

git

Что такое Git и зачем он нужен?

Git - это консольная утилита, для отслеживания и ведения истории изменения файлов, в вашем проекте. Чаще всего его используют для кода, но можно и для других файлов.

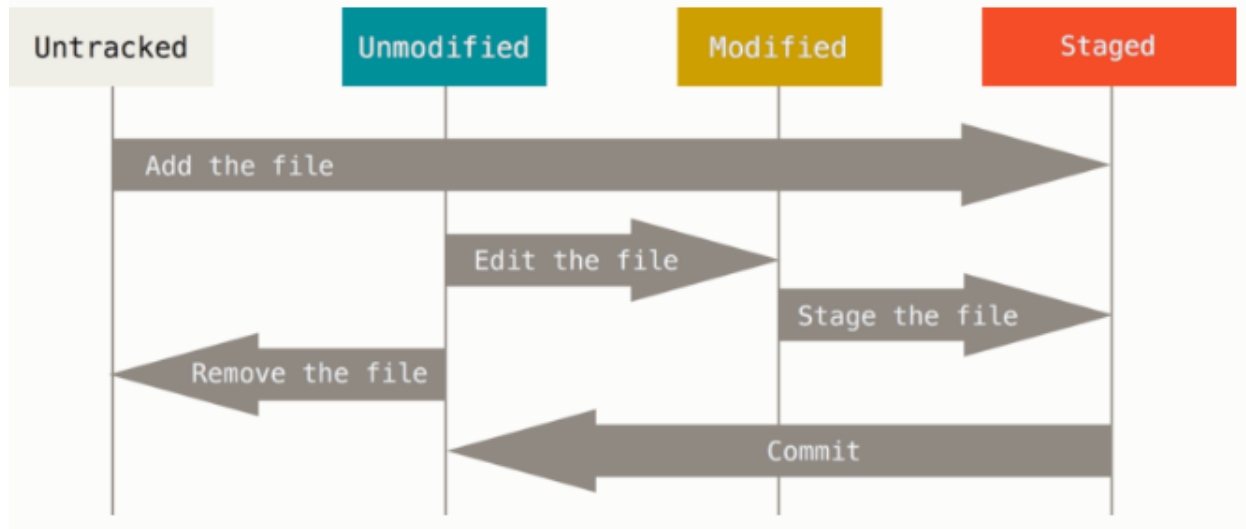
С помощью Git-а вы можете откатить свой проект до более старой версии, сравнивать, анализировать или сливать свои изменения в репозиторий.

Репозиторием называют хранилище вашего кода и историю его изменений. Git работает локально и все ваши репозитории хранятся в определенных папках на жестком диске.

Так же ваши репозитории можно хранить и в интернете. Обычно для этого используют три сервиса:

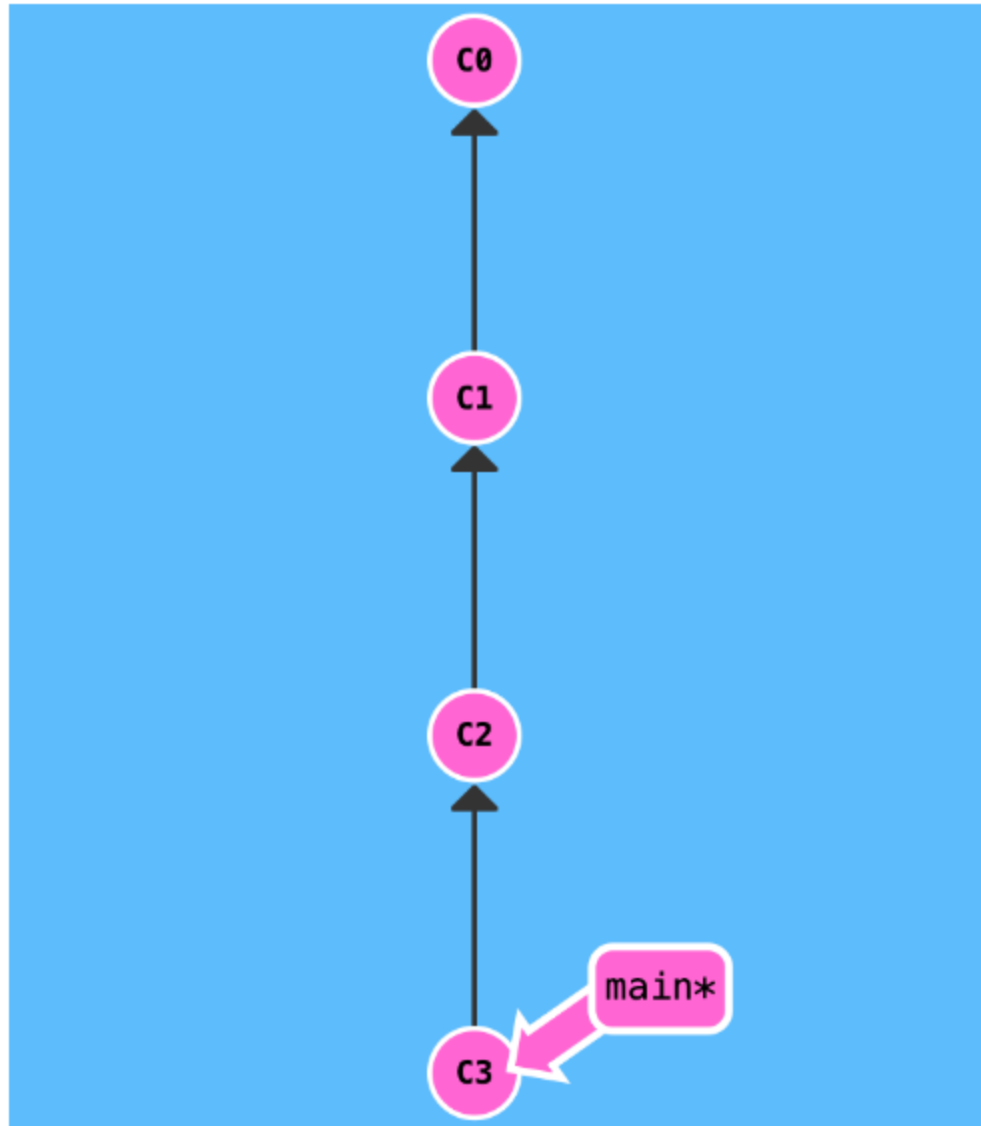
- GitHub
- Bitbucket
- GitLab

Как только вы отредактируете файлы, Git будет рассматривать их как изменённые, так как вы изменили их с момента последнего коммита. Вы индексируете эти изменения, затем фиксируете все проиндексированные изменения, а затем цикл повторяется.

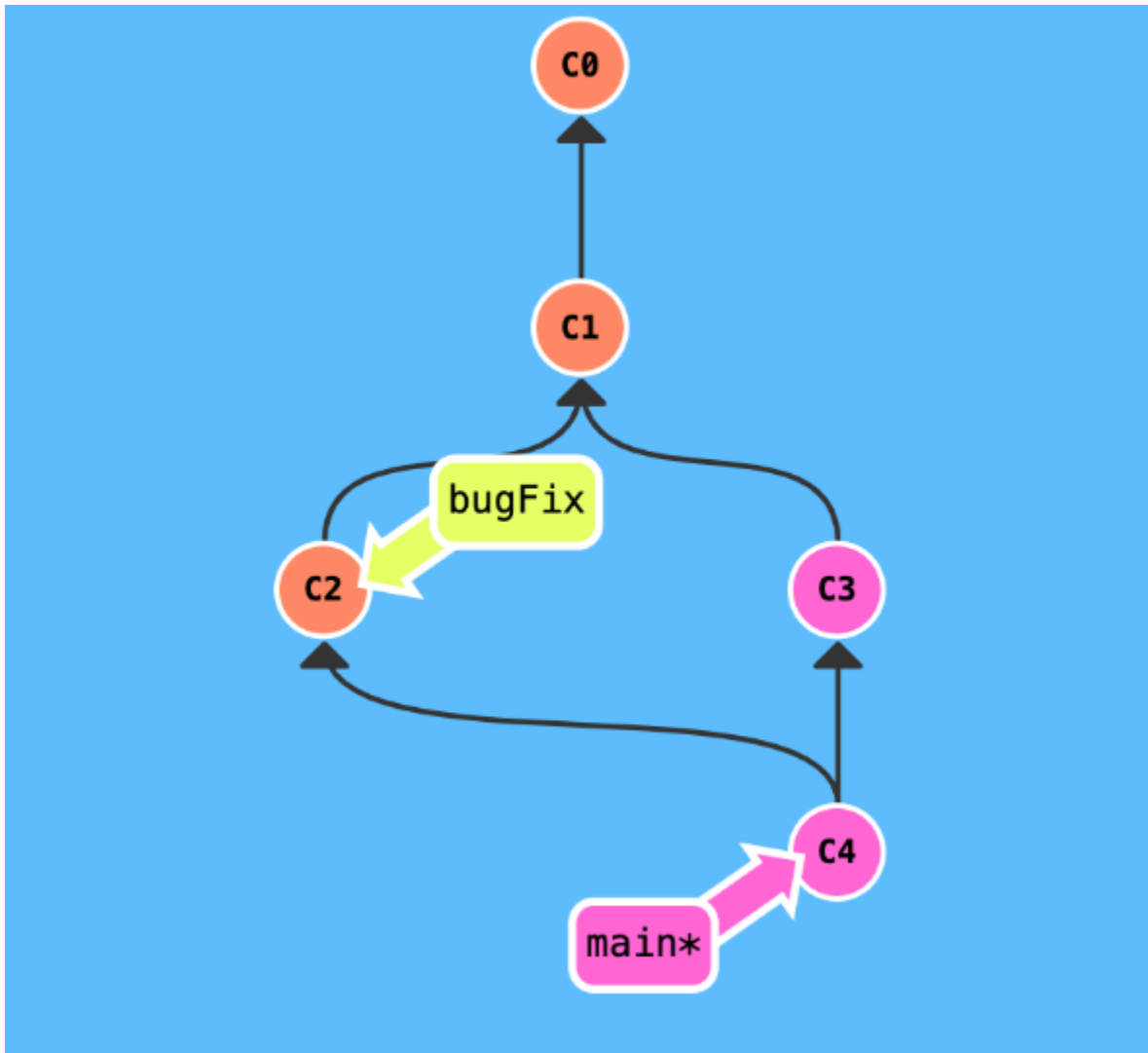


Что такое ветки и как это работает?

Ветка - это история изменений. У каждой ветки есть свое название. Репозиторий может содержать в себе несколько веток, которые создаются из других веток или вливаются в них.



В итоге получается очень простой граф, состоящий из одной ветки (master) и четырех commit. Все это может превратиться в более сложный граф, состоящий из нескольких веток, которые сливаются в одну.



Настройка git

#Установим имя для вашего пользователя

```
git config --global user.name "artem"
```

#Теперь установим email.

```
git config --global user.email "sdgsdga@gmail.com"
```

Основные команды git

```
git help # справка по всем командам
git clone
git status
git branch
git checkout
git merge
git push
git pull
```



При работе с системами контроля версий разработчики сталкиваются с определенной, повторяющейся последовательностью действий. Оно и понятно, ведь, по сути, если не брать в расчет возможности Git для управления состоянием

проекта и прочие плюшки, то как правило ваша работа ограничена рядом действий:

1. Внести изменения в проект;
2. Добавить изменения в индекс(staging area) - `git add` (таким образом вы сообщаете Git какие именно изменения должны быть занесены в историю.)
2. Закоммитить изменения - `git commit -m` (сохранить изменения в историю проекта)
3. Запустить - `git push` (отправить результаты работы на удаленный сервер, чтобы другие разработчики тоже имели к ним доступ)

Команда `git status`

отображает состояние директории и индекса(staging area). Это позволяет определить, какие файлы в проекте отслеживаются Git, а также какие изменения будут включены в следующий коммит.

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ae39b03a-0ef6-430e-bcb5-8950eb75eb9d/git.pdf>