

# С++ Лекция 1 Git

# Git

Git - система контроля версий:

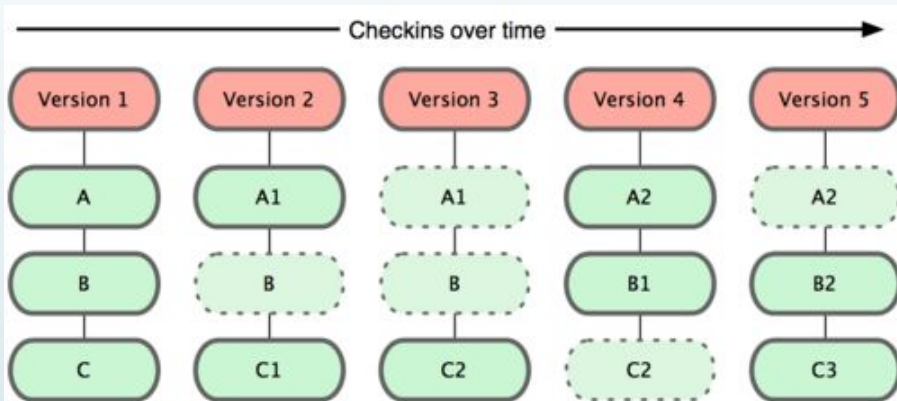
- Отслеживает изменения
- Фиксирует изменения
- Позволяет откатывать изменения

- Почти все операции локальные
- Git следит за целостностью файлов

# Хранение изменений

При коммите Git сохраняет слепок того, как выглядят все файлы проекта на текущий момент.

Если файл не менялся, Git не сохраняет файл снова, а делает ссылку на ранее сохранённый файл.



# Три состояния

- Файл зафиксирован
- Файл изменен
- Файл подготовлен

```
claorisel@claorisel ~/workspace/tmp $ git add fn1.cpp
claorisel@claorisel ~/workspace/tmp $ git status
On branch master

Initial commit

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

        new file:   fn1.cpp

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

        fn2.cpp
        fn3.cpp
```

# Git Workflow

- Внесение изменений в файлы в своём рабочем каталоге.
- Подготовка файлов, добавление их слепков в область подготовленных файлов.
- Коммит, который берёт подготовленные файлы из индекса и помещает их в каталог Git'a на постоянное хранение.

# Git настройка

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

```
$ git config --global core.editor nano
```

```
claorisel@claorisel ~/workspace/tmp $ git config --list  
user.email=claorisel@gmail.com  
user.name=claorisel  
core.repositoryformatversion=0  
core.filemode=true  
core.bare=false  
core.logallrefupdates=true  
claorisel@claorisel ~/workspace/tmp $
```

# Git init

```
claorisel@claorisel ~/workspace/tmp $ git init
Initialized empty Git repository in /home/claorisel/workspace/tmp/.git/
claorisel@claorisel ~/workspace/tmp $ git status
On branch master

Initial commit

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

    main.cpp
    stfile1.cpp
    stfile2.cpp

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

# Git commit

```
claorisel@claorisel ~/workspace/tmp $ git add main.cpp
claorisel@claorisel ~/workspace/tmp $ git status
On branch master

Initial commit

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

        new file:   main.cpp

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

        stfile1.cpp
        stfile2.cpp

claorisel@claorisel ~/workspace/tmp $ git commit -m "first commit"
[master (root-commit) d4ce842] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.cpp
claorisel@claorisel ~/workspace/tmp $
```



# Git clone

```
claorisel@claorisel ~/workspace/tmp $ git clone https://github.com/pyqtgraph/pyqtgraph.git
Cloning into 'pyqtgraph'...
remote: Counting objects: 10661, done.
remote: Compressing objects: 100% (105/105), done.
remote: Total 10661 (delta 89), reused 97 (delta 58), pack-reused 10497
Receiving objects: 100% (10661/10661), 4.16 MiB | 3.84 MiB/s, done.
Resolving deltas: 100% (7847/7847), done.
Checking connectivity... done.
```

# Git file status

Каждый файл в вашем рабочем каталоге может находиться в одном из состояний:

- **Отслеживаемые:**
  - Неизменённые
  - Изменённые
  - Подготовленные к коммиту (staged)
- **Неотслеживаемые**

```
$ git status
```

Когда вы впервые клонируете репозиторий, все файлы будут отслеживаемыми и неизменёнными, потому что вы только взяли их из хранилища и ничего пока не редактировали.

# Git remote

```
claorisel@claorisel ~/workspace/tmp $ cd pyqtgraph/  
claorisel@claorisel ~/workspace/tmp/pyqtgraph $ git remote -v  
origin https://github.com/pyqtgraph/pyqtgraph.git (fetch)  
origin https://github.com/pyqtgraph/pyqtgraph.git (push)  
claorisel@claorisel ~/workspace/tmp/pyqtgraph $
```

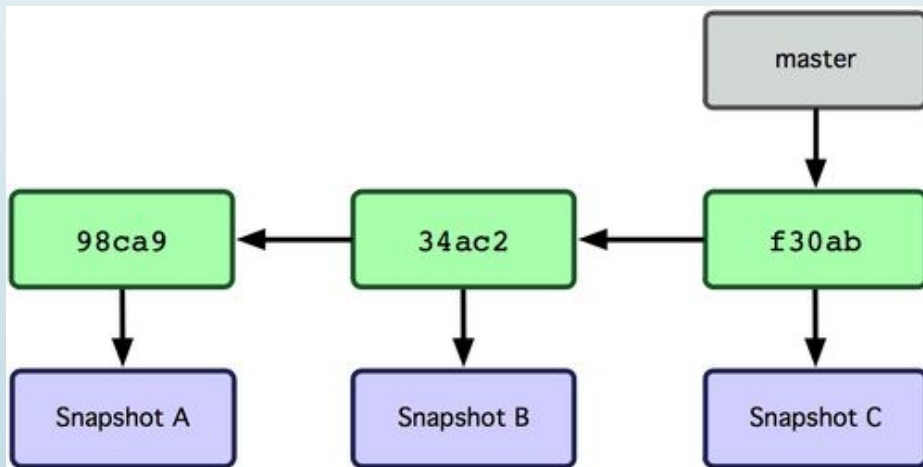
```
$ git pull origin master  
$ git push origin master
```

# Git alias

```
$ git config --global alias.co checkout  
$ git config --global alias.br branch  
$ git config --global alias.ci commit  
$ git config --global alias.st status
```

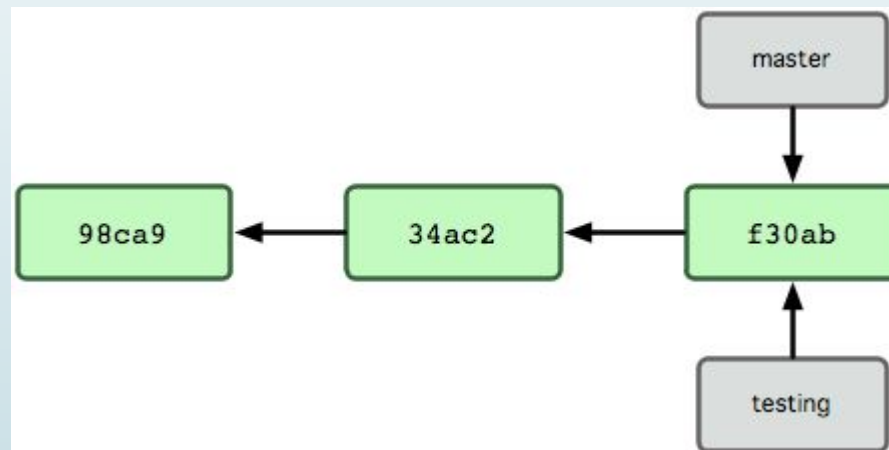
# Git branch

- Ветка по умолчанию в Git'e называется **master**.
- Когда вы создаёте коммиты на начальном этапе, вам дана ветка **master**, указывающая на последний сделанный коммит.
- При каждом новом коммите она сдвигается вперёд автоматически.



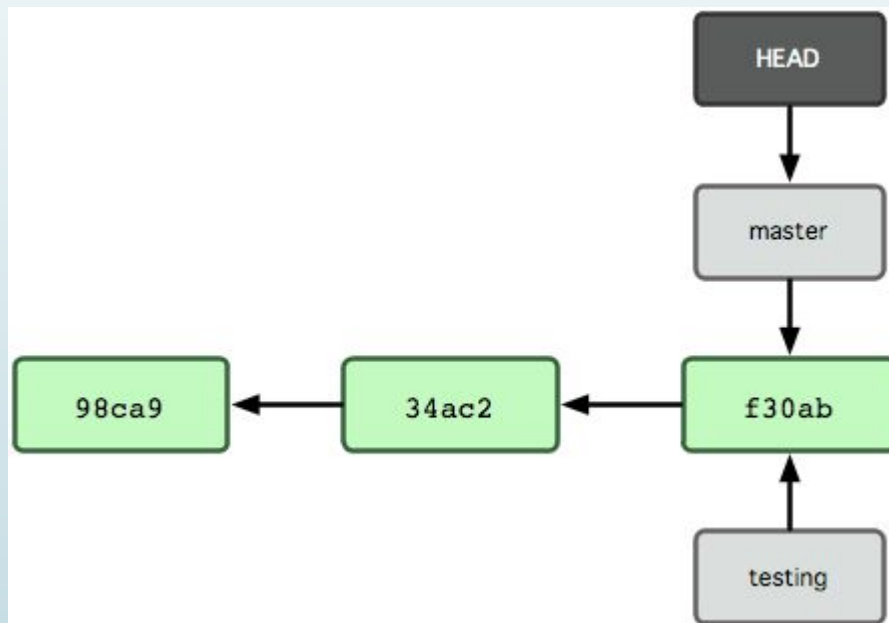
# Git branch

```
$ git branch testing
```



# Git branch

**HEAD** указатель на локальную ветку, на которой вы находитесь



# Git branch

\$ git checkout testing

