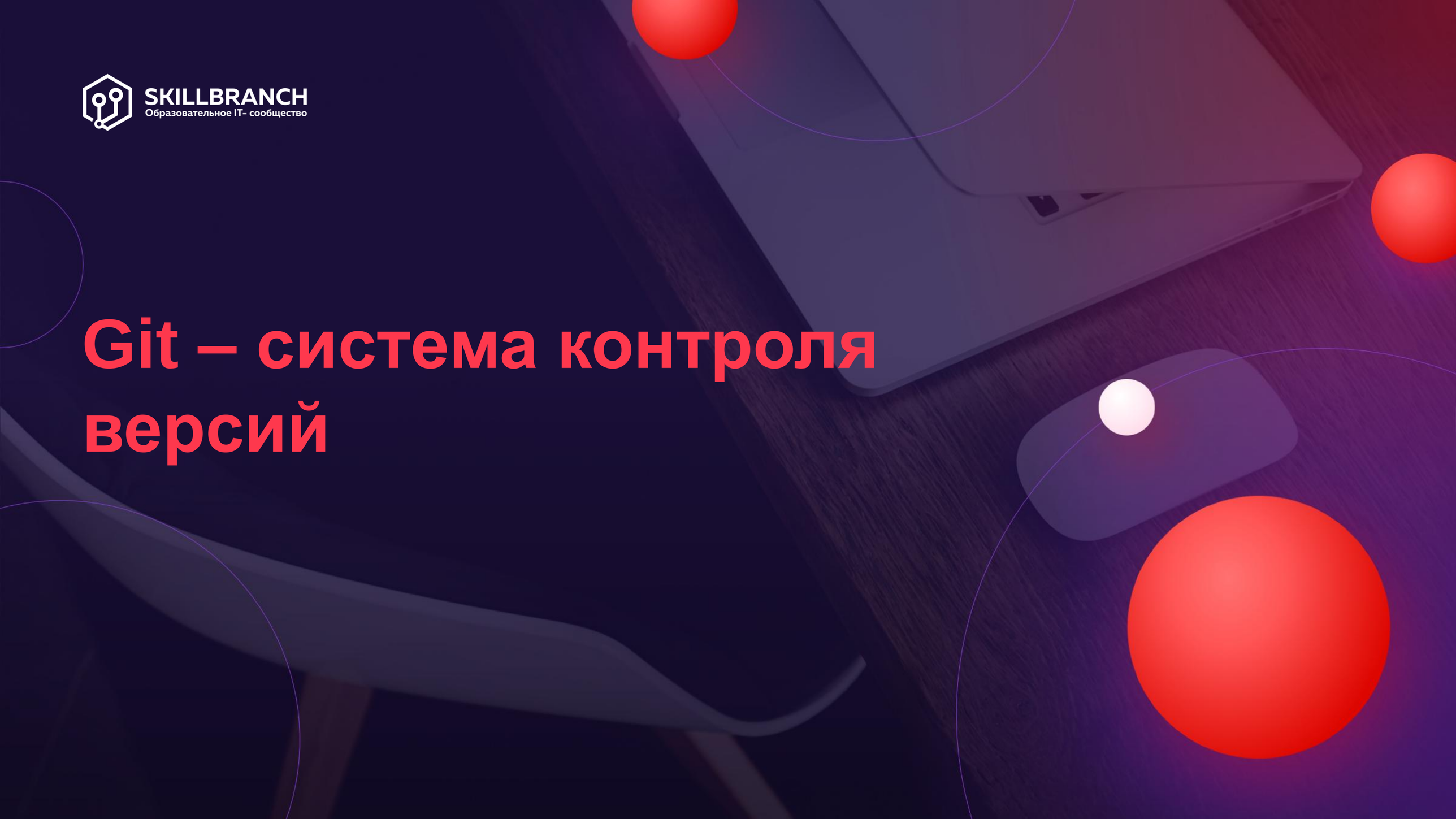
SKILLBRANCH
Образовательное IT- сообщество

Git – система контроля версий



Что такое Git

Git — распределённая система контроля версий, которая даёт возможность разработчикам отслеживать изменения в файлах и работать совместно с другими разработчиками

Бесплатный и open-source. Это значит, что его можно бесплатно скачать и вносить любые изменения в исходный код;

Небольшой и быстрый. Он выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных;

Резервное копирование. Git эффективен в хранении бэкапов, поэтому известно мало случаев, когда кто-то терял данные при использовании Git;

Простое ветвление. В других VSC создание веток— утомительная и трудоёмкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.

Git – local / remote

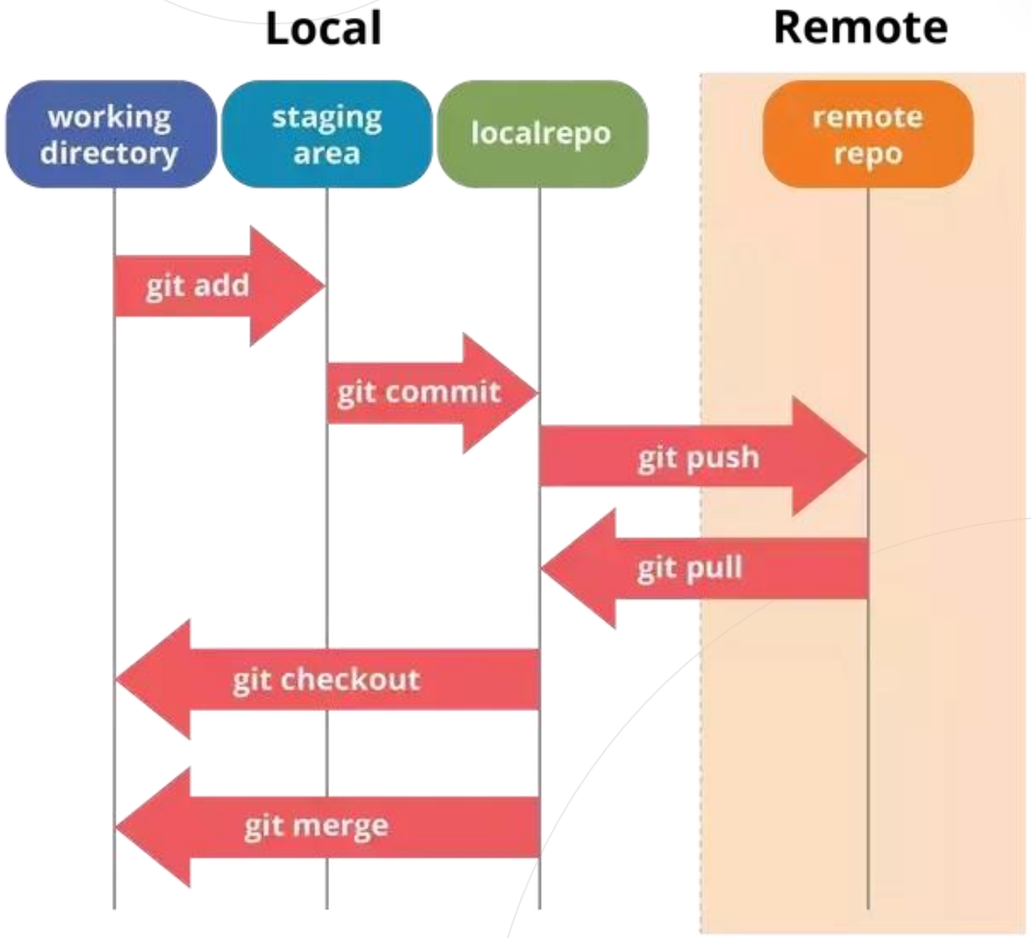
Репозиторий (repository) - место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети

Рабочая директория – это файлы в корне проекта, тот код с которым вы работаете.

Область подготовленных изменений (staging area) – область в которой находятся подготовленные файлы которые могут быть включены или не включены в коммит

Локальный репозиторий — она же директория .git. В ней хранятся коммиты и другие объекты.

Удаленный репозиторий (origin)–репозиторий который считается общим, в который вы можете передать свои коммиты из локального репозитория, что бы остальные разработчики могли их увидеть.



Git – с чего начать?

`git init` — создание нового локального репозитория

`git clone <url удаленного репозитория>` - клонирование уже имеющегося репозитория из удаленного хранилища

`git remote -v` – список имён-сокращений для всех уже указанных удалённых репозиториев с соответствующими им URL

`git remote set-url origin <url удаленного репозитория>` – установка нового url для origin вашего репозитория

Git commit

`git add <имя файла / имя директории>` - добавление файлов и директорий под версионный контроль

`git reset <имя файла / имя директории>` - удаление файла из под версионного контроль (находящегося в stage)

`git status` - просмотр состояния файлов в области подготовленных изменений (staging area)

Author	Commit	Message
Emma Paris	`c7cf7b7`	making a change in a branch
Emma Paris	`fccfa6f`	stationlocations created online with Bitbucket
Emma Paris	`5424bbf`	Initial commit

`git commit –m “комментарий”`- команда создающая слепок текущего состояния проекта и файлов в нем

`git commit --amend` – изменение информации о последнем коммите

`git reflog` - показывает список всех выполненных вами операций.

`git push <remote> <branch>` – запушить коммиты в удаленный репозиторий <remote> в ветку <master>

Каждый коммит может запоминать изменения сразу в нескольких файлах. Хорошая практика – в один коммит включать только те изменения, которые логически связаны между собой.

Git branch

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. Основная ветка по умолчанию имеет название master

git branch <имя ветки> — создание новой ветки <имя ветки>

git checkout <имя ветки> – переключение на ветку <имя ветки>

git checkout –b <имя новой ветки> создание новой ветки и переключение на нее

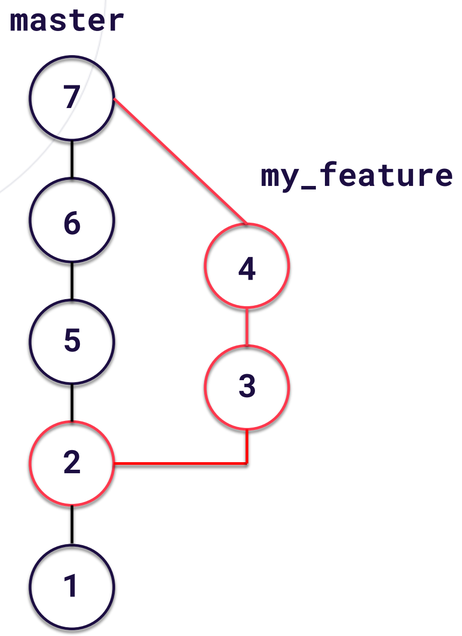
git merge <branch> — слияние текущей ветки с веткой <branch>

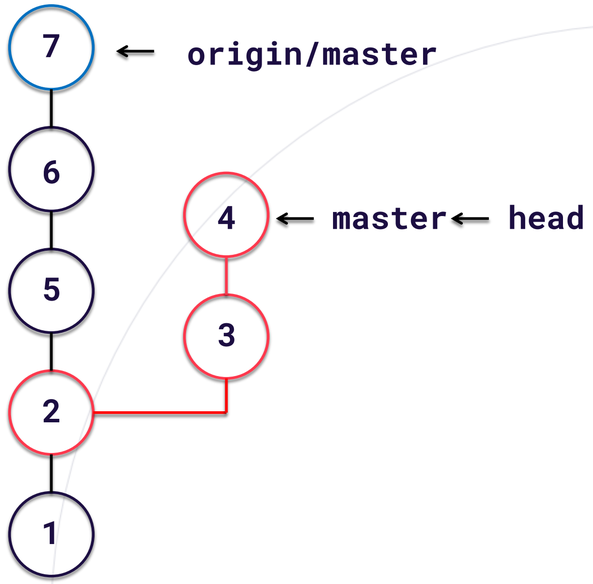
git abort — откатить изменения при слиянии в случае возникновения конфликтов

git pull — скачать коммиты и применить к рабочей директории

git fetch — скачать коммиты не применяя их к рабочей директории

git log --pretty=format:"%H [%cd]: %an - %s" --graph --date=format:%c – вывести в консоль отформатированную историю коммитов в виде графа





SKILLBRANCH
Образовательное IT- сообщество

Спасибо за внимание

пиши красивый код