretty=oneline --graph --abbrev-commit --decorate=auto"

$ git l -4
* 34b9a04c7 (HEAD -> main, origin/main, origin/HEAD) \
Merge pull request #5832 from ianhattendorf/fix/repo-is-empty-empty-config-defaultbranch
|\
| * 7891e1537 Default to GIT_BRANCH_DEFAULT if init.defaultBranch is empty string
|/
* 508361401 Merge pull request #5821 from punkymaniac/example-duplicate-line
|\
| * 8f52b0c55 Remove duplicate line, in example code
|/
```

Практика

1. Создайте псевдоним `st` , который является сокращением для команды `status` .
2. Создайте псевдоним `l` , который вызывает следующую команду `log --stat --decorate --abbrev-commit` .
3. Выполните несколько команд, используя эти псевдонимы.

Заключение

Делайте коммиты чаще!

- То, что попало в репозиторий, очень трудно потерять.
- В случае ненужных и/или неправильных изменений всегда можно откатиться назад к работающему состоянию.

Команда `git log` - основной инструмент для работы с историей.

- Она имеет параметры для изменения способа отображения и
- для изменения набора коммитов

Ваши вопросы ?