

Урок 3

Введение в Git

2024

 MobyDev



Что такое Git

Git — это распределенная система управления версиями (VCS), предназначенная для отслеживания изменений в файлах и координации работы над этими файлами среди нескольких людей.

Git изначально был создан Линусом Торвальдсом в 2005 году для управления разработкой ядра Linux, и с тех пор стал основным инструментом для управления исходным кодом в большинстве программных проектов.



Необходимость в разработке

Основные функции и преимущества Git:

- **Управление версиями:** Git позволяет отслеживать историю изменений в проектах.
- **Распределенность:** В Git каждый разработчик имеет полную копию всего репозитория, включая всю историю изменений. Это позволяет работать оффлайн и уменьшает риск потери данных.
- **Ветвление и слияние:** Git предоставляет мощные средства для работы с ветвями (branches). Ветвление позволяет работать над новыми функциями и исправлениями.
- **Совместная работа:** Каждый разработчик может работать в своей ветви и потом объединить изменения в общий репозиторий, решая конфликты по мере необходимости.
- **История и аудит:** Полная история изменений позволяет отслеживать, кто, когда и какие изменения вносил, что упрощает отладку и аудит кода.

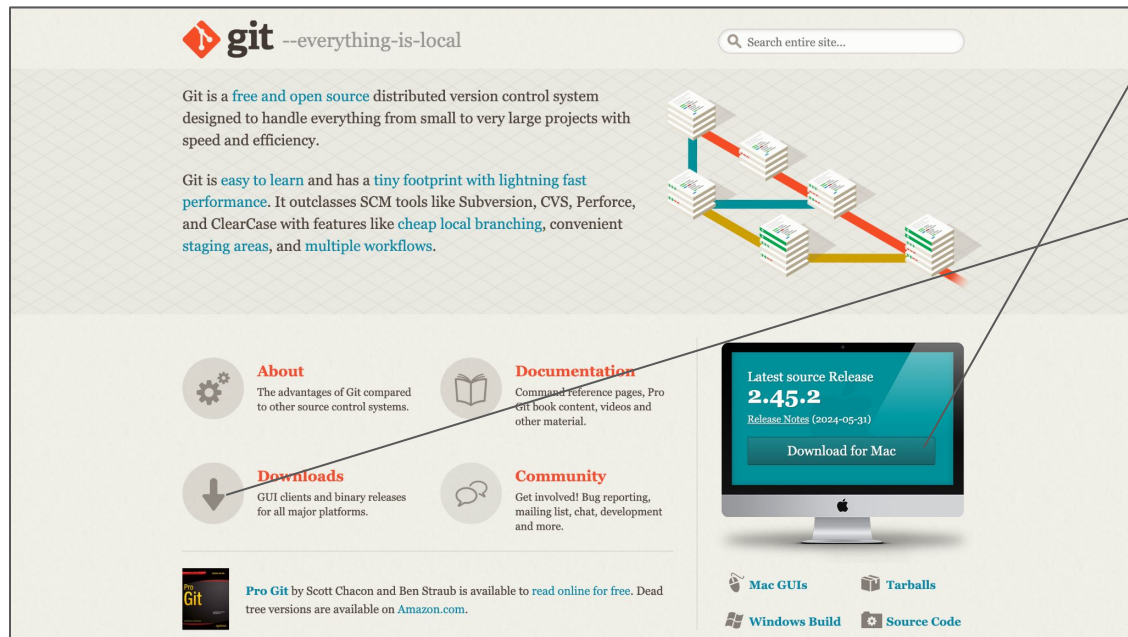


Загрузка/Установка Git

Загрузка Git <https://git-scm.com/>

Кнопка загрузки

Выбор операционной системы



The screenshot shows the Git website homepage. At the top left is the Git logo with the tagline "--everything-is-local". To the right is a search bar. Below the logo, there are two paragraphs of text describing Git as a free and open source distributed version control system. In the center, there is a diagram showing a branching model with stacks of code blocks connected by arrows. Below this, there are four circular icons with labels: "About", "Documentation", "Downloads", and "Community". The "Downloads" icon is highlighted with a red arrow. At the bottom left, there is a section for "Pro Git" by Scott Chacon and Ben Straub. At the bottom right, there is a large monitor icon displaying the "Latest source Release 2.45.2" and a "Download for Mac" button. Below the monitor, there are links for "Mac GUIs", "Tarballs", "Windows Build", and "Source Code".

Downloads



macOS



Windows



Linux/Unix

Older releases are available and the [Git source repository](#) is on GitHub.

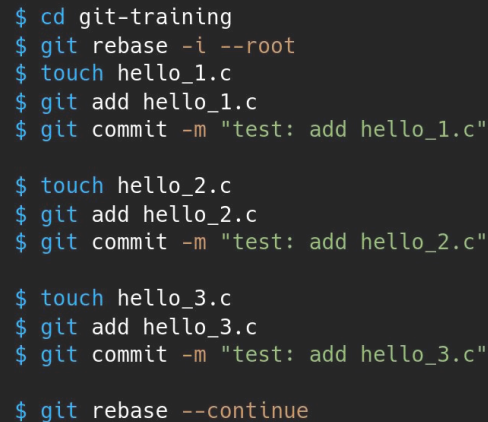


Работа с терминалом

Терминал — это текстовый интерфейс, через который пользователь взаимодействует с операционной системой. Он предоставляет доступ к командной строке (CLI), позволяя вводить команды для выполнения различных задач.

Основные преимущества работы с терминалом:

- Мощность и гибкость;
- Автоматизация задач;
- Управление системой;
- Удаленный доступ;
- Разработка программного обеспечения.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It displays a series of Git commands and their outputs, demonstrating a workflow to create and commit files.

```
$ cd git-training
$ git rebase -i --root
$ touch hello_1.c
$ git add hello_1.c
$ git commit -m "test: add hello_1.c"

$ touch hello_2.c
$ git add hello_2.c
$ git commit -m "test: add hello_2.c"

$ touch hello_3.c
$ git add hello_3.c
$ git commit -m "test: add hello_3.c"

$ git rebase --continue
```



Основные команды Git

Git предоставляет широкий набор команд для управления версиями, отслеживания изменений и совместной работы над проектами.

- **git init** — Инициализация нового репозитория.
- **git status** — Проверка статуса репозитория.
- **git add <file>** — Добавление изменений в область подготовки (staging area).
- **git commit -m "comment"** — Фиксация изменений с сообщением.
- **git remote add origin <URL>** — Добавление удалённого репозитория.
- **git diff** — Просмотр изменений перед коммитом.
- **git clone <URL>** — Клонирование существующего репозитория.
- **git pull** — Получение и интеграция изменений из удалённого репозитория.
- **git push** — Отправка изменений в удаленный репозиторий.



Глобальная конфигурация git

Конфигурация Git на глобальном уровне позволяет задать настройки, которые будут применяться ко всем репозиториям пользователя. Это позволяет задать общие настройки, чтобы не нужно было настраивать их для каждого репозитория отдельно.

Настройка:

- Откройте **терминал/командную строку**;
- Введите команду **git config --global user.name "..."** для установки имени пользователя
- Для установки email пользователя введите команду **git config --global user.email "..."**

```
git config --global user.name "Ваше Имя"
```

```
git config --global user.email "ваш.email@example.com"
```



Урок 3

Введение в Git

2024

 MobyDev

