

# Git: основные команды, состояния Git

Иманов Рахман

# Что такое Git

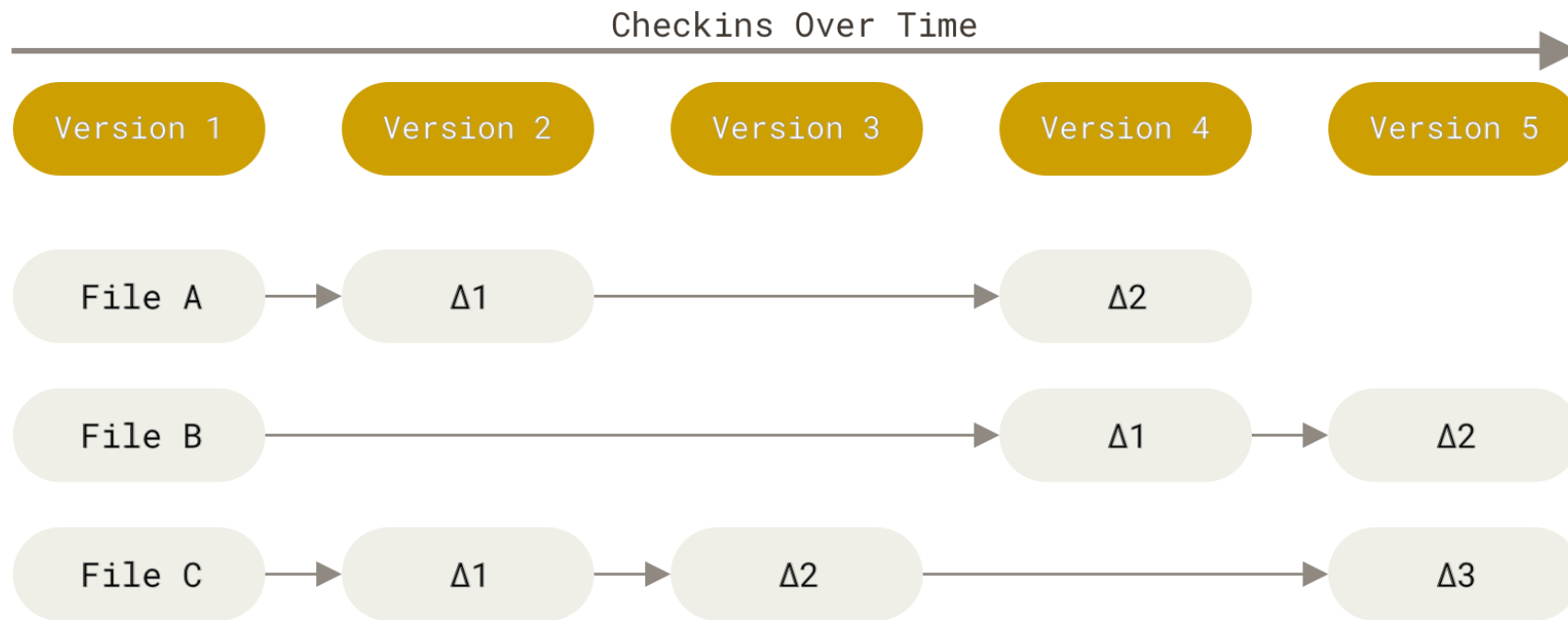
Git – система контроля версий с открытым исходным кодом.

Репозиторий – хранилище кода и истории его изменений

Коммит – некоторая точка сохранения проекта с уникальным идентификатором и комментарием.

# Хранение файлов и изменений в репозитории

После выполнения коммита система сохраняет снимок измененных файлов



# Практическая польза от использования Git

1. Сохранение состояний репозитория, файлов внутри и изменений
2. Возможность просмотра предыдущих версий и отката на них
3. Возможность работать в команде

# Сервисы для хостинга репозиториев



Основные команды для работы с git

# git config

Самое первое, что необходимо сделать – настроить конфиг, чтобы корректно отображалась информация о том, кто вносит изменения в Git.

```
git config --global user.name "Rakhman Imanov"  
git config --global user.email mail@mail.ru
```

# git init / git clone

git init сгенерирует подкаталог .git, содержащую структуру репозитория git. Происходит как бы инициализация, но контроля версий еще нет, поскольку не указаны файлы, изменения которых необходимо отслеживать

git clone копирует существующий репозиторий, все сохраненные на сервере данные.



# git add

git add предназначен для добавления файла под контроль версий, то есть для начала отслеживания изменений в нем

```
git add <файл>
```

# git status

Позволяет отслеживать состояния репозитория

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README
    modified:   CONTRIBUTING.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   CONTRIBUTING.md
```

# git commit

git commit фиксирует изменения в репозитории. Используя флаг -m, вы можете добавить комментарий, описывающий изменения. Хорошая практика — писать чёткие и информативные сообщения к коммитам, чтобы команда могла легко понять, что было сделано.

```
git commit -m "<commit message>"
```

# Работа с удаленным репозиторием

`git push`: отправка изменений на удалённый репозиторий.

`git fetch`: загрузка изменений без слияния

`git merge`: слияние изменений.

`git pull`: комбинация `fetch` и `merge`.

# git reset

git reset: возвращение к определённому коммиту

- --soft: изменения остаются
- --mixed: изменения в рабочей области
- --hard: сброс всех изменений

Восстановление коммита: git reflog

# Состояния git

- Рабочий каталог: редактируемые файлы.
- Индекс (staging directory): изменения, ожидающие коммита
- Репозиторий: актуальный код.
- Лог репозитория: хранилище всех зафиксированных состояний (git log)