Никита Соболев

github.com/sobolevn

Система контроля версий

Все начинается, когда вы просто хотите хранить историю изменений вашей работы

mymodule.py

```
1 # Previously I had some complex logic in this
2 # module, but I had to delete it due to reasons.
3 # Very sad!
4
5 def my_function(first, second):
6     return first + second
7
8 def another_function(first):
9     return first + 1
```

Вопросы

- Какие изменения были в данном файле?
- Когда они были?
- Кто их вносил?
- Почему их вносили? Какая была цель?

Без истории изменений – мы не знаем!

git = распределенная система для хранения коллективной истории изменений всех файлов в проекте

Но как мы будем хранить изменения?

Варианты

- Хранить все файлы всегда, целиком и полностью
- Хранить только первую версию и наборы изменений, чтобы можно было воссоздать по ним содержимое файла на любой момент времени

И что такое "момент времени"?

Варианты

- Мы будем сохранять вообще все: каждую секунду, любое изменение

- Мы будем сами создавать законченные "версии" наших файлов, когда посчитаем нужным

А что значит "распределенная"?

"Распределенная" = можно хранить историю сразу в нескольких местах

Наборы изменений

Было

```
1 def my_function(first, second):
2     return first * second + 10
```

Внесем первое изменение

```
1 def my_function(first, second):
2     return first * second + 10
3
4 +++ def another_function(first):
5 +++     return first + 1
```

Получим

```
1 def my_function(first, second):
2     return first * second + 10
3
4 def another_function(first):
5     return first + 1
```

Внесем второе изменение

```
+++# Previously I had some complex logic in this
+++# module, but I had to delete it due to reasons.
+++# Very sad!

def my_function(first, second):
---    return first * second + 10
+++    return first + second
```

Получим

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return first + second

def another_function(first):
    return first + 1
```

Итого

```
def my_function(first, second):
    return first * second + 10
```

+

```
+++ def another_function(first):
+++     return first + 1
```

+

```
+++# Previously I had some complex logic in this
+++# module, but I had to delete it due to reasons.
+++# Very sad!

def my_function(first, second):
---    return first * second + 10
+++    return first + second
```

=

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return first + second

def another_function(first):
    return first + 1
```

Хранение наборов изменений – позволит нам существенно сократить объем информации. И тонко контролировать их применение.

Законченные версии файлов

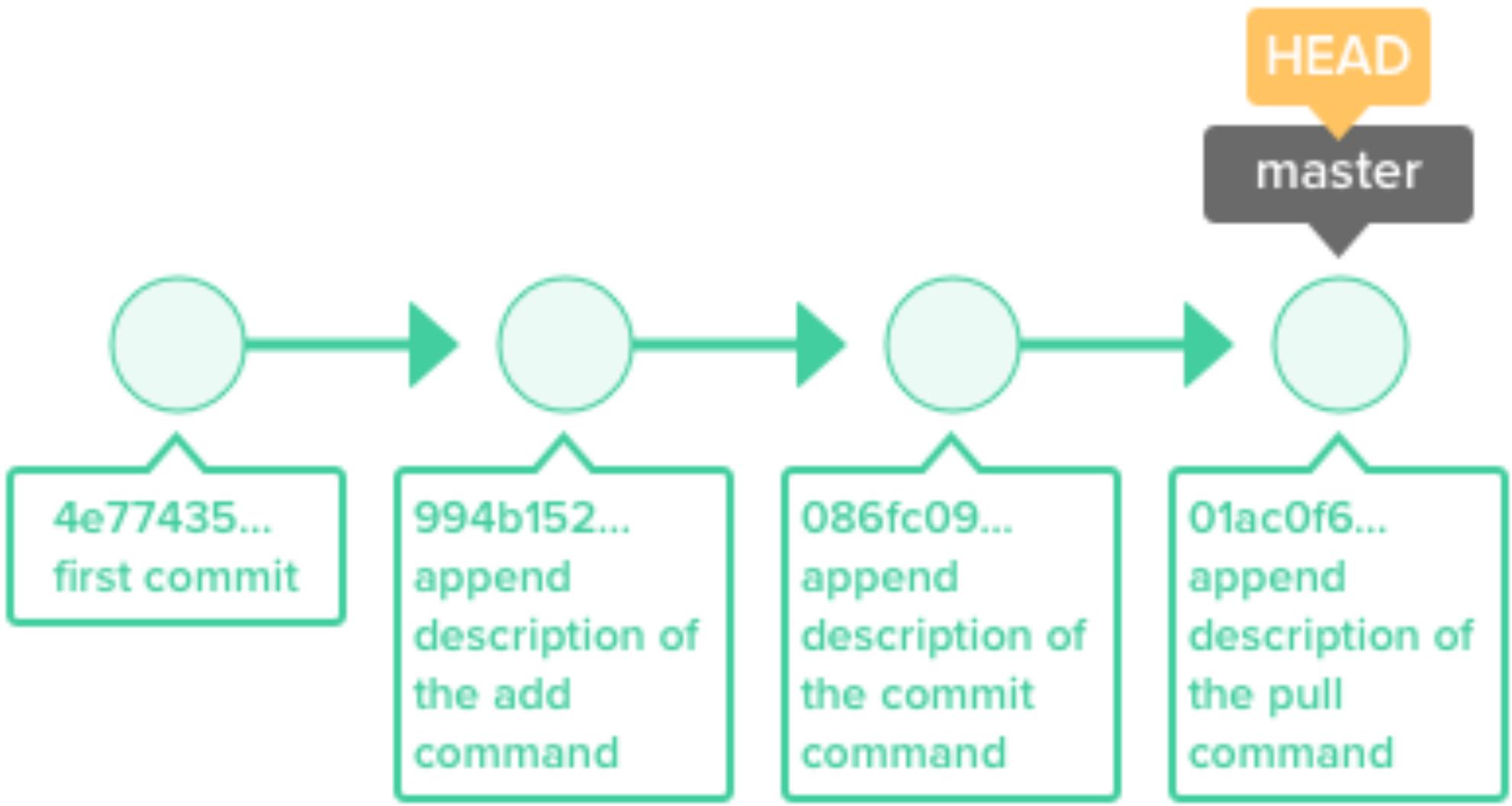
Иногда во время работы мы будем "сохранять" наши изменения как законченные варианты

Такие "законченные версии" мы будем называть "коммиты"

Каждый коммит имеет:

- Уникальное имя
- Дату и время создания
- Автора
- Пояснительное сообщение: зачем его создавали
- Набор изменений с момента прошлого коммита

Коммиты объединяются в ветки



Попробуем сделать наш пример с изменениями в git

git init

```
~/Desktop/example
» la
total 4.0K
-rw-r--r-- 1 sobolev staff 217 Sep 22 23:09 mymodule.py

~/Desktop/example
» git init
Initialized empty Git repository in /Users/sobolev/Desktop/example/.git/

~/Desktop/example  master ✗
» la
total 4.0K
drwxr-xr-x 9 sobolev staff 288 Sep 22 23:12 .git/
-rw-r--r-- 1 sobolev staff 217 Sep 22 23:09 mymodule.py

~/Desktop/example  master ✗
»
```

.git/ = папка, где будет храниться история изменений и мета-информация

У нас есть возможность смотреть информацию о состоянии проекта

git log

```
~/Desktop/example  master ✗
» git log
fatal: your current branch 'master' does not have any commits yet

~/Desktop/example  master ✗                                    128 ⚠
»
```

git status

```
~/Desktop/example  master ✗
» git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        mymodule.py

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

~/Desktop/example  master ✗
»
```

Первая версия нашего файла mymodule.py

```
1 def my_function(first, second):
2     return first * second + 10
```

Создадим первый КОММИТ

Каждый коммит имеет:

- Уникальное имя: его git присвоит автоматически
- Дату и время создания: она известна
- Автора: я автор! Но кто я?
- Пояснительное сообщение: мы его напишем
- Набор изменений с прошлого коммита: мы их добавим

Итого, потребуется:

- Единожды указать свои email и имя пользователя для git
- Пояснительное сообщение: мы его напишем
- Набор изменений с момента прошлого коммита: мы их добавим

Единожды представимся

```
git config --global user.name "Your Name"
git config --global user.email "your@email"
```

Набор изменений, который мы еще не зафиксировали коммитом, называется staging

git add

```
~/Desktop/example  master ✗
» git add mymodule.py

~/Desktop/example  master ✗                                                +
» git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   mymodule.py


~/Desktop/example  master ✗                                                +
»
```

git commit

```
~/Desktop/example  master ✗                                              +
» git commit -m 'Initial commit'
[master (root-commit) 6387609] Initial commit
 1 file changed, 2 insertions(+)
 create mode 100644 mymodule.py

~/Desktop/example  master ✔
»
```

git status && git log

```
~/Desktop/example  master ✔
» git status
On branch master
nothing to commit, working tree clean

~/Desktop/example  master ✔
» git log
commit 6387609c4ead14890a3eb09e719317687ded0e5c (HEAD -> master)
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:04:02 2020 +0300

    Initial commit

~/Desktop/example  master ✔
»
```

git show

```
~/Desktop/example  master ✔
» git show 6387609c4ead14890a3eb09e719317687ded0e5c
commit 6387609c4ead14890a3eb09e719317687ded0e5c (HEAD -> master)
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:04:02 2020 +0300

    Initial commit

diff --git a/mymodule.py b/mymodule.py
new file mode 100644
index 0000000..d952470
--- /dev/null
+++ b/mymodule.py
@@ -0,0 +1,2 @@
+def my_function(first, second):
+    return first * second + 10

~/Desktop/example  master ✔
»
```

Внесем первое изменение

```
4 +++ def another_function(first):
5 +++     return first + 1
```

git diff

```
~/Desktop/example  master ✗
» git diff
diff --git a/mymodule.py b/mymodule.py
index d952470..1170ac5 100644
--- a/mymodule.py
+++ b/mymodule.py
@@ -1,2 +1,5 @@
 def my_function(first, second):
     return first * second + 10
+
+def another_function(first):
+    return first + 1

~/Desktop/example  master ✗
»
```

git status && git log

```
~/Desktop/example  master ✗
» git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mymodule.py

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

~/Desktop/example  master ✗
» git log
commit 6387609c4ead14890a3eb09e719317687ded0e5c (HEAD -> master)
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:04:02 2020 +0300

    Initial commit

~/Desktop/example  master ✗
»
```

git add

```
~/Desktop/example  master ✗
» git add mymodule.py

~/Desktop/example  master ✗                                                +
» git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   mymodule.py


~/Desktop/example  master ✗                                                +
» 
```

git commit && git log

```
~/Desktop/example  master ✗                                                +
» git commit -m 'Adds another_function to mymodule.py'
[master b2af568] Adds another_function to mymodule.py
 1 file changed, 3 insertions(+)

~/Desktop/example  master ✔
» git log
commit b2af568974eaeea096e387020c4d974c3c6b529a (HEAD -> master)
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:13:00 2020 +0300

    Adds another_function to mymodule.py

commit 6387609c4ead14890a3eb09e719317687ded0e5c
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:04:02 2020 +0300

    Initial commit

~/Desktop/example  master ✔
»
```

Внесем второе изменение

```
+++# Previously I had some complex logic in this
+++# module, but I had to delete it due to reasons.
+++# Very sad!

def my_function(first, second):
---    return first * second + 10
+++    return first + second
```

git diff

```
~/Desktop/example  master ✗
» git diff
diff --git a/mymodule.py b/mymodule.py
index 1170ac5..cefc5aa 100644
--- a/mymodule.py
+++ b/mymodule.py
@@ -1,5 +1,9 @@
+# Previously I had some complex logic in this
+# module, but I had to delete it due to reasons.
+# Very sad!
+
 def my_function(first, second):
-    return first * second + 10
+    return first + second

 def another_function(first):
     return first + 1

~/Desktop/example  master ✗
»
```

git add

```
~/Desktop/example  master ✗
» git add mymodule.py

~/Desktop/example  master ✗                                                +
» git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   mymodule.py


~/Desktop/example  master ✗                                                +
» 
```

git commit && git log

```
~/Desktop/example  master ✗                                                +
» git commit -m 'Adds comments'
[master 74c351a] Adds comments
 1 file changed, 5 insertions(+), 1 deletion(-)

~/Desktop/example  master ✔
» git log
commit 74c351a615f9ee9ee0813d29d8e35582e0ca4cf2 (HEAD -> master)
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:16:31 2020 +0300

    Adds comments

commit b2af568974eaeea096e387020c4d974c3c6b529a
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:13:00 2020 +0300

    Adds another_function to mymodule.py

commit 6387609c4ead14890a3eb09e719317687ded0e5c
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:04:02 2020 +0300

    Initial commit
```

Готово!

Практика

Теперь поговорим про "распределенную" систему

Нам нужно сохранять историю проекта где-то в интернете

Например, на GitHub!

Create a New Repository

https://github.com/new

sobolevn / our_first_repo

Great repository names are short and memorable. Need inspiration? How about **redesigned-octo-parakeet**?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

Choose a license
A license tells others what they can and can't do with your code. Learn more.

Grant your Marketplace apps access to this repository

You are subscribed to 2 Marketplace apps

Dependabot Preview
Automated dependency updates for Ruby, JavaScript, Python, Go, PHP, Elixir, Rust, Java and .NET

Task list completed
PR check to make sure all tasklists / tick boxes are checked in the PR body

Create repository

sobolevn/our_first_repo

https://github.com/sobolevn/our_first_repo

Search or jump to... / Pull requests Issues Marketplace Explore

Dependabot Preview was installed on this repository

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/sobolevn/our_first_repo.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# our_first_repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/sobolevn/our_first_repo.git
git push -u origin master
```

...or push an existing repository from the command line

sobolevn/our_first_repo

https://github.com/sobolevn/our_first_repo

Search or jump to... / Pull requests Issues Marketplace Explore

Dependabot Preview was installed on this repository

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/sobolevn/our_first_repo.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# our_first_repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/sobolevn/our_first_repo.git
git push -u origin master
```

...or push an existing repository from the command line

git remote

```
~/Desktop/example  master ✔
» git remote

~/Desktop/example  master ✔
»
```

git remote

```
~/Desktop/example  master ✔
» git remote add origin https://github.com/sobolevn/our_first_repo.git

~/Desktop/example  master ✔
» git remote
origin

~/Desktop/example  master ✔
»
```

Мы связали локальный проект с GitHub

Теперь, нужно синхронизировать изменения в них!

git push

```
~/Desktop/example  master ✔
» git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 1.98 KiB | 1.98 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/sobolevn/our_first_repo.git
 * [new branch]      master -> master

~/Desktop/example  master ✔
»
```

sobolevn/our_first_repo

https://github.com/sobolevn/our_first_repo

Search or jump to... / Pull requests Issues Marketplace Explore

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

sobolevn Adds comments 74c351a 30 minutes ago 3 commits

mymodule.py Adds comments 30 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

https://github.com/sobolevn/our_first_repo/commits/master

Search or jump to... / Pull requests Issues Marketplace Explore

sobolevn / our_first_repo

Unwatch

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master

Commits on Sep 23, 2020

Adds comments
sobolevn committed 30 minutes ago

Adds another_function to mymodule.py
sobolevn committed 34 minutes ago

Initial commit
sobolevn committed 43 minutes ago

Newer Older

our_first_repo/mymodule.py at r

https://github.com/sobolevn/our_first_repo/blame/master/mymodule.py

Search or jump to... / Pull requests Issues Marketplace Explore

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

our_first_repo / mymodule.py

Newer Older

100644 | 9 lines (7 sloc) | 217 Bytes

Raw Normal view History

Adds comments	31 minutes ago	
Initial commit	44 minutes ago	
Adds comments	31 minutes ago	
Adds another_function to mymodule.py	35 minutes ago	

```
1  # Previously I had some complex logic in this
2  # module, but I had to delete it due to reasons.
3  # Very sad!
4
5  def my_function(first, second):
6      return first + second
7
8  def another_function(first):
9      return first + 1
```

На гитхабе мы можем добавить или поменять что-то

sobolevn/our_first_repo

https://github.com/sobolevn/our_first_repo

Search or jump to... / Pull requests Issues Marketplace Explore

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

sobolevn Adds comments 74c351a 30 minutes ago 3 commits

mymodule.py Adds comments 30 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

New File

https://github.com/sobolevn/our_first_repo/new/master

Search or jump to... / Pull requests Issues Marketplace Explore

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

our_first_repo / new_module.py Cancel

Edit new file Preview Spaces 2 No wrap

```
1  def new_function():
2      print('I am from github!')
```

New File

https://github.com/sobolevn/our_first_repo/new/master

Commit new file

Adds new_module.py from github

Add an optional extended description...

mail@sobolevn.me

Choose which email address to associate with this commit

Commit directly to the `master` branch.

Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Commit new file Cancel

sobolevn/our_first_repo

https://github.com/sobolevn/our_first_repo/tree/master

Search or jump to... / Pull requests Issues Marketplace Explore

sobolevn / our_first_repo

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

sobolevn Adds new_module.py from github ed08b4b now 4 commits

mymodule.py	Adds comments	37 minutes ago
new_module.py	Adds new_module.py from github	now

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Python** 100.0%

Before git pull

```
~/Desktop/example  master ✔
» ls
mymodule.py

~/Desktop/example  master ✔
» git log
commit 74c351a615f9ee9ee0813d29d8e35582e0ca4cf2 (HEAD -> master, origin/master)
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:16:31 2020 +0300

    Adds comments

commit b2af568974eaeea096e387020c4d974c3c6b529a
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:13:00 2020 +0300

    Adds another_function to mymodule.py

commit 6387609c4ead14890a3eb09e719317687ded0e5c
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:04:02 2020 +0300

    Initial commit

~/Desktop/example  master ✔
»
```

git pull

```
~/Desktop/example  master ✔
» git pull origin master
From https://github.com/sobolevn/our_first_repo
 * branch            master     -> FETCH_HEAD
Updating 74c351a..ed08b4b
Fast-forward
 new_module.py | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 new_module.py

~/Desktop/example  master ✔
»
```

git log

```
~/Desktop/example  master ✔
» ls
mymodule.py  new_module.py

~/Desktop/example  master ✔
» git log
commit ed08b4b6f778ce148c86b1d3e1c32db5070792ad (HEAD -> master, origin/master)
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 00:53:21 2020 +0300

    Adds new_module.py from github

commit 74c351a615f9ee9ee0813d29d8e35582e0ca4cf2
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:16:31 2020 +0300

    Adds comments

commit b2af568974eaeea096e387020c4d974c3c6b529a
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:13:00 2020 +0300

    Adds another_function to mymodule.py

commit 6387609c4ead14890a3eb09e719317687ded0e5c
Author: sobolevn <mail@sobolevn.me>
```

Готово!

https://github.com/sobolevn/our_first_repo

Полезности

Можно скопировать репозиторий из сети

git clone

```
~/Desktop
» git clone https://github.com/sobolevn/our_first_repo.git
Cloning into 'our_first_repo'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 1), reused 9 (delta 1), pack-reused 0
Unpacking objects: 100% (15/15), done.

~/Desktop
» cd our_first_repo

~/Desktop/our_first_repo  master ✔
» ls
mymodule.py  new_module.py

~/Desktop/our_first_repo  master ✔
»
```

Файл .gitignore

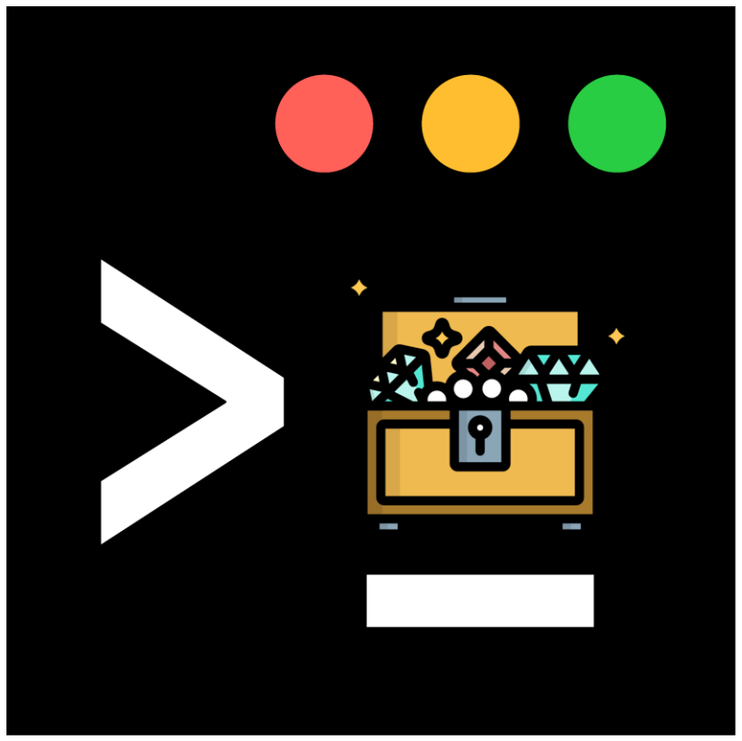
https://github.com/sobolevn/our_first_repo/blob/master/.gitignore

Пример

```
# Byte-compiled / optimized / DLL files
.pytest_cache
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg
pip-wheel-metadata/
```

Онлайн курсы

- https://git-school.github.io/visualizing-git/
- https://learngitbranching.js.org/?locale=ru_RU
- https://git-scm.com/doc
- https://www.jetbrains.com/help/pycharm/using-git-integration.html



github.com/
sobolevn/dotfiles

Выводы

- Без истории изменений в нашем проекте – будет очень сложно работать
- Хранить историю лучше всего как набор изменений в фиксированные точки времени жизни проекта
- git позволяет удобно хранить историю нашего проекта
- Его распределенная натура позволяет нам хранить историю сразу в нескольких местах и синхронизировать ее

Никита Соболев

github.com/sobolevn


git

Работа с git в команде

mymodule.py

```
1 # Previously I had some complex logic in this
2 # module, but I had to delete it due to reasons.
3 # Very sad!
4
5 def my_function(first, second):
6     return first + second
7
8 def another_function(first):
9     return first + 1
```

Начальное состояния

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

Первое изменение

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

+

Коммит "B"

```
4 +++ def another_function(first):
5 +++     return first + 1
```

Второе изменение

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

+

Коммит "B"

```
4 +++ def another_function(first):
5 +++     return first + 1
```

+

Коммит "C"

```
1 +++# Previously I had some complex logic in this
2 +++# module, but I had to delete it due to reasons.
3 +++# Very sad!
4
5 def my_function(first, second):
6 ---    return first * second + 10
6 +++    return first + second
```

Полная история

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

+

Коммит "B"

```
4 +++ def another_function(first):
5 +++     return first + 1
```

+

Коммит "C"

```
1 +++# Previously I had some complex logic in this
2 +++# module, but I had to delete it due to reasons.
3 +++# Very sad!
4
5 def my_function(first, second):
6 ---   return first * second + 10
6 +++   return first + second
```

=

==>

```
1 # Previously I had some complex logic in this
2 # module, but I had to delete it due to reasons.
3 # Very sad!
4
5 def my_function(first, second):
6     return first + second
7
8 def another_function(first):
9     return first + 1
```

Переместимся в коммит B

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

+

Коммит "B"

```
4 +++ def another_function(first):
5 +++     return first + 1
```

=

==>

```
1 def my_function(first, second):
2     return first * second + 10
3
4 def another_function(first):
5     return first + 1
```

Переместимся в коммит A

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

=

==>

```
1 def my_function(first, second):
2     return first * second + 10
```

Вернемся в коммит C

Коммит "A"

```
1 def my_function(first, second):
2     return first * second + 10
```

+

Коммит "B"

```
4 +++ def another_function(first):
5 +++     return first + 1
```

+

Коммит "C"

```
1 +++# Previously I had some complex logic in this
2 +++# module, but I had to delete it due to reasons.
3 +++# Very sad!
4
5 def my_function(first, second):
6 ---   return first * second + 10
6 +++   return first + second
```

=

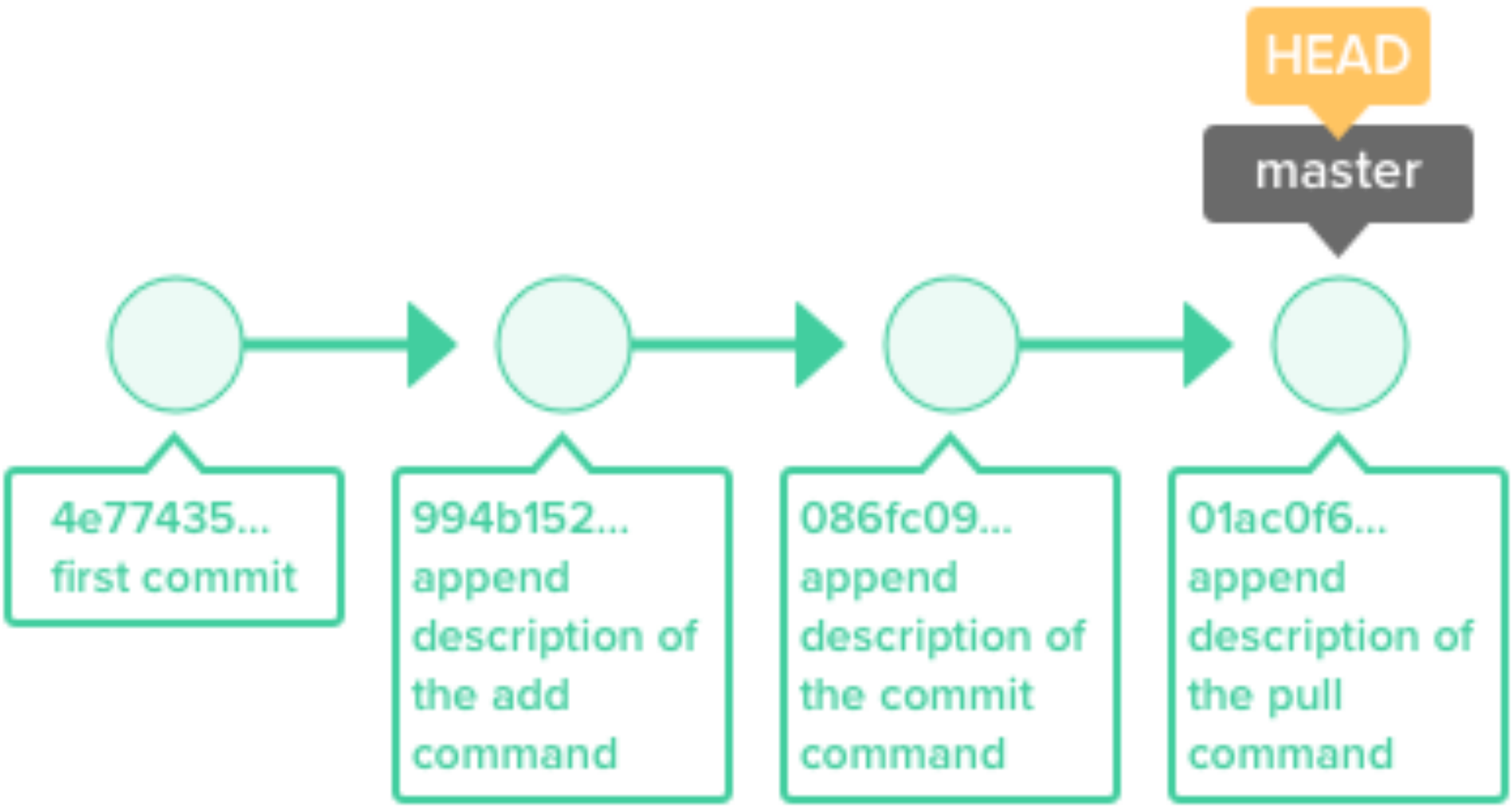
==>

```
1 # Previously I had some complex logic in this
2 # module, but I had to delete it due to reasons.
3 # Very sad!
4
5 def my_function(first, second):
6     return first + second
7
8 def another_function(first):
9     return first + 1
```

git checkout

```
~/Desktop/our_first_repo  master ✔
» git checkout 6387609c4ead14890a3eb09e719317687ded0e5c
Note: switching to '6387609c4ead14890a3eb09e719317687ded0e5c'.

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 6387609 Initial commit

~/Desktop/our_first_repo  (6387609...) ✔
»
```

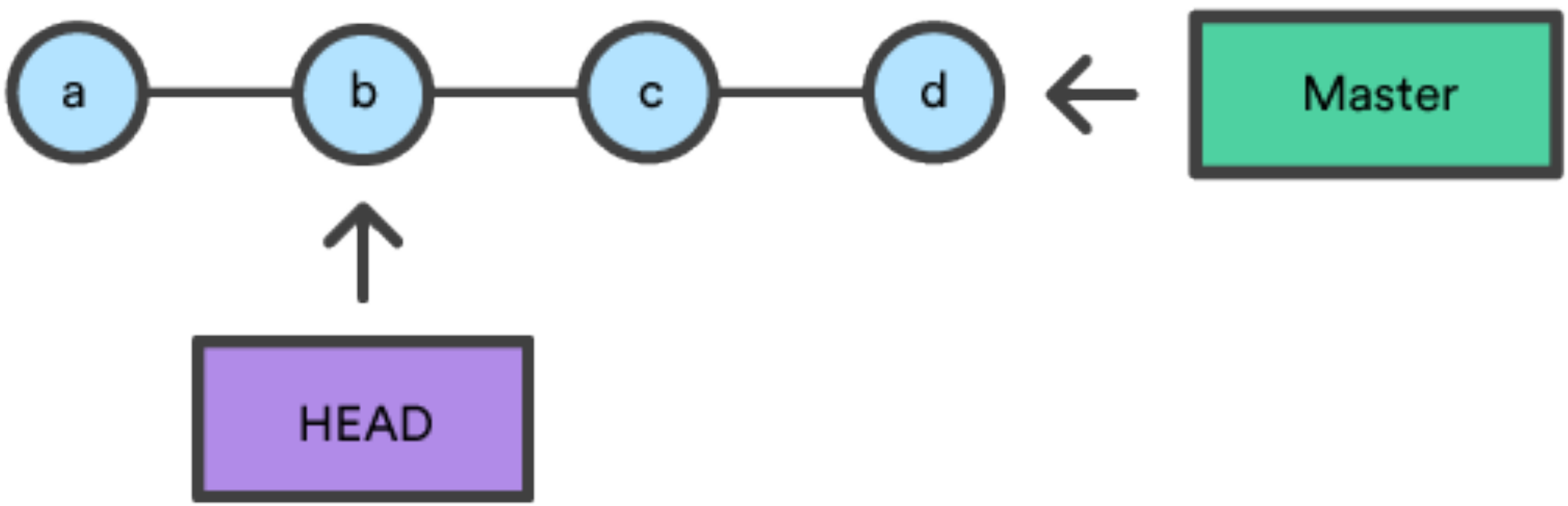
Коммиты объединяются в ветки



"git checkout" работает

- С номерами коммитов
- С названиями веток

Состояние неопределенности или "detached HEAD"



git branch

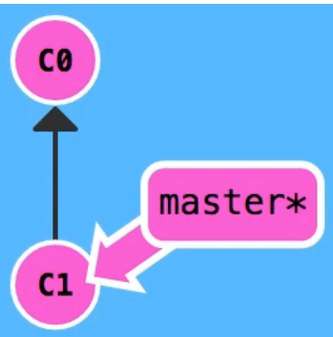
```
~/Desktop/our_first_repo  master ✔
» git branch
* master

~/Desktop/our_first_repo  master ✔
»
```

git branch second

```
~/Desktop/our_first_repo  master ✔
» git branch
* master

~/Desktop/our_first_repo  master ✔
» git branch second

~/Desktop/our_first_repo  master ✔
» git branch -a
* master
  second
  remotes/origin/HEAD -> origin/master
  remotes/origin/master

~/Desktop/our_first_repo  master ✔
»
```

Изучаем ветвление в git

```
$ git branch second
```

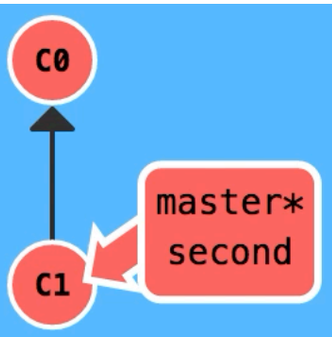


git checkout second

```
~/Desktop/our_first_repo  master ✔
» git checkout second
Switched to branch 'second'

~/Desktop/our_first_repo  second ✔
»
```

Изучаем ветвление в git

```
$ git branch second
```




```
$ git checkout second
```

В другой ветке – мы можем вести разработку другой версии нашего кода!

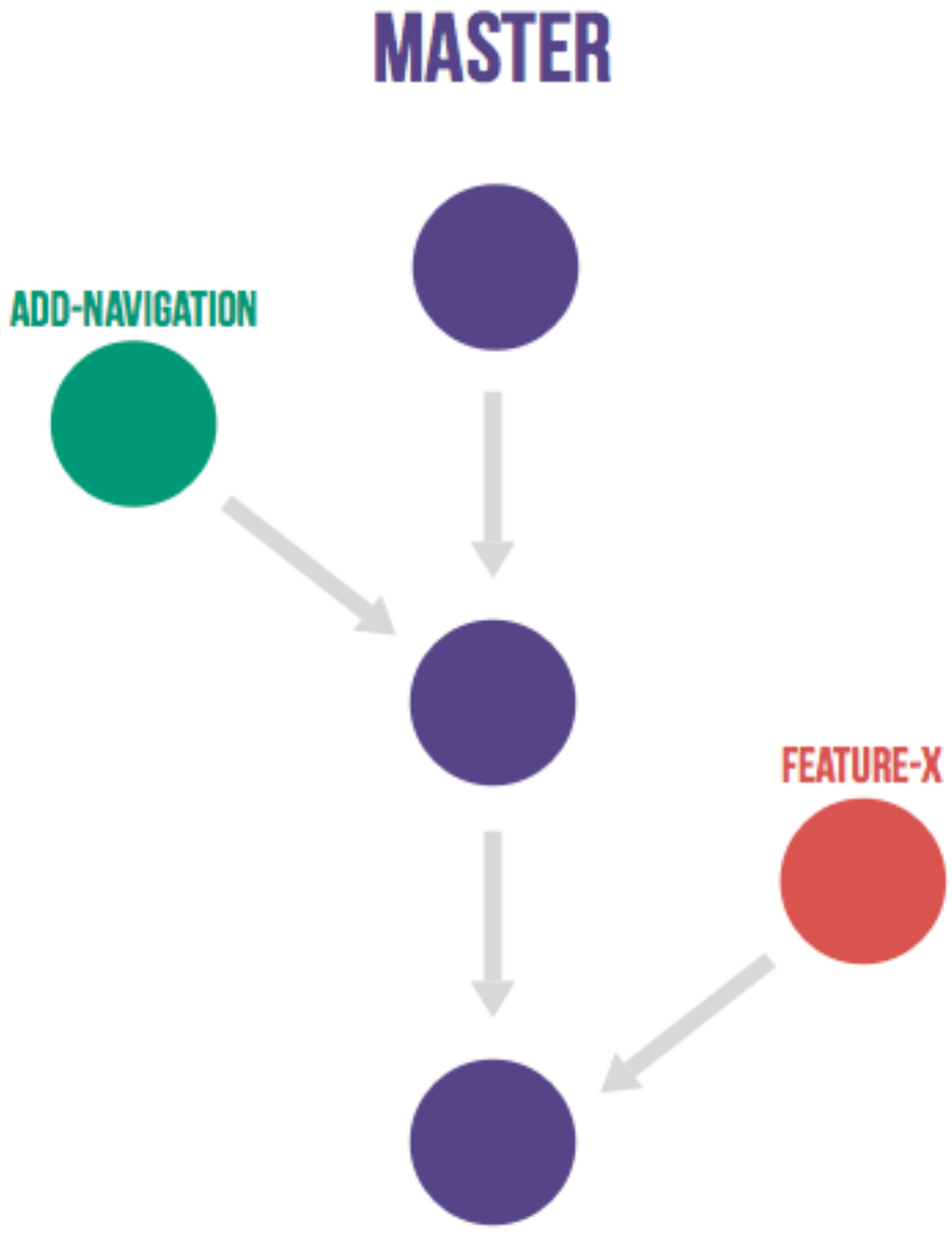
Что дает нам такое разделение?

- В рамках одного проекта может быть несколько под-версий проекта
- Изменения из одной ветки можно перетаскивать в другие
- Работать в команде с ветками – намного удобнее

Какие ветки бывают?

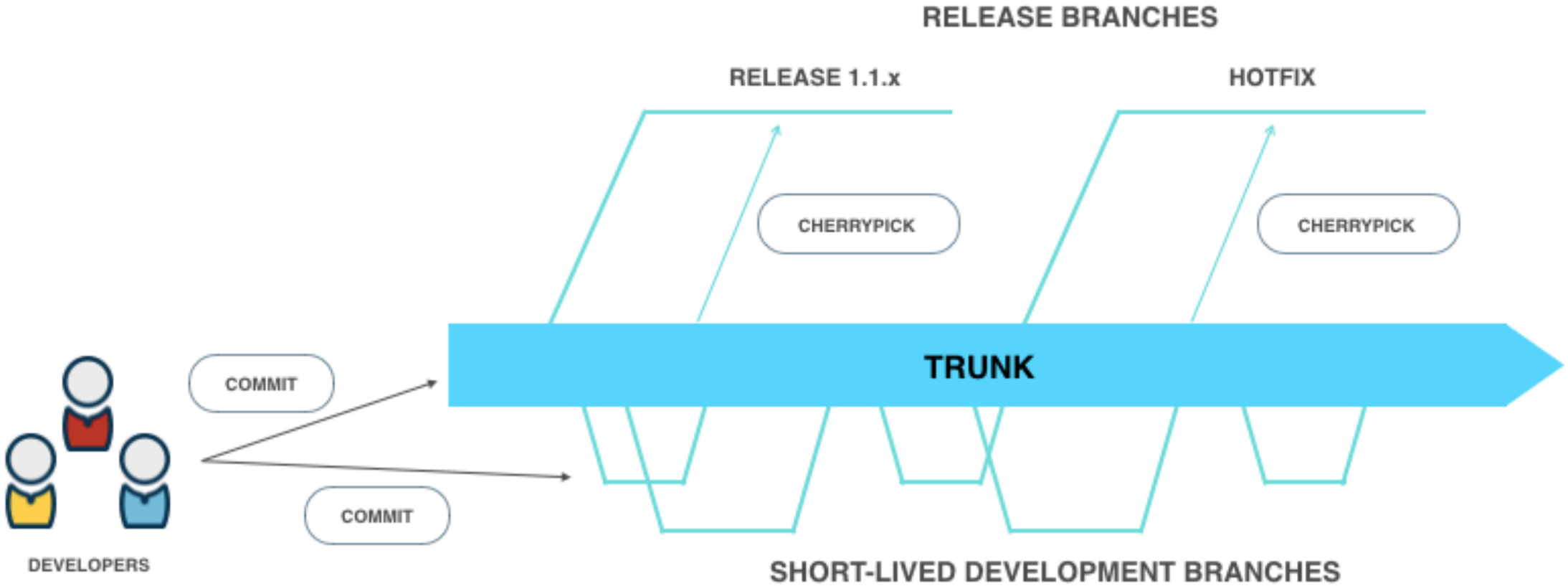
Долгоживущие ветки

Одноразовые ветки



Только одна ветка

CI - TRUNK BASED DEVELOPMENT



Создадим уникальное изменения для новой ветки!

Новое изменение в ветку second

```
11 +++ def new_function():
12 +++     print('I am new!')
```

Получим

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return first + second

def another_function(first):
    return first + 1

def new_function():
    print('I am new')
```

git add && git commit

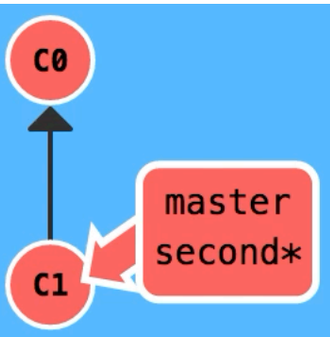
```
~/Desktop/our_first_repo  second ✗
» git status
On branch second
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mymodule.py

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

~/Desktop/our_first_repo  second ✗
» git add mymodule.py

~/Desktop/our_first_repo  second ✗                                    +
» git commit -m 'Adds new_function on branch second'
[second 1eb18cc] Adds new_function on branch second
 1 file changed, 3 insertions(+)

~/Desktop/our_first_repo  second ✔
»
```

git log

```
commit 1eb18cc965ca55da9f05c21507cf443cf30b7ecd (HEAD -> second)
Author: sobolevn <mail@sobolevn.me>
Date:   Fri Sep 25 22:43:52 2020 +0300

    Adds new_function on branch second

commit 9f8a60c2613602d46f975f5f400f3d83595da0ac (origin/master, origin/HEAD, master)
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 01:04:27 2020 +0300

    Create .gitignore

commit ed08b4b6f778ce148c86b1d3e1c32db5070792ad
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 00:53:21 2020 +0300

    Adds new_module.py from github

commit 74c351a615f9ee9ee0813d29d8e35582e0ca4cf2
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:16:31 2020 +0300

    Adds comments

commit b2af568974eaeea096e387020c4d974c3c6b529a
lines 1-25
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
```




```
$ git commit -m 'Adds new_function'
```

Проверим, что в master ничего не поменялось

git checkout master

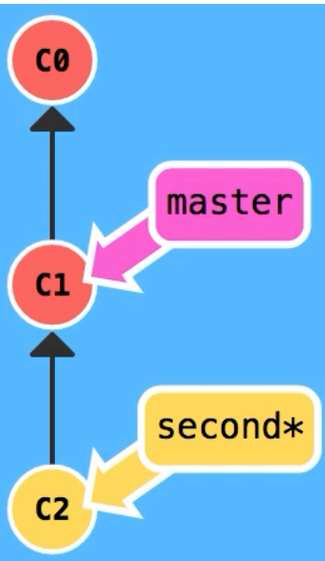
```
~/Desktop/our_first_repo  second ✔
» git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

~/Desktop/our_first_repo  master ✔
»
```

git log

```
commit 9f8a60c2613602d46f975f5f400f3d83595da0ac (HEAD -> master, origin/master, origin
/HEAD)
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 01:04:27 2020 +0300

    Create .gitignore

commit ed08b4b6f778ce148c86b1d3e1c32db5070792ad
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 00:53:21 2020 +0300

    Adds new_module.py from github

commit 74c351a615f9ee9ee0813d29d8e35582e0ca4cf2
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:16:31 2020 +0300

    Adds comments

commit b2af568974eaeea096e387020c4d974c3c6b529a
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:13:00 2020 +0300

    Adds another_function to mymodule.py

lines 1-24
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new_function'
```




```
$ git checkout master
```

Все осталось как было

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return first + second

def another_function(first):
    return first + 1
```

Теперь, внесем новое изменение в master!

Внесем второе изменение

```
5 def my_function(first, second):
6 ---    return first * second + 10
6 +++    return 5
```

Получим

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return 5

def another_function(first):
    return first + 1
```

git add && git commit

```
~/Desktop/our_first_repo  master ✗                                    +
» git add mymodule.py

~/Desktop/our_first_repo  master ✗                                    +
» git commit -m 'Now we return 5 from my_function'
[master 7dc6f0a] Now we return 5 from my_function
 1 file changed, 1 insertion(+), 1 deletion(-)

~/Desktop/our_first_repo  master ✔
»
```

git log

```
commit 7dc6f0ac6206768b3cc77a67a32e12a33f5983fc (HEAD -> master)
Author: sobolevn <mail@sobolevn.me>
Date:   Fri Sep 25 23:00:14 2020 +0300

    Now we return 5 from my_function

commit 9f8a60c2613602d46f975f5f400f3d83595da0ac (origin/master, origin/HEAD)
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 01:04:27 2020 +0300

    Create .gitignore

commit ed08b4b6f778ce148c86b1d3e1c32db5070792ad
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 00:53:21 2020 +0300

    Adds new_module.py from github

commit 74c351a615f9ee9ee0813d29d8e35582e0ca4cf2
Author: sobolevn <mail@sobolevn.me>
Date:   Wed Sep 23 00:16:31 2020 +0300

    Adds comments

commit b2af568974eaeea096e387020c4d974c3c6b529a
lines 1-25
```

git log

```
~/Desktop/our_first_repo  master ✔
» git log --graph --decorate --all --oneline
* 7dc6f0a (HEAD -> master) Now we return 5 from my_function
| * 1eb18cc (second) Adds new_function on branch second
|/
* 9f8a60c (origin/master, origin/HEAD) Create .gitignore
* ed08b4b Adds new_module.py from github
* 74c351a Adds comments
* b2af568 Adds another_function to mymodule.py
* 6387609 Initial commit

~/Desktop/our_first_repo  master ✔
»
```

Две версии кода

master

```python
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return 5

def another_function(first):
    return first + 1
```

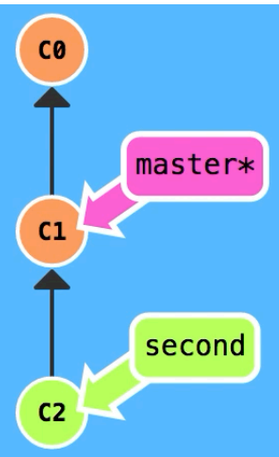
second

```python
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return first + second

def another_function(first):
    return first + 1

def new_function():
    print('I am new')
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new function'
$ git checkout master
$
```

```
$ git commit -m 'Now returns 5'
```



Слияние веток

git merge

```
~/Desktop/our_first_repo  master ✔
» git merge second -m 'Merging second into master'
Auto-merging mymodule.py
Merge made by the 'recursive' strategy.
 mymodule.py | 3 +++
 1 file changed, 3 insertions(+)

~/Desktop/our_first_repo  master ✔
»
```

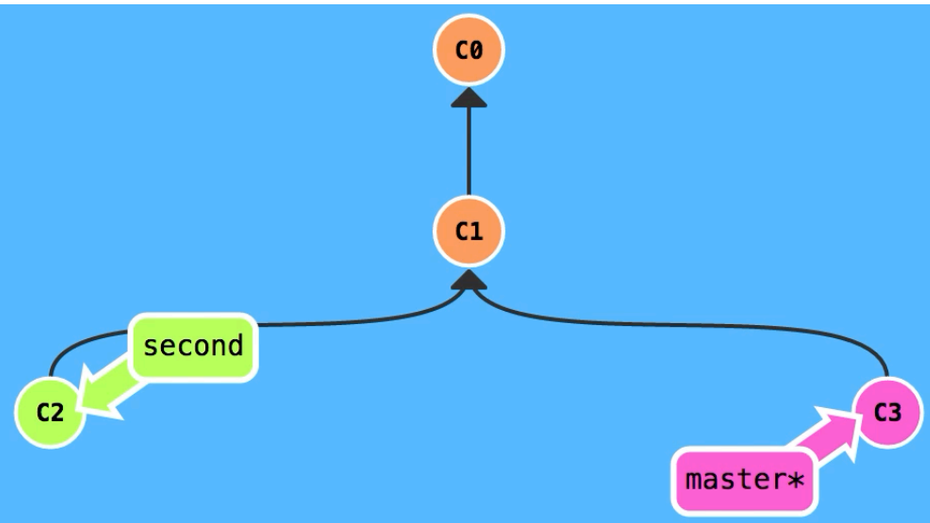
git log

```
commit de46b0e829ecde0aeae2769766b29a26e931f46b (HEAD -> master)
Merge: 7dc6f0a 1eb18cc
Author: sobolevn <mail@sobolevn.me>
Date:   Fri Sep 25 23:06:54 2020 +0300

    Merging second into master

commit 7dc6f0ac6206768b3cc77a67a32e12a33f5983fc
Author: sobolevn <mail@sobolevn.me>
Date:   Fri Sep 25 23:00:14 2020 +0300

    Now we return 5 from my_function

commit 1eb18cc965ca55da9f05c21507cf443cf30b7ecd (second)
Author: sobolevn <mail@sobolevn.me>
Date:   Fri Sep 25 22:43:52 2020 +0300

    Adds new_function on branch second

commit 9f8a60c2613602d46f975f5f400f3d83595da0ac (origin/master, origin/HEAD)
Author: Nikita Sobolev <mail@sobolevn.me>
Date:   Wed Sep 23 01:04:27 2020 +0300

    Create .gitignore

lines 1-25
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new function'
$ git checkout master
$
$ git commit -m 'Now returns 5'
```

```
$ git merge second
```

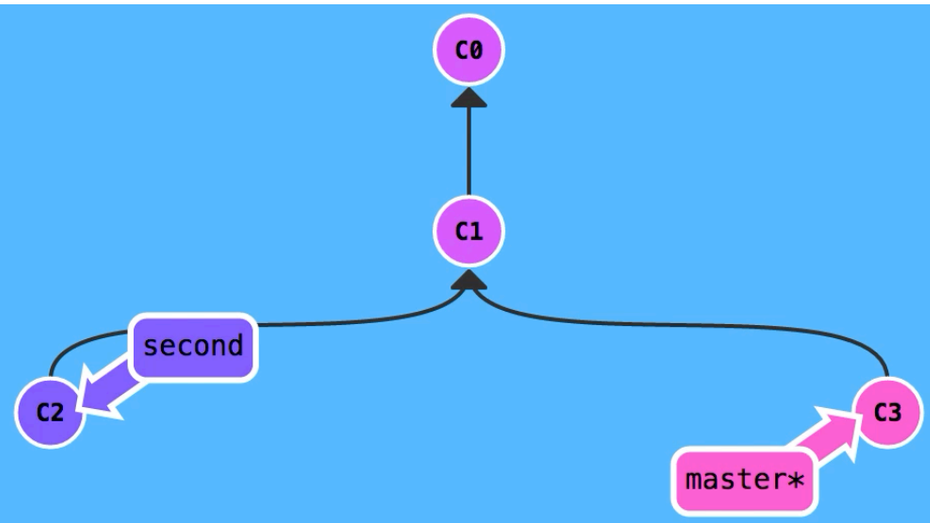


Или можно использовать "git rebase"

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new_function'
$ git checkout master
$ git commit -m 'Now returns 5'
```

```
$ git rebase second
```



В чем разница?

- "git merge" сохраняет ветвление, создает еще один дополнительный коммит

- "git rebase" делает структуру плоской, не создает дополнительный коммит

Было две версии кода

master

```python
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return 5

def another_function(first):
    return first + 1
```

second

```python
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return first + second

def another_function(first):
    return first + 1

def new_function():
    print('I am new')
```

Стала одна общая

master

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return 5

def another_function(first):
    return first + 1

def new_function():
    print('I am new')
```

А что если конфликт?

git потребует, чтобы мы его разрешили

Варианты разрешения

- Перезатри "мои" изменения "их" изменениями
- Перезатри "их" изменения "моими" изменениями
- Позволь мне руками сделать правильную версию из двух разных кусков

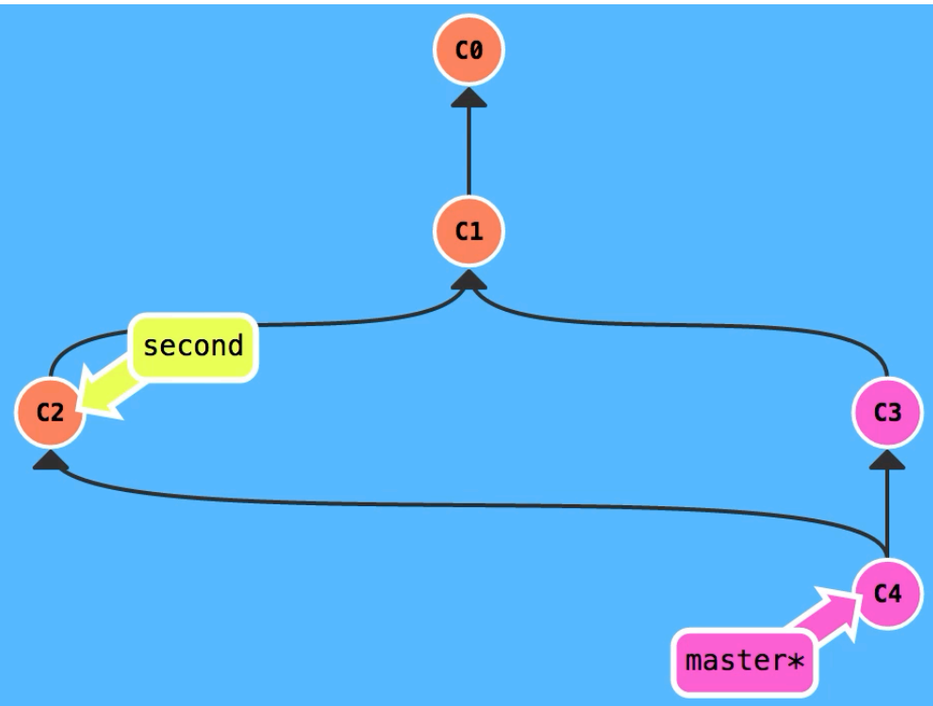
Создадим конфликт!

git checkout -b

```
~/Desktop/our_first_repo  master ✔
» git checkout -b third
Switched to a new branch 'third'

~/Desktop/our_first_repo  third ✔
»
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new_function'
$ git checkout master
$ git commit -m 'Now returns 5'
$ git merge second
```




```
$ git checkout -b third
```

Внесем третье изменение

third

```
5 def my_function(first, second):
6 ---    return 5
6 +++    return 5 + 1
```

Итого

third

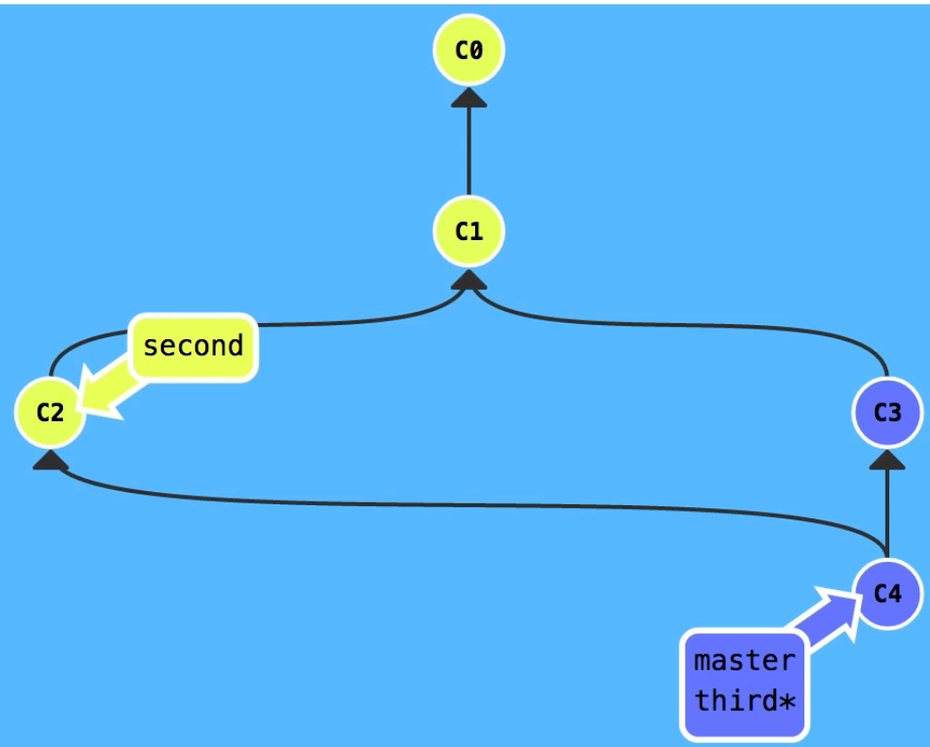
```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return 5 + 1

def another_function(first):
    return first + 1

def new_function():
    print('I am new')
```

git commit

```
~/Desktop/our_first_repo  third ✗
» git add mymodule.py

~/Desktop/our_first_repo  third ✗                                    +
» git commit -m 'Adds new +1'
[third 0094fd3] Adds new +1
 1 file changed, 1 insertion(+), 1 deletion(-)

~/Desktop/our_first_repo  third ✔
»
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new_function'
$ git checkout master
$ git commit -m 'Now returns
5'
$ git merge second
$ git checkout -b third
```




```
$ git commit -m 'Adds new +1'
```

Теперь вернемся в master и сделаем что-нибудь противоположное

git checkout master

```
~/Desktop/our_first_repo  third ✔
» git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

~/Desktop/our_first_repo  master ✔
»
```

Внесем третье* изменение

master

```
5 def my_function(first, second):
6 ---    return 5
6 +++    return 5 - 2
```

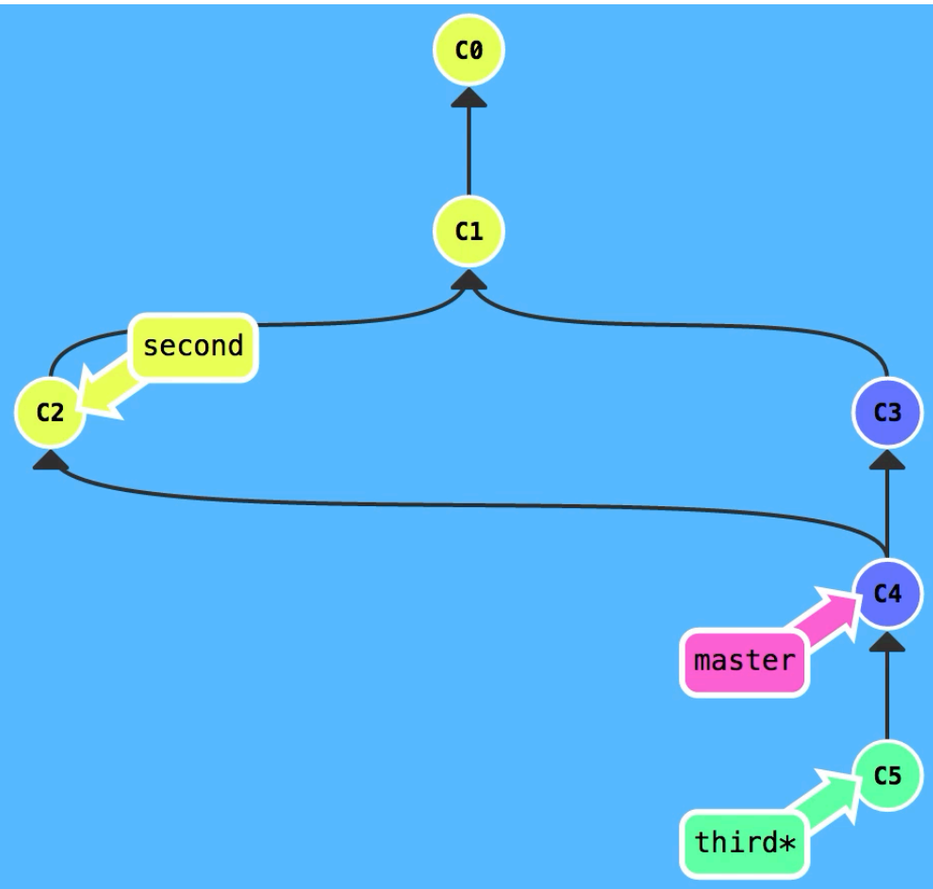
git commit

```
~/Desktop/our_first_repo  master ✗
» git add mymodule.py

~/Desktop/our_first_repo  master ✗                                    +
» git commit -m 'Adds new -2'
[master 40625f3] Adds new -2
 1 file changed, 1 insertion(+), 1 deletion(-)

~/Desktop/our_first_repo  master ✔
»
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new_function'
$ git checkout master
$ git commit -m 'Now returns
5'
$ git merge second
$ git checkout -b third
$ git commit -m 'Adds new +1'
```

```
$ git checkout master
```



Две версии кода

master

```
1 # Previously I had some complex logic in this
2 # module, but I had to delete it due to reasons.
3 # Very sad!
4
5 def my_function(first, second):
6     return 5 - 2
7
8 def another_function(first):
9     return first + 1
10
11 def new_function():
12     print('I am new')
```

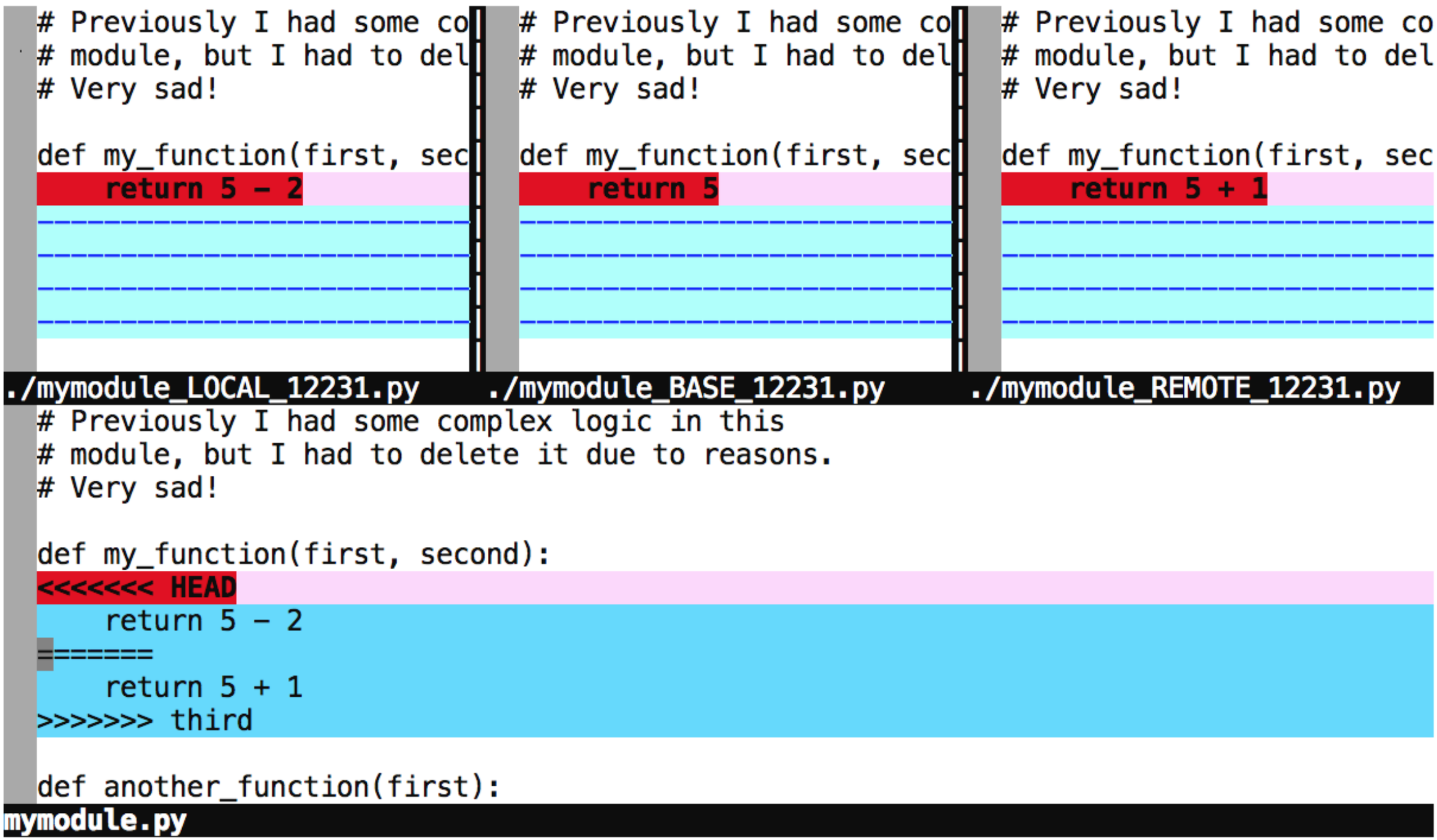
third

```
1 # Previously I had some complex logic in this
2 # module, but I had to delete it due to reasons.
3 # Very sad!
4
5 def my_function(first, second):
6     return 5 + 1
7
8 def another_function(first):
9     return first + 1
10
11 def new_function():
12     print('I am new')
```

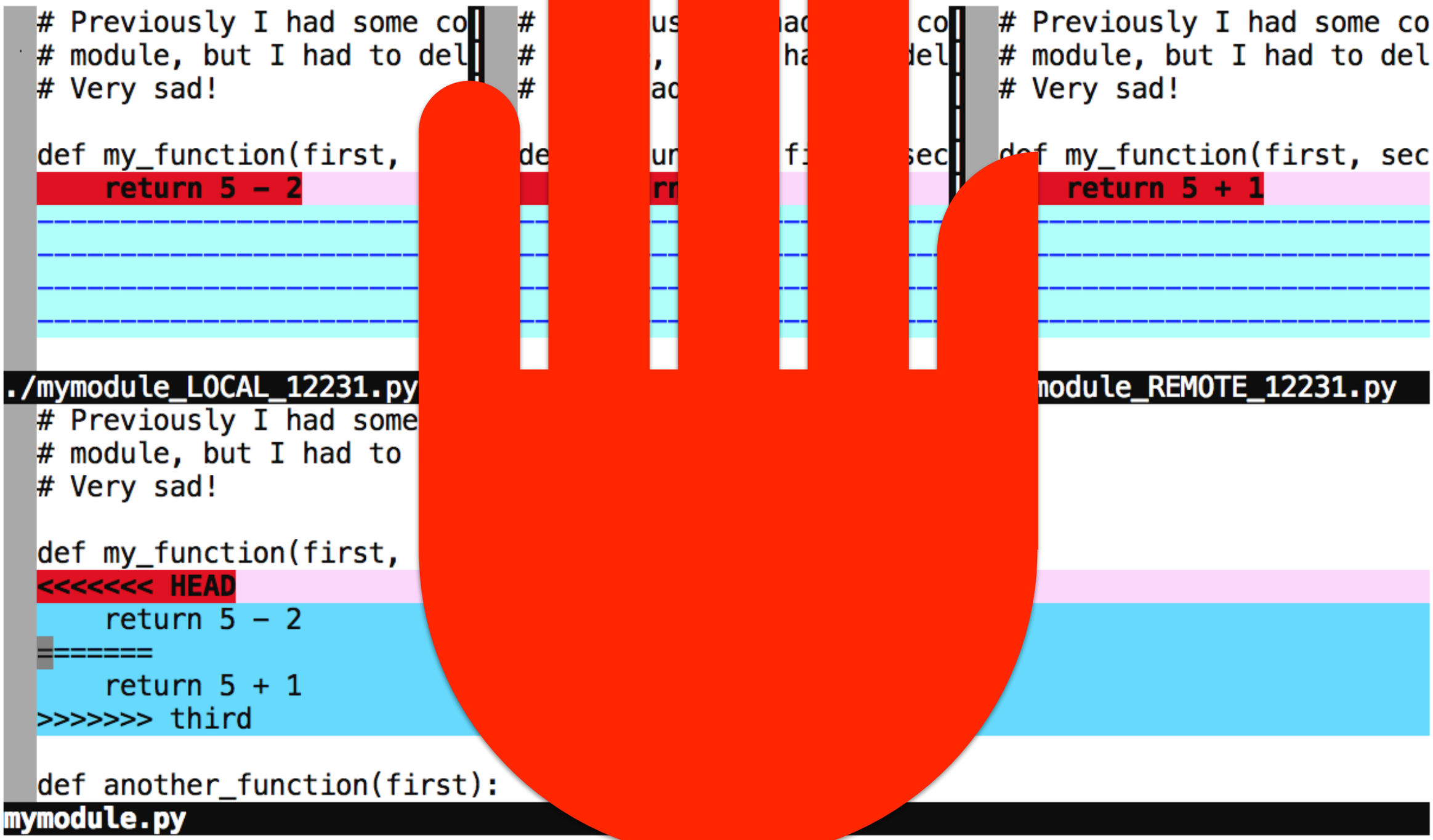
git merge

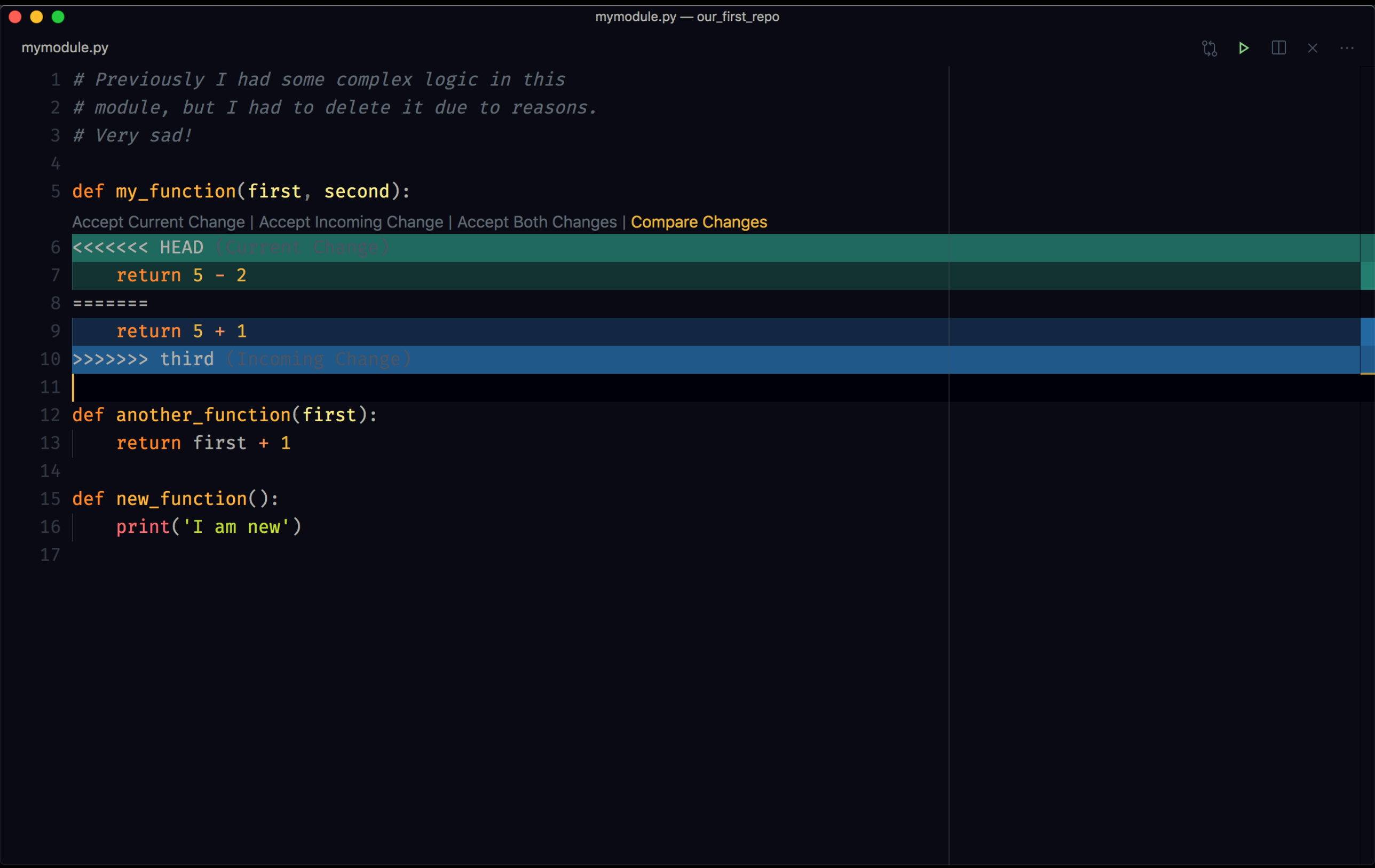
```
~/Desktop/our_first_repo  master ✔
» git merge third
Auto-merging mymodule.py
CONFLICT (content): Merge conflict in mymodule.py
Recorded preimage for 'mymodule.py'
Automatic merge failed; fix conflicts and then commit the result.

~/Desktop/our_first_repo  master|MERGING ✗                          § 1 ⚠
»
```

git mergetool



git mergetool




mymodule.py — our_first_repo
mymodule.py
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!
def my_function(first, second):
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
return 5 - 2
=======
return 5 + 1
>>>>>>> third (Incoming Change)
def another_function(first):
return first + 1
def new_function():
print('I am new')

Теперь поправим конфликт

Стала одна общая

master

```
# Previously I had some complex logic in this
# module, but I had to delete it due to reasons.
# Very sad!

def my_function(first, second):
    return 4

def another_function(first):
    return first + 1

def new_function():
    print('I am new')
```

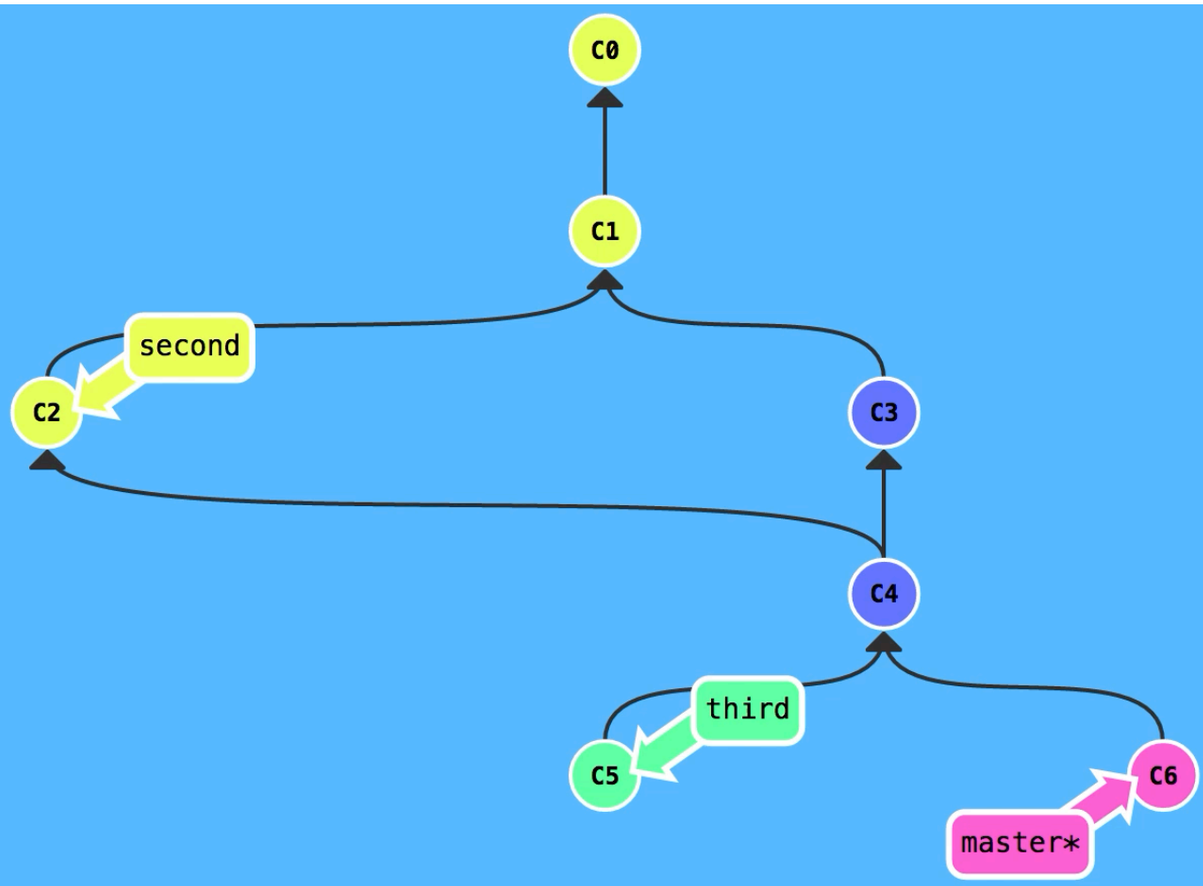
git commit

```
~/Desktop/our_first_repo  master|MERGING ✗                                +
» git add mymodule.py

~/Desktop/our_first_repo  master|MERGING ✗                                +
» git commit -m 'Merging third into master'
Recorded resolution for 'mymodule.py'.
[master 544462a] Merging third into master

~/Desktop/our_first_repo  master ✗
»
```

Изучаем ветвление в git

```
$ git branch second
$ git checkout second
$ git commit -m 'Adds new_function'
$ git checkout master
$ git commit -m 'Now returns 5'
$ git merge second
$ git checkout -b third
$ git commit -m 'Adds new +1'
$ git checkout master
$ git commit -m 'Adds new -2'
```

```
$ git merge third
```



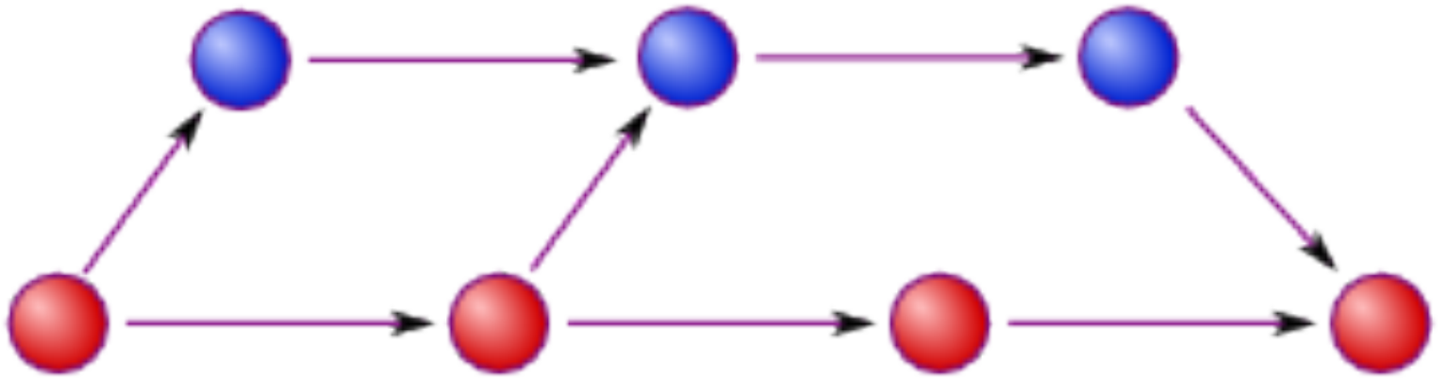
Готово!

https://github.com/sobolevn/our_first_repo

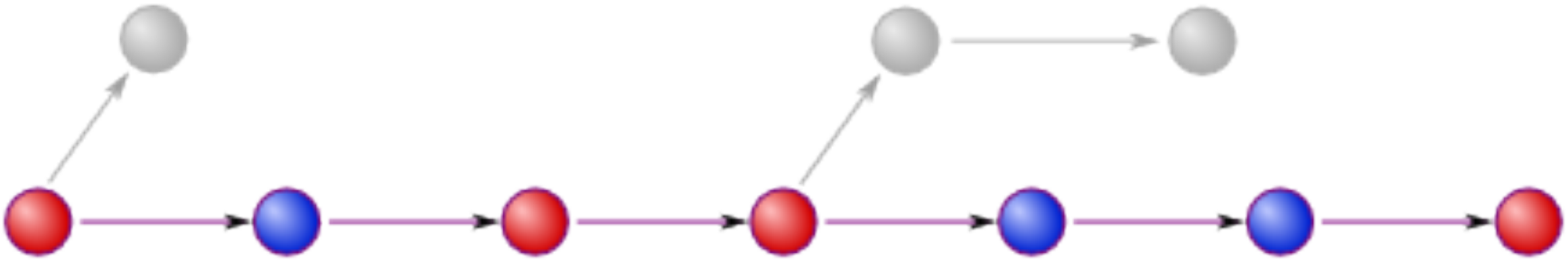
Полезности

git pull --rebase

Loopy history (git pull):



Nice history (git pull --rebase):



git cherry-pick

Изучаем ветвление в git

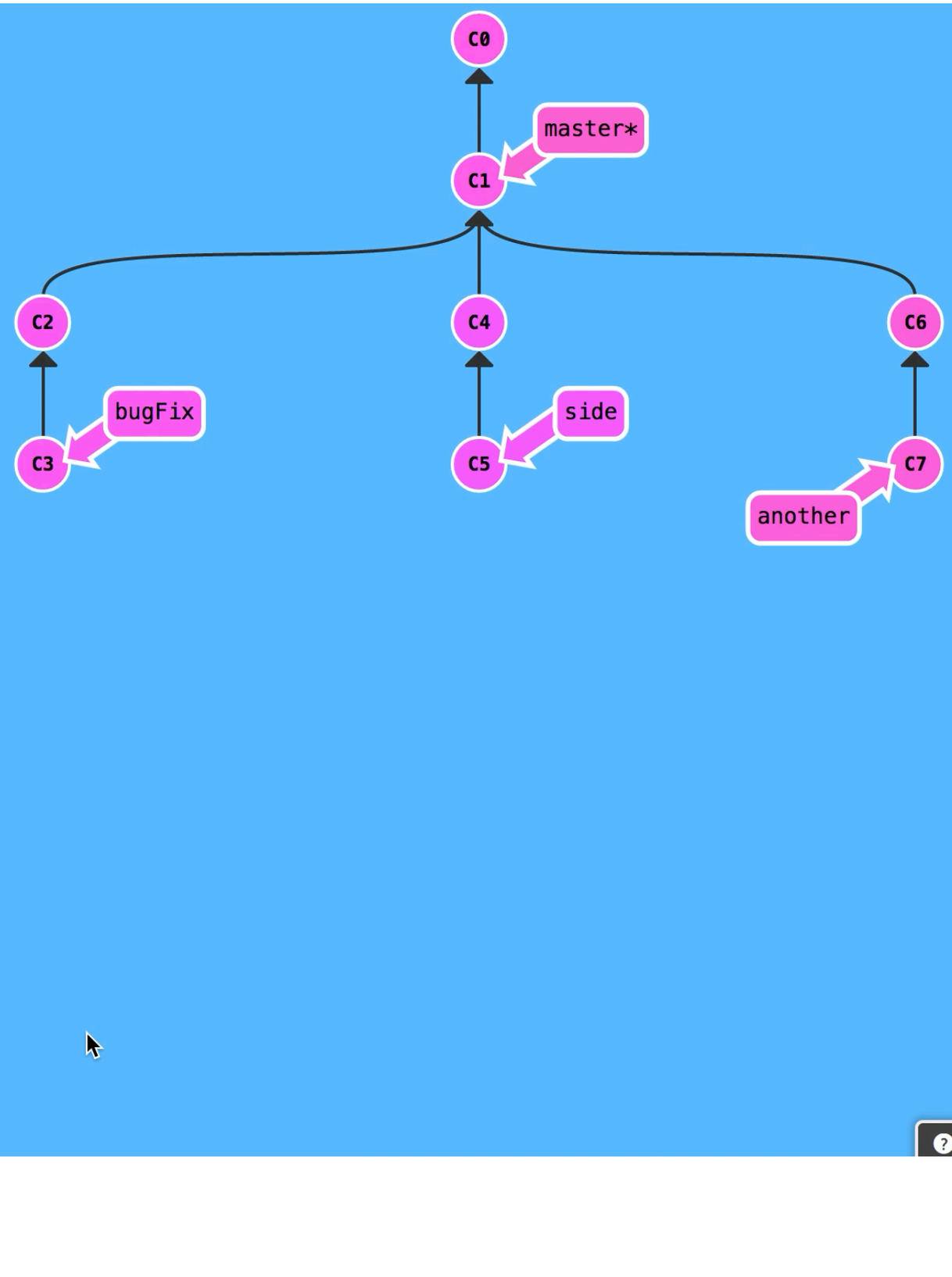
Цель уровня

Level ***Введение в Cherry-pick***

Задача

```
$ clear
```

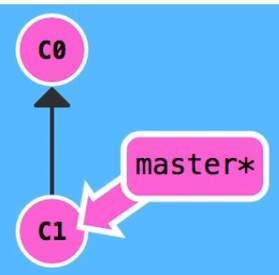
```
$ git cherry-pick C3 C4 C7
```



Fork me on GitHub

git revert

Изучаем ветвление в git



Fork me on GitHub

```
$ git revert C1
```

git config

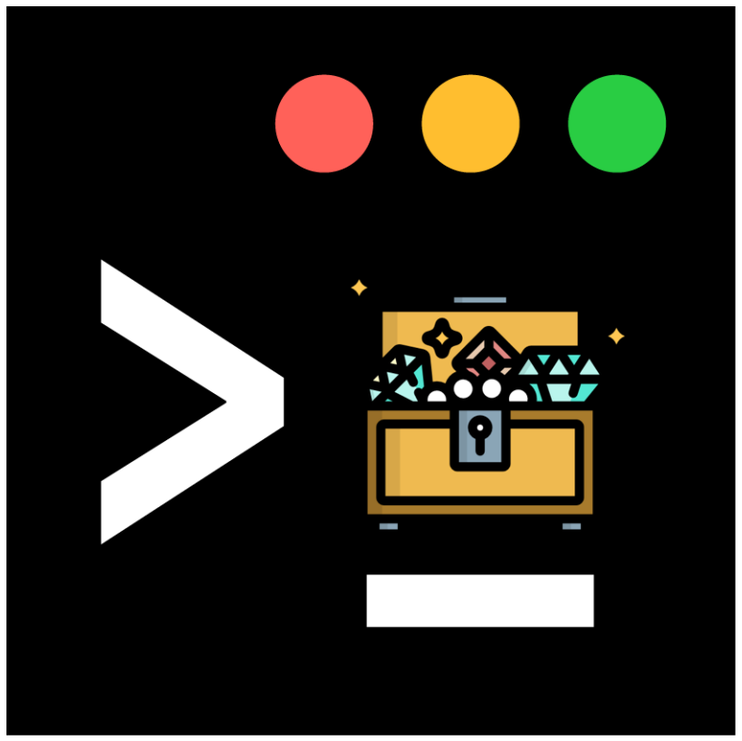
github.com/sobolevn/dotfiles/blob/master/config/gitconfig

Без настроек

```
~/Desktop/our_first_repo  master ✗
» git diff
diff --git mymodule.py mymodule.py
index 8f1cdeb..da82341 100644
--- mymodule.py
+++ mymodule.py
@@ -3,7 +3,7 @@
 # Very sad!

 def my_function(first, second):
-    return 4
+    return MY_CONST

 def another_function(first):
     return first + 1

~/Desktop/our_first_repo  master ✗
»
```

С правильными настройками

```
~/Desktop/our_first_repo  master ✗
» git diff

mymodule.py

  3  :  3   # Very sad!
  4  :  4
  5  :  5   def my_function(first, second):
  6  :        return 4
     :  6     return MY_CONST
  7  :  7
  8  :  8   def another_function(first):
  9  :  9       return first + 1

~/Desktop/our_first_repo  master ✗
»
```



github.com/
sobolevn/dotfiles

Плагины!

github.com/stevemao/awesome-git-addons

Хранение секретных файлов

github.com/sobolevn/git-secret

Дополнительные материалы

- https://stackoverflow.com/questions/57265785/whats-the-difference-between-git-switch-and-git-checkout-branch

- https://git-scm.com/book/en/v2/Git-Tools-Reset-Demystified

- https://www.atlassian.com/ru/git/tutorials/rewriting-history

Онлайн курсы

- https://lab.github.com/githubtraining/introduction-to-github

- https://git-rebase.io/

Выводы

- Для работы в команде – потребуется создавать ветки, чтобы удобно работать с несколькими версиями кода
- В ветках могут быть конфликты, их бывает легко решить, а бывает – сложно
- Вокруг git'а есть очень много удобных инструментов