

**NFE** Networking  
For  
Everyone

**GIT**



# Системы контроля версий (СКВ)

Что:

- Система, регистрирующая изменения в одном или нескольких файлах с возможностью вернуться к определенным старым версиям этих файлов

Зачем:

- Помогает отследить всю историю изменений файлов
- Упрощает поиск изменений
- Дает возможность отката определенных изменений в проекте
- Упрощает и контролирует работу в команде, гарантирует отсутствие перезаписи и дублирования

Существует много различных систем контроля версий: SVN, Mercurial, Perforce, CVS, Bitkeeper. GIT (Global Information Tracker) - наиболее распространенная.

# Виды СКВ

## Локальная (SCCS, RCS)

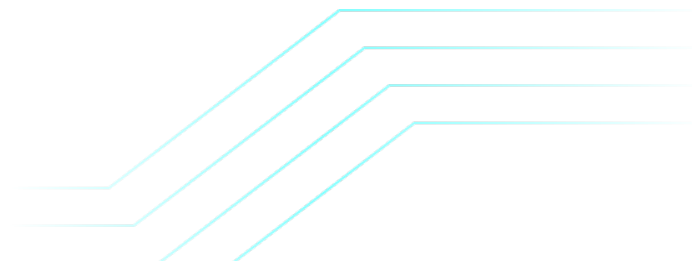
- Хранит информацию о всех изменениях в базе данных, которая находится там же локально на компьютере.
  - Предоставляет нужный функционал с высокой скоростью.
  - Единая точка отказа
  - Невозможность совместной разработки

## Централизованная (CVS, SVN)

- Есть удаленный центральный репозиторий и разработчики загружают копию файлов, вносят изменений и выгружают измененные файлы обратно
  - Возможность распределенной разработки
  - Полный контроль администратором

## Децентрализованная (Git)

- Нет центрального репозитория, все копии считаются равными
- Возможность работы без подключения к репозиторию



# Установка GIT

<https://git-scm.com/>

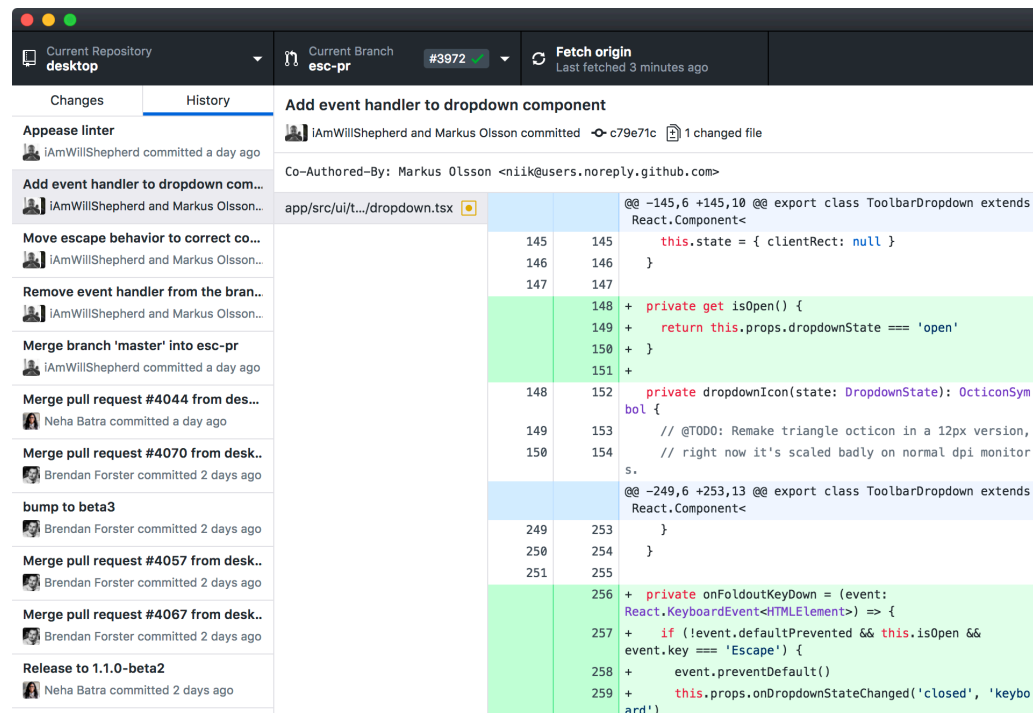
```
~ % git --version
git version 2.37.1 (Apple Git-137.1)

~ % git config --global user.name alexigna
~ % git config --global user.email ignatov.alx@gmail.com

~ % git config --list | grep user
user.name=alexigna
user.email=ignatov.alx@gmail.com
```

```
alexigna@mbp-alexigna 01.init % git config --list --show-origin
file:/Library/Developer/CommandLineTools/usr/share/git-core/gitconfig
file:/Library/Developer/CommandLineTools/usr/share/git-core/gitconfig
file:/Users/alexigna/.gitconfig user.name=alexigna
file:/Users/alexigna/.gitconfig user.email=ignatov.alx@gmail.com
file:/Users/alexigna/.gitconfig init.defaultbranch=master
file:./git/config core.repositoryformatversion=0
file:./git/config core.filemode=true
file:./git/config core.bare=false
file:./git/config core.logallrefupdates=true
file:./git/config core.ignorecase=true
file:./git/config core.precomposeunicode=true
```

```
credential.helper=osxkeychain
init.defaultbranch=main
```



# Создание репозитория

```
alexigna@mbp-alexigna 01.init % git init
Initialized empty Git repository in /Users/alexigna/Documents/n4e devnet/n4e_git/01.init/.git/

alexigna@mbp-alexigna 01.init % git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

alexigna@mbp-alexigna 01.init % echo "version 1" > f1.txt
alexigna@mbp-alexigna 01.init % git add f1.txt
alexigna@mbp-alexigna 01.init % git commit -m "init"
[master (root-commit) 36eacd7] init
1 file changed, 1 insertion(+)
create mode 100644 f1.txt
```

git init - инициализация репозитория

git status - текущее состояние репозитория

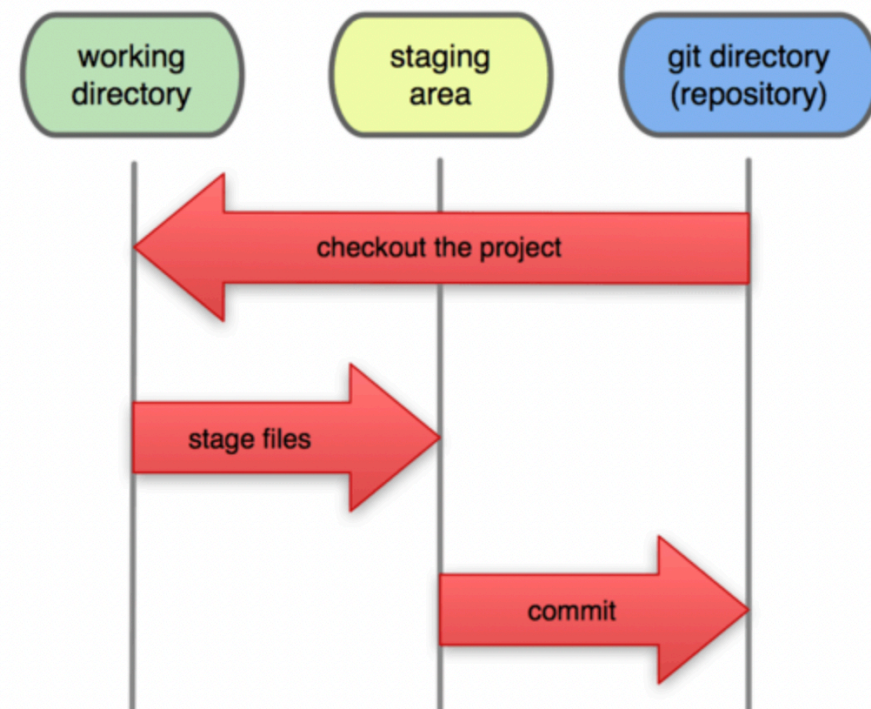
git add - добавляем файл в stage

git commit - создание commit, фиксация состояния

# Состояния git

Три основных состояния, в которых могут находиться файлы репозитория:

- Зафиксированное - файл уже сохранён в вашей локальной базе
  - Измененное - файл поменялся, но ещё не был зафиксирован
  - Подготовленное - измененный файл, отмеченный для включения в следующий commit
- 
- Рабочая директория - содержит то, с чем сейчас идет работа
  - Директория Git - репозиторий, в котором сохранена вся история проекта
  - Область подготовленных файлов - файл, хранящийся в каталоге Git и содержащий информацию о том, что должно войти в следующий commit



# Объектная модель Git

## SHA:

Вся информация для предоставления истории проекта хранится в специально организованных объектах. Все объекты ссылаются друг на друга с помощью SHA хешей (40 символов).

- git определяет идентичность файлов сравнивая их хеш
- имена объектов вычисляются одинаково, поэтому во всех репозитория одинаковые объекты будут храниться с одинаковыми именами
- git может определить ошибки просто считав файл, вычислив его хеш и сравнив его с именем файла

**Объекты**, характеризуются типом, размером, содержимым:

- blob - используется для хранения содержимого файла, обычно это просто файл
- tree - аналог директории, ссылается на другие trees и blobs
- commit - указывает на определенное tree, отмечая его и фиксируя в истории
- tag - способ особым образом отменить commit, например версия релиза

# Папка .git

HEAD - указатель на верхушку ветки, с которой мы работаем

config - настройки текущего репозитория

description - описание репозитория для gitweb

hooks - содержит скрипты, которые могут выполняться на разных этапах работы с git

info/exclude - описанные в файле исключения не будут попадать в репозиторий

```
alexigna@mbp-alexigna 01.init % cat .git/HEAD
ref: refs/heads/master
```

```
alexigna@mbp-alexigna 01.init % cat .git/refs/heads/master
36eacd763b5b5d0559b54b9909ab85bd25208d56
```

```
|— HEAD
|— branches
|— config
|— description
|— hooks
|   |— pre-commit.sample
|   |— pre-push.sample
|   └─ ...
|— info
|   └─ exclude
|— objects
|   |— info
|   └─ pack
└─ refs
    |— heads
    └─ tags
```



# blob

blob обычно хранит содержимое файла

```
alexigna@mbp-alexigna 01.init % echo "version 1" > f1.txt
alexigna@mbp-alexigna 01.init % echo "version 1" > f2.txt

alexigna@mbp-alexigna 01.init % git hash-object -w f1.txt
83baae61804e65cc73a7201a7252750c76066a30

alexigna@mbp-alexigna 01.init % git hash-object -w f2.txt
83baae61804e65cc73a7201a7252750c76066a30

alexigna@mbp-alexigna 01.init % echo "version 1" | git hash-object --stdin
83baae61804e65cc73a7201a7252750c76066a30

alexigna@mbp-alexigna 01.init % ll .git/objects/83
total 8
-r--r--r--  1 alexigna  staff  26 Jul 20 22:06 baae61804e65cc73a7201a7252750c76066a30

alexigna@mbp-alexigna 01.init % git cat-file -p 83ba
version 1
```

5b1d3..

blob	size
<pre>#ifndef REVISION_H #define REVISION_H  #include "parse-options.h"  #define SEEN (1u&lt;&lt;0) #define UNINTERESTING (1u #define TREESAME (1u&lt;&lt;2)</pre>	

# tree

tree это объект который включает в себе группу указателей на blobs и другие trees.

```
alexigna@mbp-alexigna 01.init % git cat-file -p da10bc
100644 blob 83baae61804e65cc73a7201a7252750c76066a30 f1.txt
```

SHA значения двух деревьев будет совпадать если если их содержимое (рекурсивно включая содержимое всех поддиректорий) идентично.

c36d4..

tree		size
blob	5b1d3	README
tree	03e78	lib
tree	cdc8b	test
blob	cba0a	test.rb
blob	911e7	xdiff

# commit

Сохраняет состояние дерева с добавлением мета-данных и описанием.

```
alexigna@mbp-alexigna 01.init % echo "version 1" > f2.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "add f2"
[master 7131287] add f2
 1 file changed, 1 insertion(+)
 create mode 100644 f2.txt

alexigna@mbp-alexigna 01.init % git cat-file -p 7131287
tree 68a3fb63d989b30cb58fc8c5474b01b00985f4a2
parent 36eacd763b5b5d0559b54b9909ab85bd25208d56
author alexigna <ignatov.alx@gmail.com> 1689875726 +0500
committer alexigna <ignatov.alx@gmail.com> 1689875726 +0500

add f2
```

tree - sha объекта tree

parent - sha предыдущего объекта commit (родитель)

commit сам по себе не содержит информации о том, что изменилось, изменения вычисляются путем сравнения текущего tree и tree на которой ссылается родитель

ae668..

commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		

# tag

Сохраняет состояние дерева с добавлением мета-данных и описанием.

```
alexigna@mbp-alexigna 01.init % git tag -a v1.0 -m "version 1.0"

alexigna@mbp-alexigna 01.init % cat .git/refs/tags/v1.0
124caab8e28c81464e42d729070c2b027ef81766

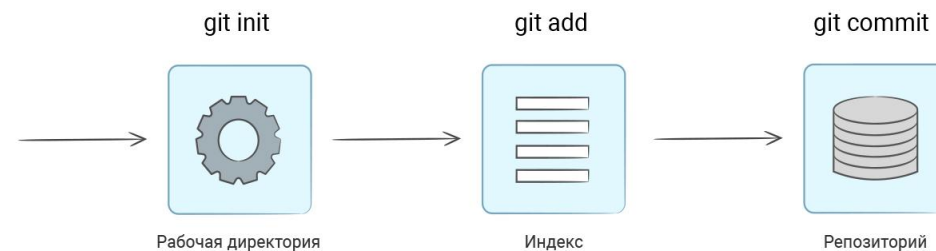
alexigna@mbp-alexigna 01.init % git cat-file -p 124caa
object 7131287224a2109cde671a162fb083b65844aeb8
type commit
tag v1.0
tagger alexigna <ignatov.alx@gmail.com> 1689876331 +0500
```

49e11..

tag		size
object	ae668	
type	commit	
tagger	Scott	
my tag message that explains this tag		

# Стандартный рабочий процесс

- Внесение изменений в файлы в рабочей области
- Добавление файлов в индекс (staged area)
- Создание commit, файлы из staged area сохраняются в репозиторий



# Стандартный рабочий процесс

```
alexigna@mbp-alexigna 01.init % git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f3.txt

nothing added to commit but untracked files present (use "git add" to track)

alexigna@mbp-alexigna 01.init % git add .

alexigna@mbp-alexigna 01.init % git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f3.txt

alexigna@mbp-alexigna 01.init % git commit -m "new f3 file"
[master 9418473] new f3 file
1 file changed, 1 insertion(+)
create mode 100644 f3.txt
```

`git status` - текущее состояние репозитория, просмотр индекса

`git add` - добавляем файл в stage area

`git commit` - создание commit, фиксация состояния (сохранение в репозиторий)

# commit

```
alexigna@mbp-alexigna 01.init % git log --oneline
cc567f7 (HEAD -> master) change f2
7131287 (tag: v1.0) add f2
36eacd7 init

alexigna@mbp-alexigna 01.init % git reset HEAD~3
Unstaged changes after reset:
M   f1.txt
M   f2.txt

alexigna@mbp-alexigna 01.init % git commit --amend -m "updated commit"
[master 57a8b32] updated commit
```

`git reset` - сбрасывает состояние  
в указанную точку

`git commit --amend` -  
перезапись последнего commit

# log

```
alexigna@mbp-alexigna 01.init % git log --oneline --since=10.minutes
79df0e8 (HEAD -> master) some changes
```

```
alexigna@mbp-alexigna 01.init % git log -p -1 --relative-date
commit 79df0e8a3cede19740af73404cbaa4394ac9f16f (HEAD -> master)
Author: alexigna <ignatov.alx@gmail.com>
Date: 12 seconds ago
```

some changes

```
diff --git a/f1.txt b/f1.txt
index d00491f..1f7a7a4 100644
--- a/f1.txt
+++ b/f1.txt
@@ -1,1 @@
-1
+version 2
```

```
alexigna@mbp-alexigna 01.init % git reflog --since=30.minutes
79df0e8 (HEAD -> master) HEAD@{0}: commit: some changes
e4628f4 HEAD@{1}: commit: c5
57a8b32 HEAD@{2}: reset: moving to HEAD~2
87b7499 HEAD@{3}: commit: c2
f74e3ae HEAD@{4}: commit: c1
57a8b32 HEAD@{5}: reset: moving to 57a8b32
b3a6eaa HEAD@{6}: commit: changed
97d5629 HEAD@{7}: commit: test
57a8b32 HEAD@{8}: commit (amend): updated commit
```

--until="Apr 4 2023"  
--since="Apr 1 2023"  
--since="20 minutes ago"

-p -3 - показывает патч (изменения)  
последних 3 commits

reflog - история действий в  
репозитории



# Ветвление

- Ветка (branch) - набор объектов commits, которые идут друг за другом (история commit).  
Указатель на определенный commit
- У каждой ветки есть название
- Новая ветка – реализация нового функционала, исправление багов и пр.
- Создание ветки = создание нового указателя

```
alexigna@mbp-alexigna 01.init % git branch feature/fix-031

alexigna@mbp-alexigna 01.init % git branch
feature/fix-031
* master

alexigna@mbp-alexigna 01.init % cat .git/refs/heads/feature/fix-031
79df0e8a3ced19740af73404cbaa4394ac9f16f

alexigna@mbp-alexigna 01.init % cat .git/refs/heads/master
79df0e8a3ced19740af73404cbaa4394ac9f16f

alexigna@mbp-alexigna 01.init % git cat-file -t 79df0e8a3ced19740af73404cbaa4394ac9f16f
commit
```

# Ветвление

```
alexigna@mbp-alexigna 01.init % git checkout feature/fix-031
Switched to branch 'feature/fix-031'

alexigna@mbp-alexigna 01.init % git switch feature/fix-031
Already on 'feature/fix-031'

alexigna@mbp-alexigna 01.init % git branch
* feature/fix-031
  master

alexigna@mbp-alexigna 01.init % cat .git/HEAD
ref: refs/heads/feature/fix-031

alexigna@mbp-alexigna 01.init % git checkout -b feature/fix-192
Switched to a new branch 'feature/fix-192'

alexigna@mbp-alexigna 01.init % git branch
  feature/fix-031
* feature/fix-192
  master

alexigna@mbp-alexigna 01.init % cat .git/HEAD
ref: refs/heads/feature/fix-192
```

checkout - переключает между ветками,  
обновляется HEAD

switch - новая команда

checkout -b - создает новую ветку и сразу  
переключает на нее

branch - список локальных веток

# Ветвление

```
alexigna@mbp-alexigna 01.init % echo "3\nfix this line\n1\n" > f2.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "fix f2 issue"
[feature/fix-192 c37d3e0] fix f2 issue
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
alexigna@mbp-alexigna 01.init % git cat-file -p c37d3e0
tree 532f0a96f24379d1023e0243d9e154bb6d25d238
parent 2f993f56b2b0af9c51482ebabb131d6af8d0cbcd
author alexigna <ignatov.alx@gmail.com> 1689935801 +0500
committer alexigna <ignatov.alx@gmail.com> 1689935801 +0500
```

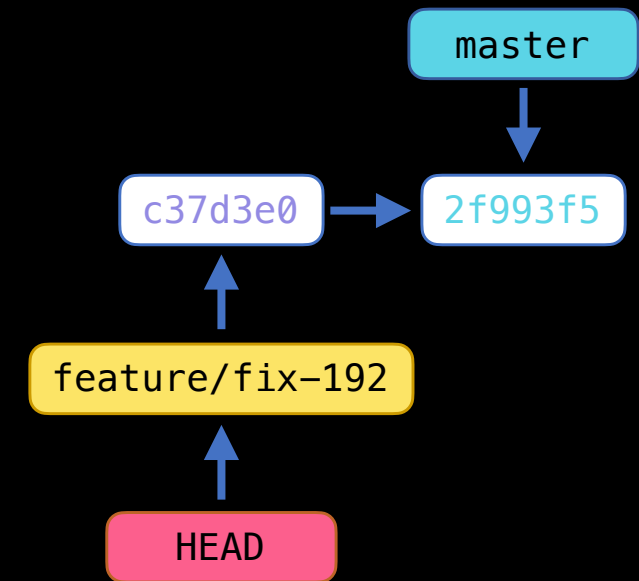
fix f2 issue

```
alexigna@mbp-alexigna 01.init % git cat-file -p 2f993f5
tree 5c4ce2926ee3c157ed629dc094528926ee3e
parent bb03e7657bb8244b0d9d39f70412db1a0e514cbd
author alexigna <ignatov.alx@gmail.com> 1689935731 +0500
committer alexigna <ignatov.alx@gmail.com> 1689935731 +0500
```

third file

```
alexigna@mbp-alexigna 01.init % git cat-file -p 5c4ce2926ee3c157ed629dc094528926ee3e
100644 blob d70f6df361f511cfd354513f3eadf7f2fec135af    f1.txt
100644 blob 8fc96c1977e8044b9457ddfad36d28b74432676e    f2.txt
100644 blob 7b57bd29ea8afbdeb9bac64cf7074f4b531492a8    f3.txt
```

```
alexigna@mbp-alexigna 01.init % git cat-file -p 8fc96c1977e8044b9457ddfad36d28b74432676e
3
2
1
```



# Ветвление

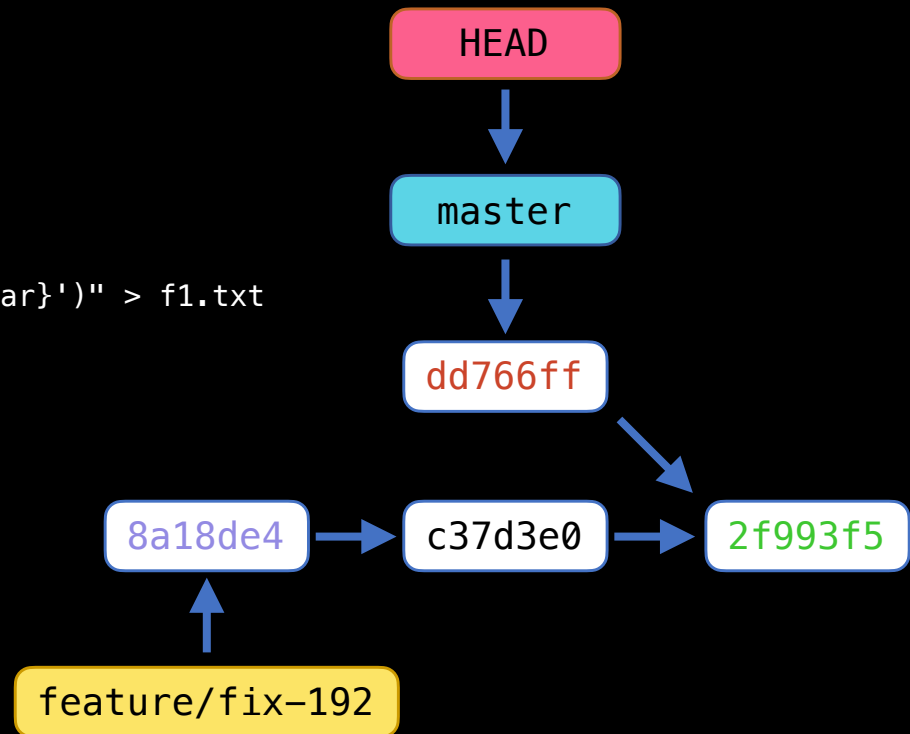
```
alexigna@mbp-alexigna 01.init % echo "new code" > f2.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "refactor f2"
[feature/fix-192 8a18de4] refactor f2
1 file changed, 1 insertion(+), 4 deletions(-)
```

```
alexigna@mbp-alexigna 01.init % git checkout master
Switched to branch 'master'
```

```
alexigna@mbp-alexigna 01.init % echo "var = 'hello'\nprint(f'hello {var}')" > f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "add python"
[master dd766ff] add python
1 file changed, 2 insertions(+), 4 deletions(-)
```

```
alexigna@mbp-alexigna 01.init % git cat-file -p dd766ff
tree 426c6a76061ca780632229aab49d8b7d31ee608d
parent 2f993f56b2b0af9c51482ebabb131d6af8d0cbcd
author alexigna <ignatov.alx@gmail.com> 1689936514 +0500
committer alexigna <ignatov.alx@gmail.com> 1689936514 +0500
```

```
add python
```



# Слияние

```
alexigna@mbp-alexigna 01.init % git merge feature/fix-192 -m "fix-192 to prod"
```

```
Merge made by the 'ort' strategy.
```

```
f2.txt | 5 +—
```

```
1 file changed, 1 insertion(+), 4 deletions(-)
```

```
alexigna@mbp-alexigna 01.init % cat .git/refs/heads/master
```

```
6351e8eb9918f52aa66a14b886813b91324bda35
```

```
alexigna@mbp-alexigna 01.init % git cat-file -t 6351e8e
```

```
commit
```

```
alexigna@mbp-alexigna 01.init % git cat-file -p 6351e8e
```

```
tree dc6f34d78c669ed2199a98a36329251767ba95d8
```

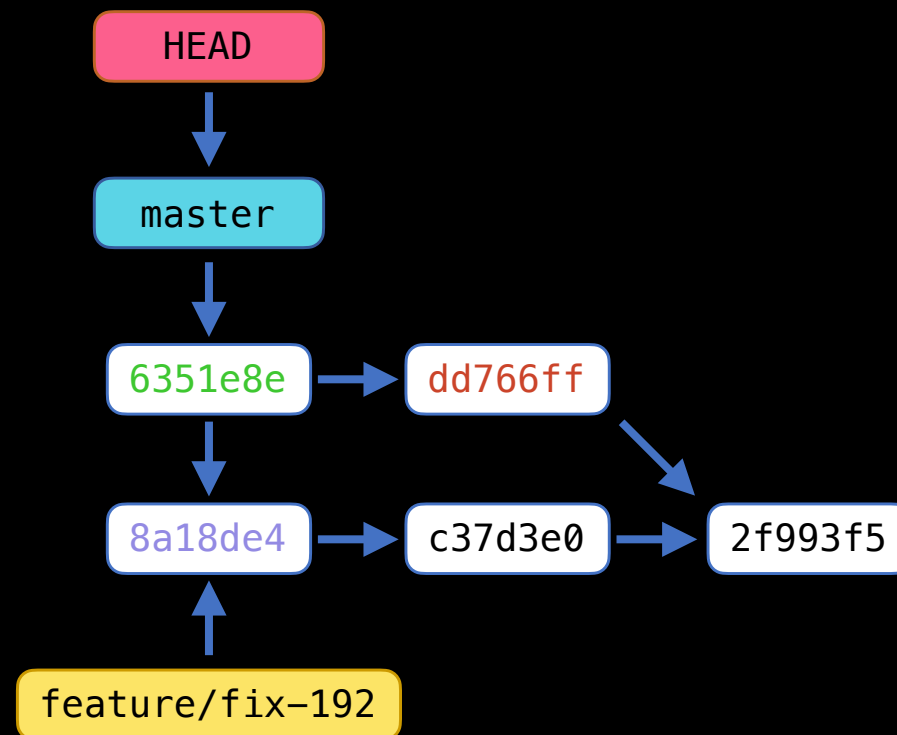
```
parent dd766ffaaf0b6c0e6f445cb378bd035aa2893d86
```

```
parent 8a18de46ab0bd13e39a22d198472607dbdf3cd0f
```

```
author alexigna <ignatov.alx@gmail.com> 1689936885 +0500
```

```
committer alexigna <ignatov.alx@gmail.com> 1689936885 +0500
```

```
fix-192 to prod
```



# Слияние с конфликтом

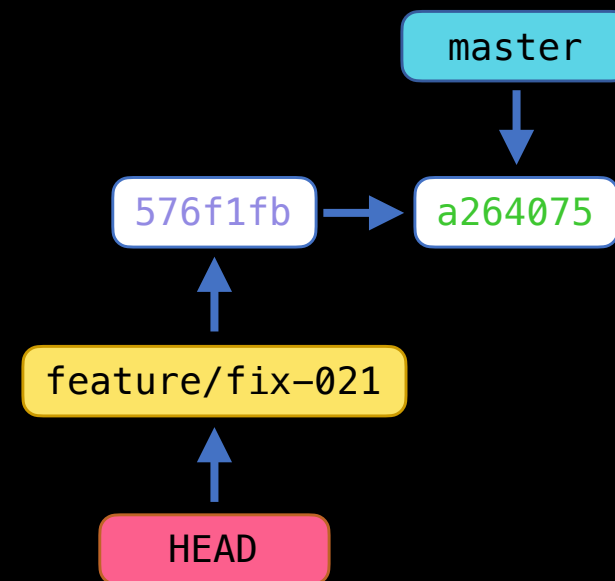
```
alexigna@mbp-alexigna 01.init % echo "1\n2\n3" > f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m init
[master (root-commit) a264075] init
1 file changed, 3 insertions(+)
create mode 100644 f1.txt

alexigna@mbp-alexigna 01.init % git cat-file -p a264075
tree 808a6e6f622e2170dceacc66374c10efeeba0d34
author alexigna <ignatov.alx@gmail.com> 1689942201 +0500
committer alexigna <ignatov.alx@gmail.com> 1689942201 +0500

init

alexigna@mbp-alexigna 01.init % git checkout -b feature/fix-021
Switched to a new branch 'feature/fix-021'

alexigna@mbp-alexigna 01.init % echo "1\nline from fix\n3" > f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "fix some issue"
[feature/fix-021 576f1fb] fix some issue
1 file changed, 1 insertion(+), 1 deletion(-)
```



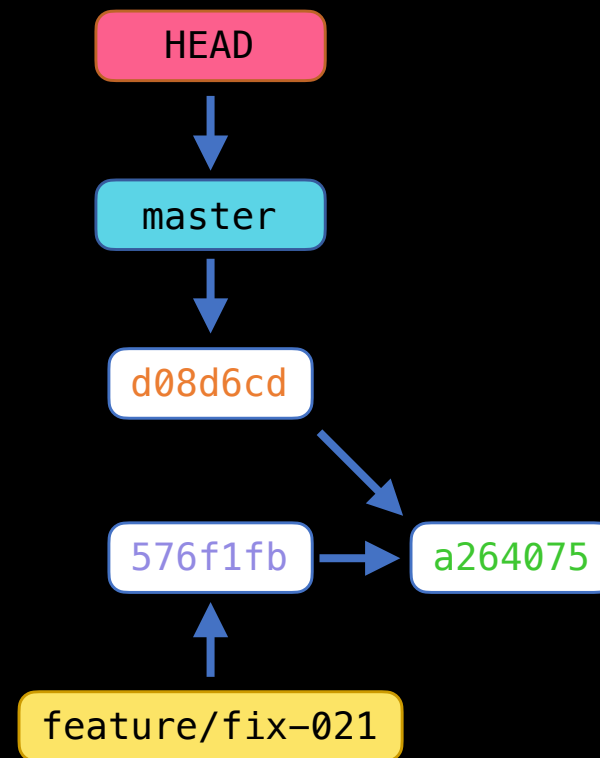
# Слияние с конфликтом

```
alexigna@mbp-alexigna 01.init % git checkout master
Switched to branch 'master'

alexigna@mbp-alexigna 01.init % echo "line from master\n2\n3" > f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "change from master"
[master d08d6cd] change from master
1 file changed, 1 insertion(+), 1 deletion(-)

alexigna@mbp-alexigna 01.init % git cat-file -p d08d6cd
tree 13eaa0f8f5ef7e378185a39f152a574c618646e2
parent a264075b24f573abc30d02c589cf01705835ed3a
author alexigna <ignatov.alx@gmail.com> 1689942690 +0500
committer alexigna <ignatov.alx@gmail.com> 1689942690 +0500

change from master
```

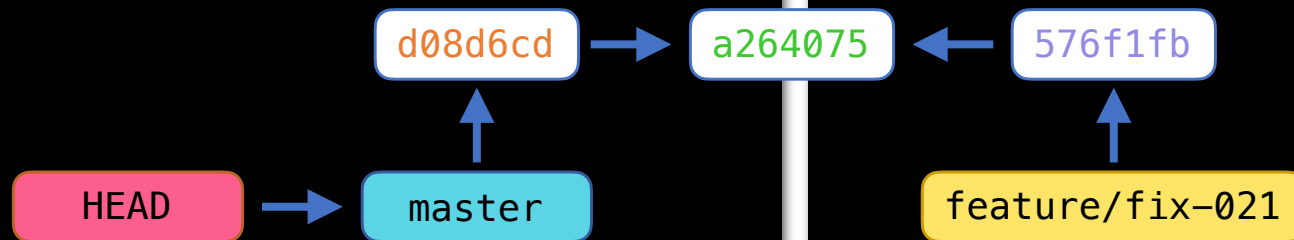


# Слияние с конфликтом

```
alexigna@mbp-alexigna 01.init % git cat-file -p d08d6cd
tree 13eaa0f8f5ef7e378185a39f152a574c618646e2
parent a264075b24f573abc30d02c589cf01705835ed3a
author alexigna <ignatov.alx@gmail.com> 1689942690 +0500
committer alexigna <ignatov.alx@gmail.com> 1689942690 +0500
```

change from master

```
alexigna@mbp-alexigna 01.init % git cat-file -p 13eaa0f
100644 blob 6e22879af6ea086d3a3ab0f6bf2711f8d4f7b048 f1.txt
alexigna@mbp-alexigna 01.init % git cat-file -p 6e22879
line from master
2
3
```



```
alexigna@mbp-alexigna 01.init % git cat-file -p 576f1fb
tree 965d632ba4c8ac0b782934cf26e7f2c18623dd0b
parent a264075b24f573abc30d02c589cf01705835ed3a
author alexigna <ignatov.alx@gmail.com> 1689942519 +0500
committer alexigna <ignatov.alx@gmail.com> 1689942519 +0500
```

fix some issue

```
alexigna@mbp-alexigna 01.init % git cat-file -p 965d632
100644 blob ba52f905483b1dec70138c855fb9f554f22ce8c7 f1.txt
alexigna@mbp-alexigna 01.init % git cat-file -p ba52f90
1
line from fix
3
```



# Слияние с конфликтом

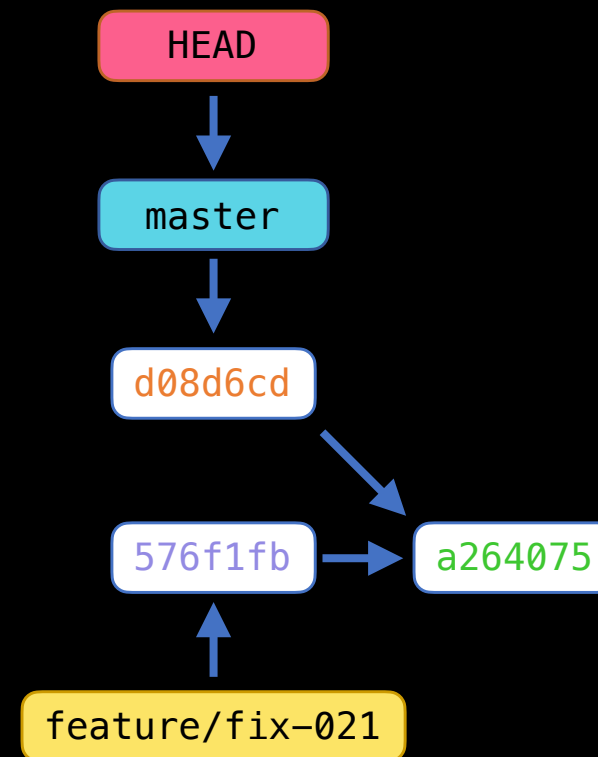
```
alexigna@mbp-alexigna 01.init % git merge feature/fix-021
Auto-merging f1.txt
CONFLICT (content): Merge conflict in f1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
alexigna@mbp-alexigna 01.init % git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
```

```
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   f1.txt
```

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

```
alexigna@mbp-alexigna 01.init % cat f1.txt
<<<<<<< HEAD
line from master
2
=====
1
line from fix
>>>>>>> feature/fix-021
3
```



# Слияние с конфликтом

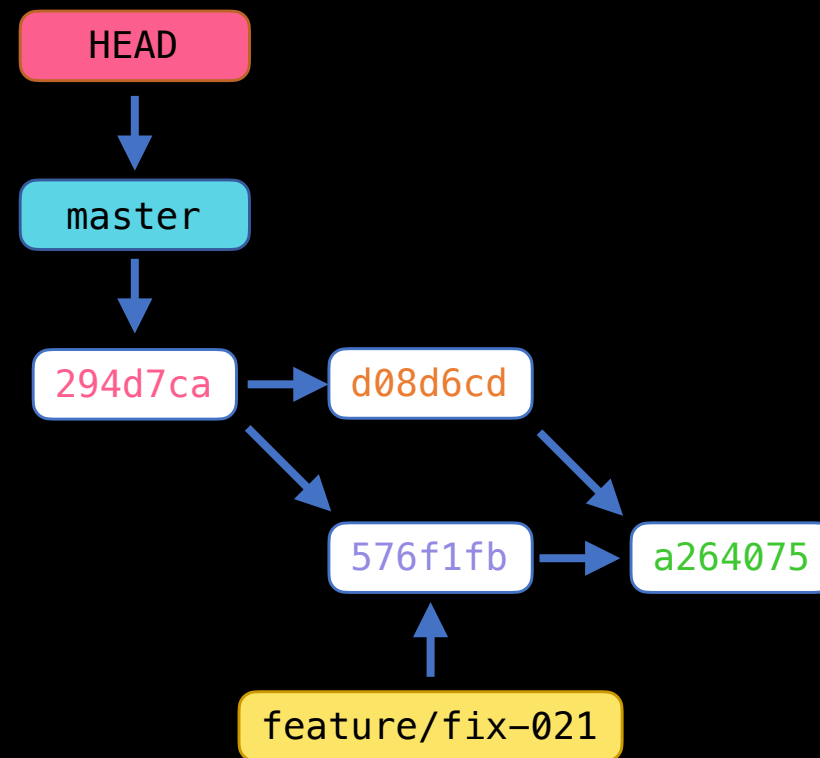
```
alexigna@mbp-alexigna 01.init % cat f1.txt
line from master
line from fix
3
alexigna@mbp-alexigna 01.init % git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   f1.txt

no changes added to commit (use "git add" and/or "git commit -a")
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   f1.txt

alexigna@mbp-alexigna 01.init % git commit -m "merged"
[master 294d7ca] merged
```



# squash

```
alexigna@mbp-alexigna 01.init % git checkout -b feature/fix-312
Switched to a new branch 'feature/fix-312'
```

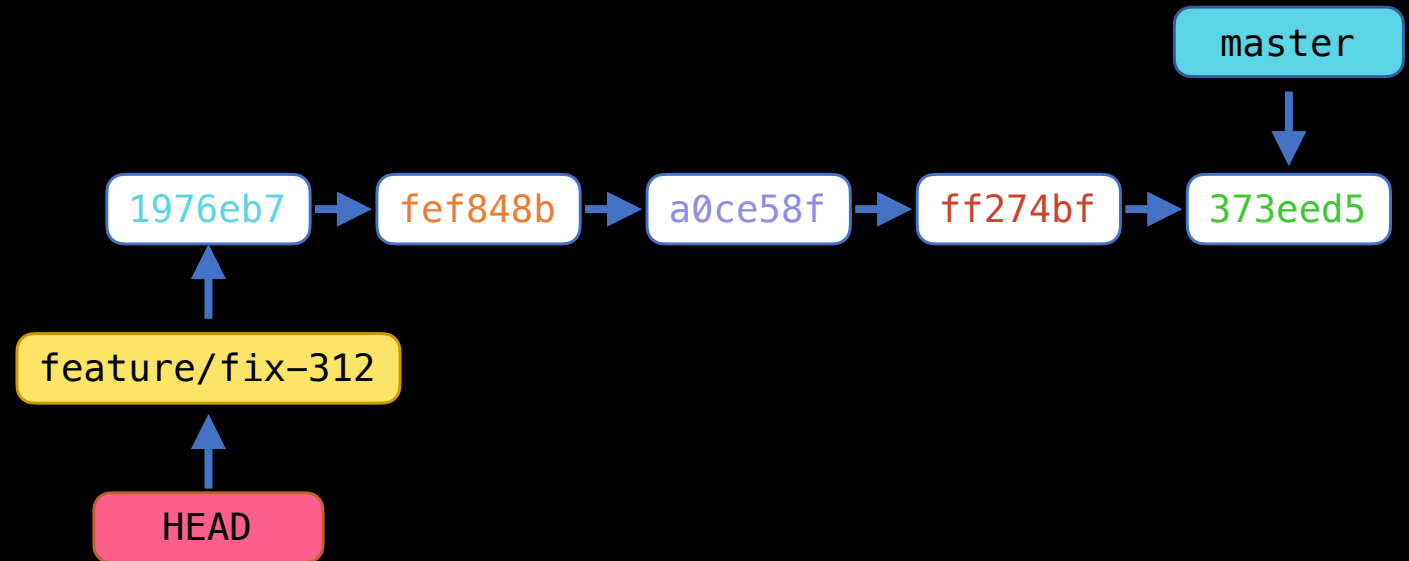
```
alexigna@mbp-alexigna 01.init % echo 1 > f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "change to 1"
[feature/fix-312 ff274bf] change to 1
1 file changed, 1 insertion(+), 3 deletions(-)
```

```
alexigna@mbp-alexigna 01.init % echo 2 >> f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "add 2"
[feature/fix-312 a0ce58f] add 2
1 file changed, 1 insertion(+)
```

```
alexigna@mbp-alexigna 01.init % echo 3 >> f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "add 3"
[feature/fix-312 fef848b] add 3
1 file changed, 1 insertion(+)
```

```
alexigna@mbp-alexigna 01.init % echo 4 >> f1.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "add 4"
[feature/fix-312 1976eb7] add 4
1 file changed, 1 insertion(+)
```

```
alexigna@mbp-alexigna 01.init % git log --oneline
1976eb7 (HEAD -> feature/fix-312) add 4
fef848b add 3
a0ce58f add 2
ff274bf change to 1
373eed5 (master) custom merge
```



# squash

```
alexigna@mbp-alexigna 01.init % git checkout master
Switched to branch 'master'

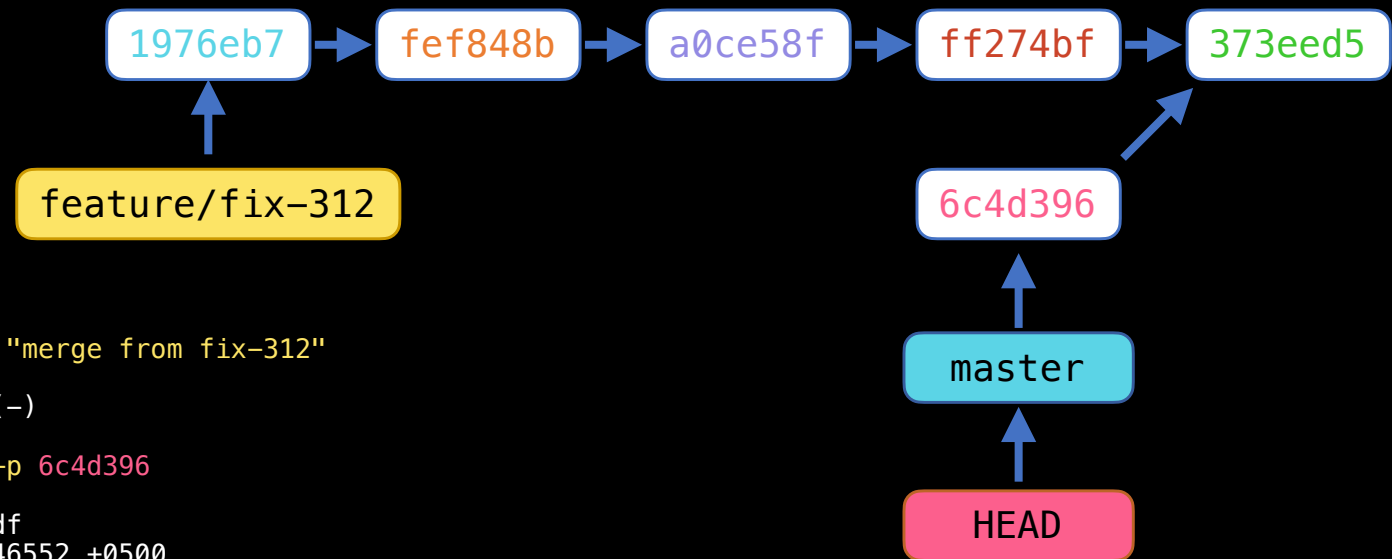
alexigna@mbp-alexigna 01.init % git merge feature/fix-312 --squash
Updating 373eed5..1976eb7
Fast-forward
Squash commit -- not updating HEAD
 f1.txt | 5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)

alexigna@mbp-alexigna 01.init % git diff --staged
diff --git a/f1.txt b/f1.txt
index f4e28ef..94ebaf9 100644
--- a/f1.txt
+++ b/f1.txt
@@ -1,3 +1,4 @@
-line from master-2
-line from fix
+1
+2
 3
+4

alexigna@mbp-alexigna 01.init % git commit -m "merge from fix-312"
[master 6c4d396] merge from fix-312
 1 file changed, 3 insertions(+), 2 deletions(-)

alexigna@mbp-alexigna 01.init % git cat-file -p 6c4d396
tree 4de91fe197e449deab8f2e6f1a4fe8d7ec2820c4
parent 373eed527d080ae65db507f56aff00d2570a93df
author alexigna <ignatov.alx@gmail.com> 1689946552 +0500
committer alexigna <ignatov.alx@gmail.com> 1689946552 +0500

merge from fix-312
```



# log

```
alexigna@mbp-alexigna 01.init % git log v1.2.. --oneline
6c4d396 (HEAD -> master) merge from fix-312
373eed5 custom merge
c82d0b6 new changes from master

alexigna@mbp-alexigna 01.init % git log master..feature/fix-312 --oneline
1976eb7 (feature/fix-312) add 4
fef848b add 3
a0ce58f add 2
ff274bf change to 1

alexigna@mbp-alexigna 01.init % git log --oneline --no-merges f1.txt
6c4d396 (HEAD -> master) merge from fix-312
c82d0b6 new changes from master
d08d6cd change from master
576f1fb fix some issue
a264075 init

alexigna@mbp-alexigna 01.init % git log --graph --oneline f1.txt
* 6c4d396 (HEAD -> master) merge from fix-312
* c82d0b6 new changes from master
* 294d7ca merged
| \
| * 576f1fb fix some issue
* | d08d6cd change from master
| /
* a264075 init
```

```
alexigna@mbp-alexigna 01.init % git log --oneline
6c4d396 (HEAD -> master) merge from fix-312
373eed5 custom merge
c82d0b6 new changes from master
afdb1e7 (tag: v1.2) new changes
294d7ca merged
d08d6cd change from master
576f1fb fix some issue
a264075 init
```

v1.2.. - КОММИТЫ НАЧИНАЯ С ТЕГА v1.2

master..feature/fix-312 -  
КОММИТЫ ИЗ ВЕТКИ feature/fix-312, НО НЕ  
master

--no-merges - НЕ ПОКАЗЫВАТЬ КОММИТЫ  
СЛИЯНИЯ

--graph - графическое представление

# diff

```
alexigna@mbp-alexigna 01.init % git diff master..feature/fix-093 f1.txt
diff --git a/f1.txt b/f1.txt
index 94ebaf9..00750ed 100644
--- a/f1.txt
+++ b/f1.txt
@@ -1,4 +1 @@
-1
-2
3
-4

alexigna@mbp-alexigna 01.init % echo "new text" > f1.txt

alexigna@mbp-alexigna 01.init % git diff --cached

alexigna@mbp-alexigna 01.init % git diff HEAD
diff --git a/f1.txt b/f1.txt
index 94ebaf9..eee417f 100644
--- a/f1.txt
+++ b/f1.txt
@@ -1,4 +1 @@
-1
-2
-3
-4
+new text

alexigna@mbp-alexigna 01.init % git diff --stat
f1.txt | 5 +----
1 file changed, 1 insertion(+), 4 deletions(-)
```

master..feature/fix-093 f1.txt -  
разница файла f1.txt между двумя ветками

diff --cached - разница между stage area и  
репозиторием (то, что попадает в репозиторий  
при commit)

diff HEAD - разница между рабочей  
директории и репозиторием

diff --stat - краткая статистика изменений

# rebase

```
alexigna@mbp-alexigna 01.init % git checkout -b feature/fix-159
Switched to a new branch 'feature/fix-834'
alexigna@mbp-alexigna 01.init % touch f2.txt
alexigna@mbp-alexigna 01.init % git add .
alexigna@mbp-alexigna 01.init % git commit -m "add new f2 file"
[feature/fix-159 4e6b59e] add new f2 file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f2.txt
alexigna@mbp-alexigna 01.init % echo 1 >> f2.txt
alexigna@mbp-alexigna 01.init % git commit -a -m "1 to f2 file"
[feature/fix-159 38322ab] 1 to f2 file
1 file changed, 1 insertion(+)
alexigna@mbp-alexigna 01.init % echo 2 >> f2.txt
alexigna@mbp-alexigna 01.init % git commit -a -m "2 to f2 file"
[feature/fix-159 e13613c] 2 to f2 file
1 file changed, 1 insertion(+)
alexigna@mbp-alexigna 01.init % echo 3 >> f2.txt
alexigna@mbp-alexigna 01.init % git commit -a -m "3 to f2 file"
[feature/fix-159 d36584d] 3 to f2 file
1 file changed, 1 insertion(+)
alexigna@mbp-alexigna 01.init % git log --oneline master..HEAD
d36584d (HEAD -> feature/fix-159) 3 to f2 file
e13613c 2 to f2 file
38322ab 1 to f2 file
4e6b59e add new f2 file
```

# rebase

```
alexigna@mbp-alexigna 01.init % git rebase -i master
```

```
--  
pick 4e6b59e add new f2 file  
squash 38322ab 1 to f2 file  
squash e13613c 2 to f2 file  
squash d36584d 3 to f2 file  
--
```

```
# This is a combination of 4 commits.  
# This is the 1st commit message:
```

```
add new f2 file with values  
--
```

```
alexigna@mbp-alexigna 01.init % git rebase -i master  
[detached HEAD ae6b8e6] add new f2 file with values  
Date: Fri Jul 21 20:31:05 2023 +0500  
1 file changed, 3 insertions(+)  
create mode 100644 f2.txt  
Successfully rebased and updated refs/heads/feature/fix-159.
```

```
alexigna@mbp-alexigna 01.init % git log master..HEAD --oneline  
ae6b8e6 (HEAD -> feature/fix-159) add new f2 file with values
```

`rebase -i master` - изменения в интерактивном режиме всех commits до commit, на который ссылается master

`rebase -i HEAD~2` - последние два commits

`rebase -i 6c4d396` - все commits до указанного



# tag

```
alexigna@mbp-alexigna 01.init % git log --oneline
6c4d396 (HEAD -> master) merge from fix-312
373eed5 custom merge
c82d0b6 new changes from master
afdb1e7 (tag: v1.2) new changes
294d7ca merged
d08d6cd change from master
576f1fb fix some issue
alexigna@mbp-alexigna 01.init % git tag -a v0.8 -m "add new tag" d08d6cd
alexigna@mbp-alexigna 01.init % git tag v0.9 294d7ca

alexigna@mbp-alexigna 01.init % git log --oneline

6c4d396 (HEAD -> master) merge from fix-312
373eed5 custom merge
c82d0b6 new changes from master
afdb1e7 (tag: v1.2) new changes
294d7ca (tag: v0.9) merged
d08d6cd (tag: v0.8) change from master
576f1fb fix some issue
a264075 init

alexigna@mbp-alexigna 01.init % git tag
v0.8
v0.9
v1.2
alexigna@mbp-alexigna 01.init % git tag -l "v1.*"
v1.2
```

Tag помечает определенные commits в истории как важные (например релизы)

Могут быть:

- Легковесные - просто указатель на commit в папке .git/refs/tags/ (похожи на ветку)
- Аннотированные - хранятся как объекты git типа tag, содержат метадату с временем создания, сообщением, автором и пр.

# tag

При переключение на tag репозиторий переходит в состояние «detached HEAD» (в рабочий каталог извлечен не тот commit, на который указывает HEAD)

```
alexigna@mbp-alexigna 01.init % git checkout v1.2
Note: switching to 'v1.2'.
```

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 switching back to a branch.

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

```
git switch -c <new-branch-name>
```

Or undo this operation with:

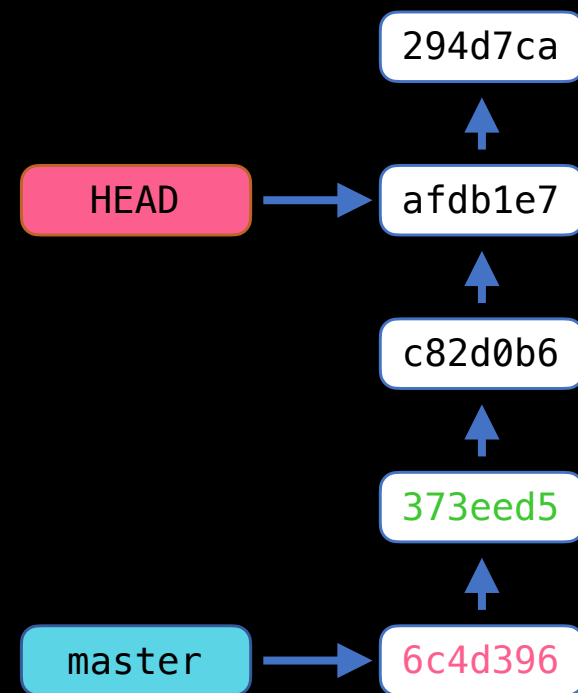
```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to false

HEAD is now at afdb1e7 new changes

```
alexigna@mbp-alexigna 01.init % git switch -
Previous HEAD position was afdb1e7 new changes
Switched to branch 'master'
```

```
alexigna@mbp-alexigna 01.init % git checkout -b fix-for-v1.2 v1.2
Switched to a new branch 'fix-for-v1.2'
```



# Удаленный репозиторий

Удалённый репозиторий это версии проекта, сохранённая в локальной сети или Интернет.

```
alexigna@mbp-alexigna 02.scrapli-community % git clone https://github.com/scrapli/scrapli_community.git .
Cloning into '.'...
remote: Enumerating objects: 3359, done.
remote: Counting objects: 100% (1931/1931), done.
remote: Compressing objects: 100% (605/605), done.
remote: Total 3359 (delta 976), reused 1728 (delta 888), pack-reused 1428
Receiving objects: 100% (3359/3359), 2.80 MiB | 2.86 MiB/s, done.
Resolving deltas: 100% (1678/1678), done.

alexigna@mbp-alexigna 02.scrapli-community % git remote
origin

alexigna@mbp-alexigna 02.scrapli-community % git remote -v
origin https://github.com/scrapli/scrapli_community.git (fetch)
origin https://github.com/scrapli/scrapli_community.git (push)
```

`remote` - имя удаленного репозитория, `origin` имя по-умолчанию

`remote -v` - адреса чтения/записи

# Удаленный репозиторий

Удаленных репозиториум у проекта может быть несколько, все они равнозначны

```
alexigna@mbp-alexigna 02.scrapli-community % git remote add my-scrapli https://github.com/alsigna/my-scrapli-community.git

alexigna@mbp-alexigna 02.scrapli-community % git remote -v
my-scrapli https://github.com/alsigna/my-scrapli-community.git (fetch)
my-scrapli https://github.com/alsigna/my-scrapli-community.git (push)
origin https://github.com/scrapli/scrapli_community.git (fetch)
origin https://github.com/scrapli/scrapli_community.git (push)

alexigna@mbp-alexigna 02.scrapli-community % git push my-scrapli
Enumerating objects: 2142, done.
Counting objects: 100% (2142/2142), done.
Delta compression using up to 8 threads
Compressing objects: 100% (785/785), done.
Writing objects: 100% (2142/2142), 325.35 KiB | 162.68 MiB/s, done.
Total 2142 (delta 1116), reused 2134 (delta 1113), pack-reused 0
remote: Resolving deltas: 100% (1116/1116), done.
To https://github.com/alsigna/my-scrapli-community.git
 * [new branch]      main -> main
```

# Удаленный репозиторий

```
alexigna@mbp-alexigna 02.scrapli-community % git fetch my-scrapli
From https://github.com/alsigna/my-scrapli-community
* [new branch]      some-test-feature -> my-scrapli/some-test-feature

alexigna@mbp-alexigna 02.scrapli-community % git branch -a
* main
remotes/my-scrapli/main
remotes/my-scrapli/some-test-feature
remotes/origin/2021.07.30
remotes/origin/2022.01.30
remotes/origin/2022.07.30
remotes/origin/2023.01.30
remotes/origin/HEAD -> origin/main
remotes/origin/gh-pages
remotes/origin/main
```

`fetch` - получает коммиты с удаленного сервера которых нет в текущей локальной ветке и сохраняет их в репозитории, но не сливает с локальной веткой

`pull` - получает коммиты с удаленного сервера которых нет в текущей локальной ветке, сохраняет их в репозитории, и делает merge в локальную ветку. `pull` = `fetch` + `merge`

# Удаленный репозиторий

```
alexigna@mbp-alexigna 02.scrapli-community % git checkout -b some-test-feature
Switched to a new branch 'some-test-feature'

alexigna@mbp-alexigna 02.scrapli-community % git pull my-scrapli some-test-feature
From https://github.com/alsigna/my-scrapli-community
 * branch      some-test-feature -> FETCH_HEAD
Updating 4b2be4e..9536084
Fast-forward
 scrapli_community/cisco/asa/cisco_asa.py | 1 +
 1 file changed, 1 insertion(+)

alexigna@mbp-alexigna 02.scrapli-community % git diff remotes/origin/main..some-test-feature
diff --git a/scrapli_community/cisco/asa/cisco_asa.py b/scrapli_community/cisco/asa/cisco_asa.py
index d775bce..180d766 100644
--- a/scrapli_community/cisco/asa/cisco_asa.py
+++ b/scrapli_community/cisco/asa/cisco_asa.py
@@ -1,4 +1,5 @@
 """scrapli_community.cisco.asa.cisco_asa"""
+""" add some text here """

from scrapli.driver.network.base_driver import PrivilegeLevel
from scrapli_community.cisco.asa.async_driver import (
```

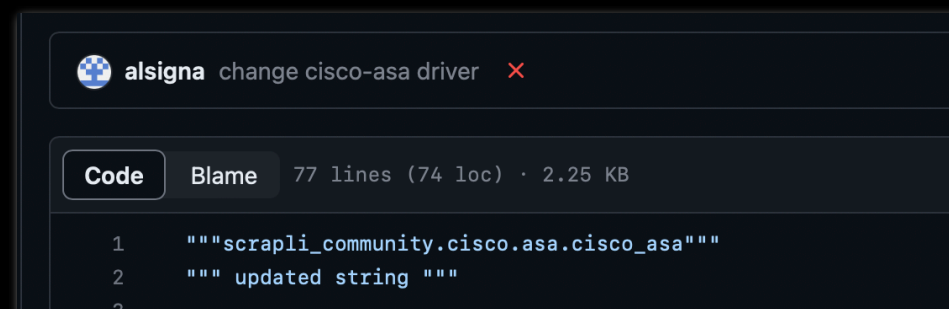
# Удаленный репозиторий

```
alexigna@mbp-alexigna 02.scrapli-community % vi scrapli_community/cisco/asa/cisco_asa.py

alexigna@mbp-alexigna 02.scrapli-community % git diff
diff --git a/scrapli_community/cisco/asa/cisco_asa.py b/scrapli_community/cisco/asa/cisco_asa.py
index 180d766..b930156 100644
--- a/scrapli_community/cisco/asa/cisco_asa.py
+++ b/scrapli_community/cisco/asa/cisco_asa.py
@@ -1,5 +1,5 @@
-"""scrapli_community.cisco.asa.cisco_asa"""
-""" add some text here """
+""" updated string """

from scrapli.driver.network.base_driver import PrivilegeLevel
from scrapli_community.cisco.asa.async_driver import (
alexigna@mbp-alexigna 02.scrapli-community % git commit -a -m "change cisco-asa driver"
[some-test-feature a86d990] change cisco-asa driver
1 file changed, 1 insertion(+), 1 deletion(-)

alexigna@mbp-alexigna 02.scrapli-community % git push my-scrapli
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 569 bytes | 569.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/alsigna/my-scrapli-community.git
9536084..a86d990 some-test-feature -> some-test-feature
```



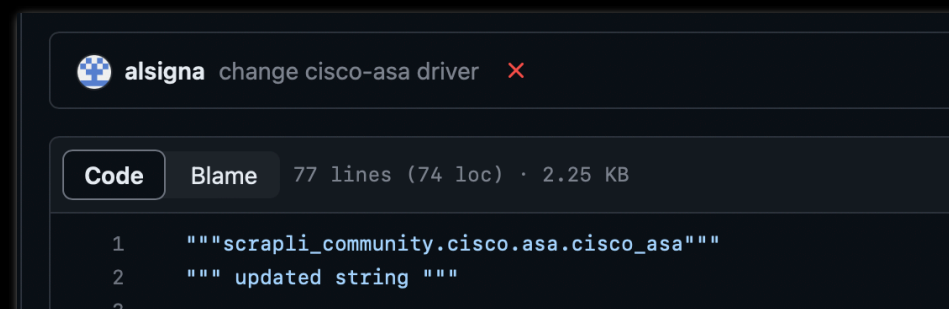
# Удаленный репозиторий

```
alexigna@mbp-alexigna 02.scrapli-community % vi scrapli_community/cisco/asa/cisco_asa.py

alexigna@mbp-alexigna 02.scrapli-community % git diff
diff --git a/scrapli_community/cisco/asa/cisco_asa.py b/scrapli_community/cisco/asa/cisco_asa.py
index 180d766..b930156 100644
--- a/scrapli_community/cisco/asa/cisco_asa.py
+++ b/scrapli_community/cisco/asa/cisco_asa.py
@@ -1,5 +1,5 @@
-"""scrapli_community.cisco.asa.cisco_asa"""
-""" add some text here """
+""" updated string """

from scrapli.driver.network.base_driver import PrivilegeLevel
from scrapli_community.cisco.asa.async_driver import (
alexigna@mbp-alexigna 02.scrapli-community % git commit -a -m "change cisco-asa driver"
[some-test-feature a86d990] change cisco-asa driver
1 file changed, 1 insertion(+), 1 deletion(-)

alexigna@mbp-alexigna 02.scrapli-community % git push my-scrapli
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 569 bytes | 569.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/alsigna/my-scrapli-community.git
9536084..a86d990 some-test-feature -> some-test-feature
```





# .gitignore

- Файлы, которые не нужно добавлять в репозиторий (logs/venv/tmp/etc)
- Список файлов для игнорирования задаётся в файле .gitignore
- Возможно использование regex
- Уже есть готовые шаблоны <https://github.com/github/gitignore>

```
gitignore / Python.gitignore
Code Blame 160 lines (131 loc) · 3.01 KB
119 # SageMath parsed files
120 *.sage.py
121
122 # Environments
123 .env
124 .venv
125 env/
126 venv/
127 ENV/
128 env.bak/
129 venv.bak/
```

# .gitignore

```
# Исключить все файлы с расширением .a
*.a
# Но отслеживать файл lib.a даже если он подпадает под исключение выше
!lib.a
# Игнорировать все файлы в каталоге build/
build/
# Игнорировать файл doc/notes.txt, но не файл doc/server/arch.txt
doc/*.txt
# Игнорировать все .txt файлы в каталоге doc/
doc/**/*.txt
```

# hooks

Git позволяет запускать пользовательские скрипты как реакции на какие-либо события

Хранятся скрипты в каталоге `.git/hooks`

Два типа:

- Клиентские (локальные) - работают на локальной машине, не загружаются в удаленный репозиторий, можно настраивать для себя как угодно. Можно игнорировать через ключ `--no-verify`, например `git commit --no-verify`.
- Серверные (удаленные) - работают на стороне удаленного репозитория и позволяют применять требуемые политики централизованно. Невозможно игнорировать и невозможно скачать из удаленного репозитория.

Скрипты пишутся на любом скриптовом языке: `bash`, `python`, `php`, `ruby`, ...

Результат работы скрипта - код. Любое значение, отличное от 0 прервет выполнение операции

# hooks

Серверные:

- pre-receive

Выполняется каждый раз при push в репозиторий. Скрипт параметры не принимает, информация передается через stdin в виде `<sha_old_value> <sha_old_value> <ref_name>`

- update

Вызывается после pre-receive, но выполняется для каждой ветки отдельно (pre-receive выполняется единожды для всех). Данные передаются не через stdin, а как аргументы.

- post-receive

Вызывается после успешного push. Обычно используется для отправки сообщений, запуска CI/CD, взаимодействие с другими системами. Принимает ту же информацию через stdin, что и pre-receive.

# hooks

Локальные:

- `pre-commit`

Выполняется при каждой команде `commit` перед тем как будет запрошен комментарий или создастся объект `commit`. Нет входных параметров. Можно использовать, например для запуска линтеров.

- `prepare-commit-msg`

Вызывается после `pre-commit`, и нужен для правки `commit` сообщения по-умолчанию. Позволяет исправить его до того, как сообщение будет видно автору. Актуально для `commit` через текстовый редактор.

- `commit-msg`

Вызывается после `prepare-commit-msg` и на этом этапе уже доступно сообщение, которое ввел автор. На этом этапе можно отклонить `commit` если он не соответствует шаблону.

# hooks

Локальные:

- post-commit

Выполняется после commit-msg, на этом этапе уже нельзя поменять commit сообщение. Обычно используется для отсылке уведомлений.

- post-checkout

Вызывается после переключения между ветками (git checkout). Можно использовать, например, для очистки рабочего каталога от временных файлов.

- pre-rebase

Вызывается перед операций rebase. Через этот hook можно, например, полностью запретить rebase (всегда exit code 1).

**NFE** Networking  
For  
Everyone

