



Системы контроля версий



git



mercurial



Что такое контроль версий?

Система управления версиями (также используется определение «система контроля версий», от [англ. Version Control System, VCS](#) или *Revision Control System*) — [программное обеспечение](#) для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. (https://ru.wikipedia.org/wiki/Система_управления_версиями)



Обзор систем контроля версий

<https://biz30.timedoctor.com/ru/система-контроля-версий/>



Git



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.



Репозиторий (Repository)

Репозиторием называют хранилище вашего кода и историю его изменений.
(<https://habr.com/ru/post/541258/>)

Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов, и хранилище, содержащее собственно файлы.
(<https://ru.wikipedia.org/wiki/Git>)



Коммит (Commit)

Коммит — это *снимок* ваших локальных файлов, записанный в локальный репозиторий.

(https://github.com/RomuloOliveira/commit-messages-guide/blob/master/README_ru-RU.md)

Каждому коммиту присваивается уникальный номер вида:

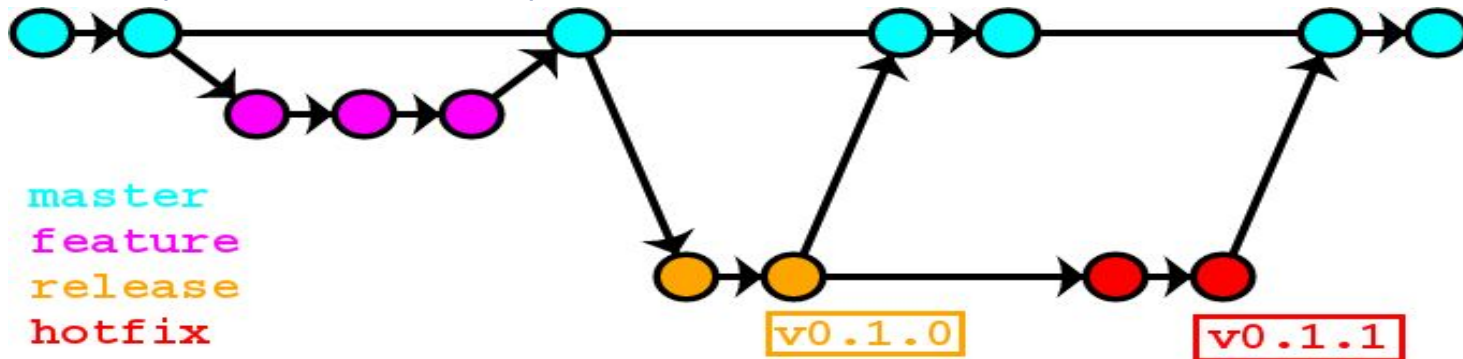
17f38c20afeb22274b0826a8dfd60a6d7699e779

Прим.

git commit -m "Комментарий к коммиту"

Ветка

Ветка в Git это подвижный указатель на один из коммитов. Обычно ветка указывает на последний коммит в цепочке коммитов. Ветка берет свое начало от какого-то одного коммита. Визуально это можно представить вот так. (<https://habr.com/ru/post/342116/>)





Операции с Git

!Все операции с git выполняются в рабочем каталоге (в папке с проектом)!

- Создание репозитория - **git init**
- Добавление файлов проекта в индекс - **git add наименование_файла**
 - **git add .** - добавить все файлы из рабочего каталога
- Фиксация изменений - **git commit**



Операции с Git

- Создание ветки - **git branch feature_1**
- Создание ветки с автоматическим переключением на неё -

git checkout -b feature_1

- Просмотр истории коммитов на текущей ветке -

git log



Удаленный репозиторий

- GitHub
- GitLab
- BitBucket



Работа с удаленным репозиторием Git

- Клонировать проект с удаленного репозитория в локальный - **git clone**
<https://github.com/schacon/ticgit>
- Добавление удаленного репозитория к текущему проекту - **git remote**
add origin <https://github.com/schacon/ticgit>
- Получение обновлений из удаленного репозитория - **git pull**
- Отправка изменений на удаленный репозиторий - **git push origin master**