

Git beginners guide

✎ Предварительные приготовления

Предисловие: к большинству используемых команд приложены скриншоты результата выполнения. Некоторые данные на скриншотах могут отличаться в зависимости от ОС и версий программного обеспечения каждого пользователя.

✎ Все команды вводятся в терминал/консоль

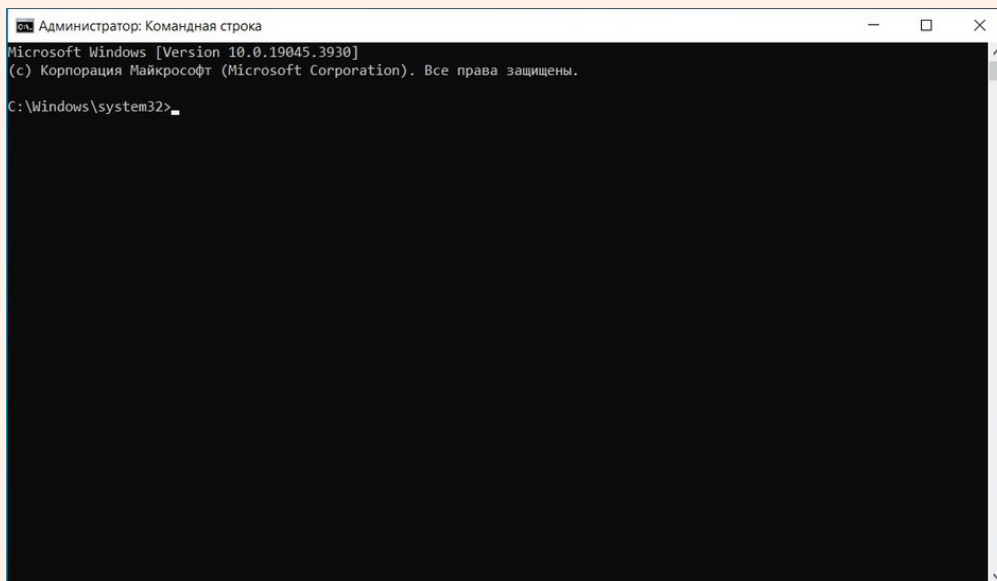
Понятия "терминал" и "консоль" смежны между собой и по сути представляют собой текстовую оболочку для взаимодействия с системой. Исторически сложилось, что на **Windows** эта оболочка называется *консолью*, а на **Mac/Linux** *терминалом*.

❓ Как открыть консоль на Windows?

Нажимаем сочетание клавиш `WIN + R`

Вводим `cmd`

Открывается окно на подобии этого:



P.S: для любителей "по красивее" можно установить [Windows Terminal](#)

❓ Как открыть консоль на Mac?

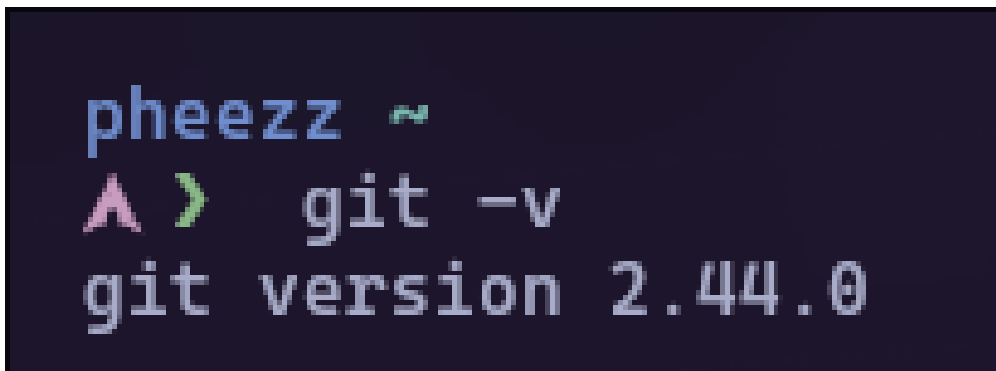
Нажимаем сочетание клавиш `CMD + T`

Вуаля, терминал открыт:



- Проверяем установлен ли `git`

```
git -v
```



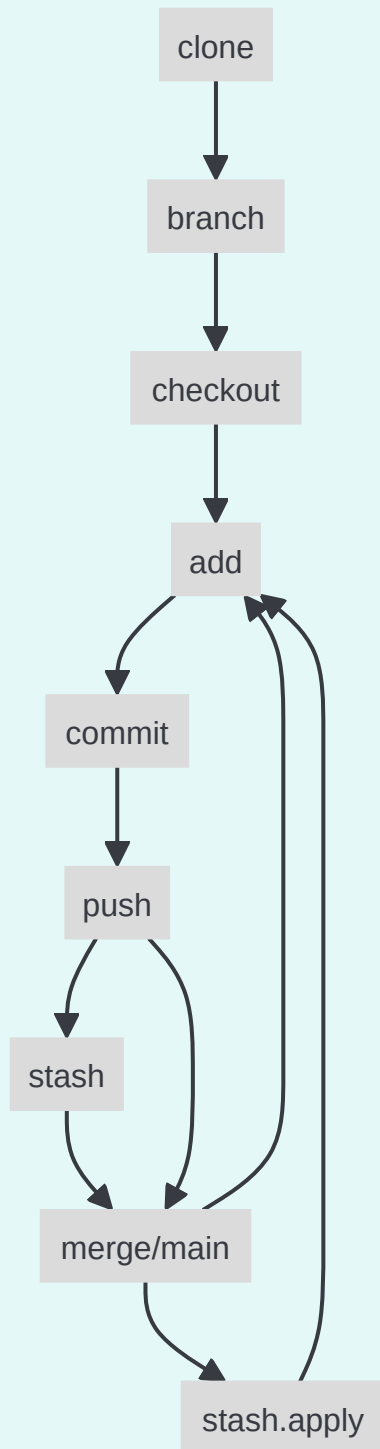
- Устанавливаем свои имя и email (они будут использоваться для идентификации того, кто внес конкретные изменения в проект)

```
git config --global user.name "Neon Lover"  
git config --global user.email "neon.lover@gmail.com"
```

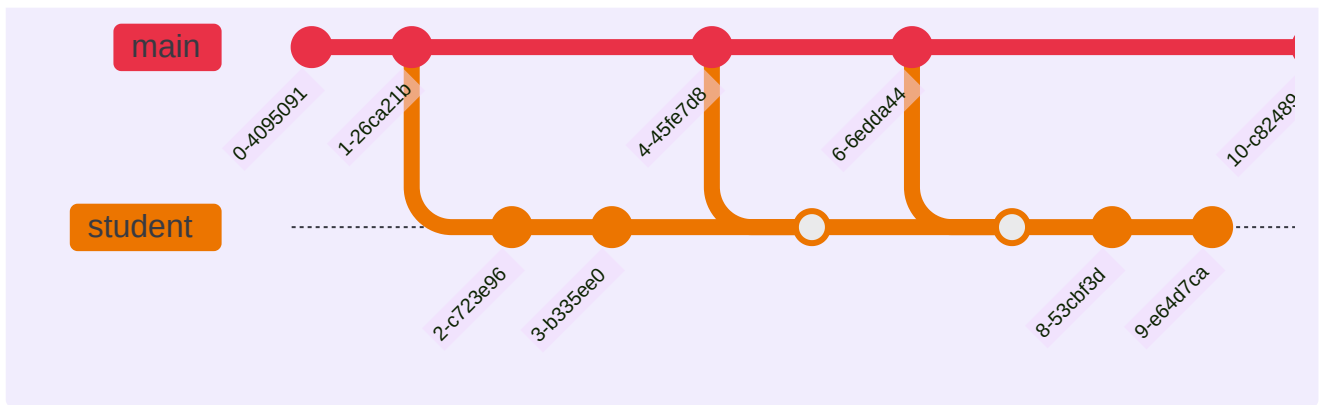
🔗 Проверить текущие имя и email можно с помощью команды:

```
git config --list
```

📋 Порядок выполнения Git команд



☰ Пример Git ветвления для параллельной разработки



git clone

Команда `git clone`: Клонировать удаленный репозиторий на ваш компьютер. Это создает локальную копию удаленного репозитория.

```
git clone url.git
```

git branch

Команда `git branch`: Создает новую ветку с указанным названием. Это позволяет вам работать над различными частями проекта параллельно, не вмешиваясь в основную ветку.

```
git branch branch_name
```

git checkout

Команда `git checkout`: Переключает вас на другую ветку в вашем репозитории. Вы можете переключаться между существующими ветками или создать новую ветку с этой командой.

| Переключение на созданную ветку

```
git checkout branch_name
```

| Создание новой ветки и переключение на нее

```
git checkout -b new_branch_name
```

Info

Команда `git add`:

Добавляет изменения в индекс, подготавливая их к фиксации (коммиту) в локальном репозитории.

| Добавление всех файлов в индекс

```
git add .
```

| Добавление конкретного файла

```
git add filename
```

git pull

Команда `git pull`: Получает изменения из удаленного репозитория и интегрирует их в вашу локальную ветку. Это позволяет вам обновлять свою локальную копию репозитория с изменениями, внесенными другими участниками проекта.

```
git pull
```

git commit

Команда `git commit`: Фиксирует изменения, добавленные в индекс, в локальном репозитории. Комментарий к коммиту позволяет объяснить, какие изменения были внесены.

```
git commit -m "commit message"
```

git push

Команда `git push`: Отправляет ваши фиксации изменений (коммиты) в удаленный репозиторий. Это делает вашу работу доступной для других участников проекта.

Info

Команда `git merge`:

Объединяет изменения из одной ветки в другую. Это позволяет вам интегрировать изменения из одной ветки в другую, обычно используется для слияния веток feature в основную ветку (обычно master или main).