# git

Система контроля версий

#### Что такое VCS?

Version Control System.

ПО для управления изменениями в информационных объектах (программы, приложения, документы и другое).

Много людей, много версий - это нужно организовать.

# Для чего?

- полная история изменений,
- отслеживание действий,
- ветвление и слияние.

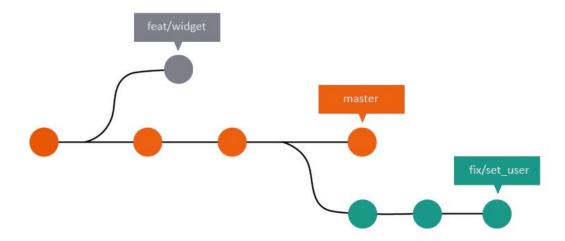
# Виды

- локальная,
- централизованная,
- распределенная.

# git. История

Линус Торвальдс, апрель 2005 года.

# git



- D STC
- > test
- types
  - - .editorconfig
    - .eslintignore

    - .flowconfig
    - gitignore
    - BACKERS.md

v2.6.0-beta.2 | build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom

origin/dynamic-directive-arguments feat: dynamic args for customers

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync origin/slot-optimization perf: improve scoped slots charge

feat: dynamic directive arguments for v-bind and v-on-

refactor: extend dom-props update skip to more

fix: fix checkbox event edge case in First

test: fix tests in IE/Edge

refactor: simplify times are a street

LICENSE

# git – это не Github





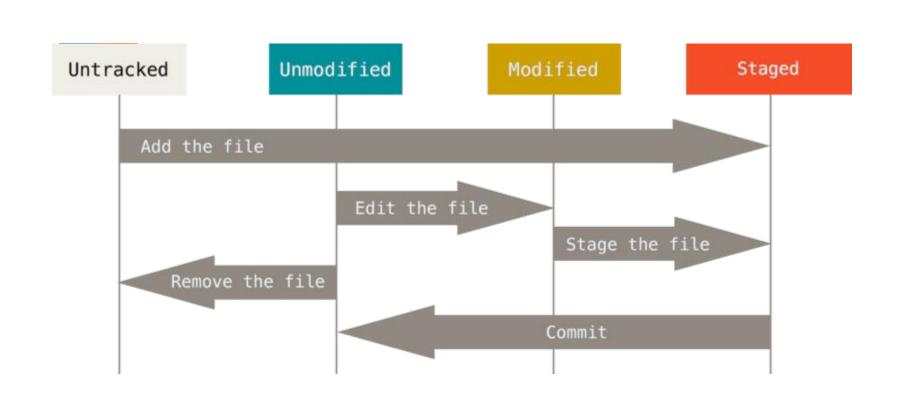


# git. Три фундаментальные команды

- pull (обновление рабочей копии),
- push (фиксация изменений),
- merge (слияние ветвей).

# git. Три базовых состояния файлов

- вносим изменения в файл проекта -> modified
- git add -> staged
- git commit -> committed



# git. Репозитории

- локальный (add, commit, merge),
- удаленный (pull, push).

# Начало работы с git



- 1) установка git,
- 2) настройка конфига,
- 3) создание репозитория

ИЛИ

клонирование репозитория.

#### **Установка**

https://git-scm.com/book/ru/v2/Введение-Установка-Git

# Настройка

git config --global user.name "Big Lump"

git config --global user.email biglump9000@example.com

## Виды конфигов

git config --list --show-origin

- **СИСТЕМНЫЙ** [path]/etc/gitconfig,
- глобальный ~/.gitconfig,
- локальный .git/config в корне проекта.

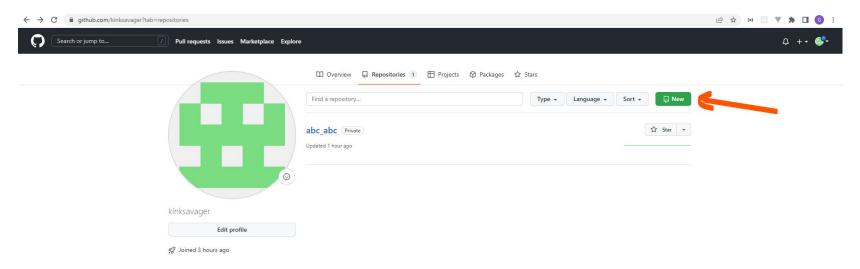
# Создание репозитория

1) git init

после этого папка, в которой вы находитесь, становится вашим локальным репозиторием

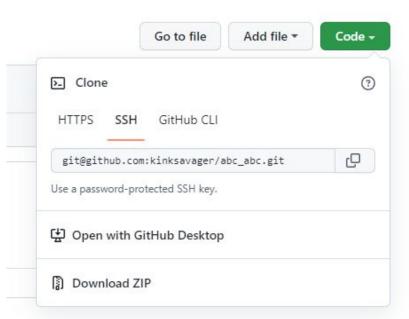
#### Создание репозитория

2) добавляем репозиторий на Github ЛИБО используем gh repo create и следуем интерактивным подсказкам



#### Создание репозитория

3) добавление ссылки на удаленный репозиторий git remote add origin git@github.com:kinksavager/abc\_abc.git



#### SSH соединение

ssh-keygen -t ed25519 -C "your\_email@example.com"

Сгенерированная пара ключей хранится в ~/.ssh

Публичный ключ необходимо добавить в раздел SSH and GPG keys в настройках аккаунта на Github.

### Клонирование репозитория

git clone git@github.com:kinksavager/abc\_abc.git

После того, как мы настроили защищенный канал связи с удаленным репозиторием, мы можем свободно стягивать или отправлять изменения.

# .gitignore

touch .gitignore B KOPHe Προεκτα.

```
# See https://help.github.com/articles/ignoring-files for more about ignoring
files.
# Ignore bundler config.
/.bundle
# Ignore all logfiles and tempfiles.
/log/*
/tmp/*
!/log/.keep
!/tmp/.keep
*.mmdb
# Ignore Byebug command history file.
.byebug_history
.DS_Store
*.log
# Ignore application configuration
node_modules
master.key
*.rdb
# Ignore env files
.env*
public/uploads
public/packs*
public/assets
```

#### Домашнее задание

- 1) создать приватный репозиторий на Github,
- 2) настроить доступ к нему через SSH-ключ,
- 3) сделать свой первый коммит и запушить его.