

**Командная**

**строка**

# Что такое командная строка Bash

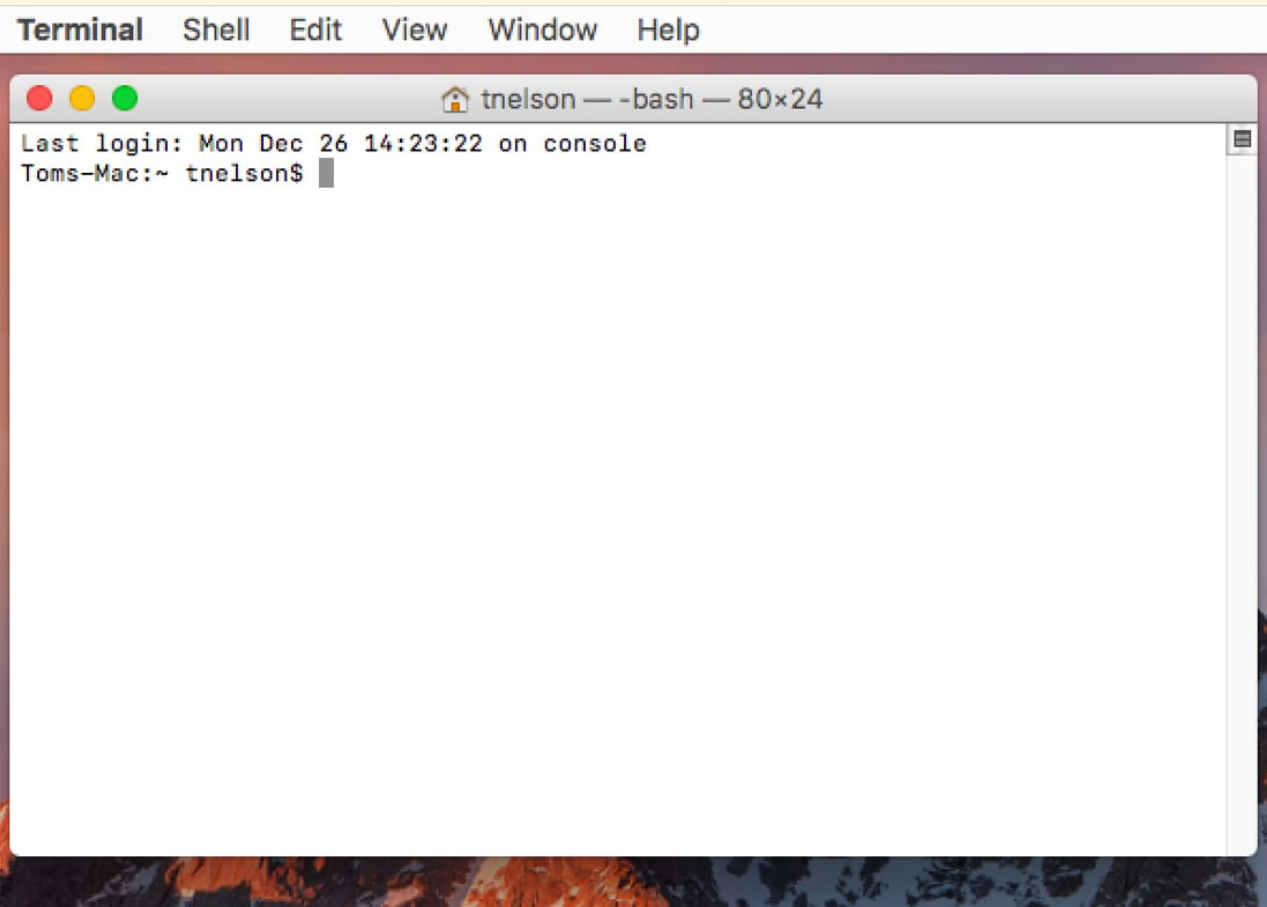
Командная строка — это текстовый интерфейс для работы человека с операционной системой, путём отправки компьютеру команд.

**Bash** — это компьютерный язык; он задаёт слова и правила для составления команд для операционной системы (ОС).

# Командная строка:

- имя компьютера
- текущий каталог (~)
- имя пользователя
- приглашение к вводу команды (■)

■ ~ означает домашний каталог



# Навигация по файловой системе

Одна из самых важных вещей, которые вы можете сделать с помощью Bash — это навигация по файловой системе.

- `pwd` (print working directory) — отображает *текущий каталог*, в котором вы находитесь.
- `ls` (list) — показывает *содержимое текущего каталога*.
- `cd` (change directory) — используется для перехода в другой каталог, например: `cd /home/user/documents`.

# Манипулирование файлами и каталогами

- **cp** (copy) — может быть использована для копирования файла, например: `cp file1.txt file2.txt`.
- **mv** (move) — может использоваться для перемещения или переименования файла, например:  
`mv file1.txt documents/file1.txt`.
- **rm** (remove) — удаляет файл, например: `rm file1.txt`.
- **rmdir** (remove directory) — удаляет каталог, например:  
`rm file1.txt`.

# Создание файлов и каталогов

- `mkdir` (make directory) — *создаёт директорию*, например:
  - ○ `mkdir dir1` — 1 директория.
  - ○ `mkdir dir1 dir2 dir3` — несколько директорий.
- `touch` — *может использоваться для создания файла*:
  - ○ `touch file1` — 1 файл.
  - ○ `touch file1 file2 file3` — несколько файлов.

# Особенность названий команд в Bash

`cp` ≠ Cp ≠ CP ≠ cP

`mkdir` ≠ mKdir

Регистр важен как для команд, так и для названий файлов!

(В файловых системах семейства Unix и языках программирования семейства Си)

# Просмотр и редактирование файлов

Вы также можете просматривать и редактировать файлы с помощью Bash.

Команда `cat` (concatenate) отображает содержимое файла, например: `cat file1.txt`.

Команды `nano` или `vi` - это текстовые редакторы для редактирования файлов, например: `nano file1.txt`.



vim

# Режимы в vi

## Командный режим

- перемещение курсора
- удаление текста
- ввод команд

## Режим вставки

- ввод и удаление текста

Для переключения между режимами используйте:  
клавишу **Esc**, чтобы перейти в командный режим;  
**i** или **a**, чтобы перейти в режим вставки.

# Навигация по тексту

Vi предоставляет несколько команд для перемещения курсора и навигации по тексту

**h** перемещает курсор влево, **j** - вниз, **k** - вверх, а **l** - вправо.

**w** перемещает курсор к началу следующего слова, а **b** - к началу предыдущего слова

**0** (ноль) перемещается в начало строки, а **\$** - в конец строки.

# Манипулирование текстом

Vi также предоставляет команды для работы с текстом

**x** — удаляет символ под курсором.

**dd** — удаляет текущую строку

**u** — отменяет последнее изменение, а **Ctrl + r** повторяет последнее изменение.

Чтобы скопировать текст, используйте **y** для выделения (копирования) и **p** для вставки.

# Сохранение и выход из программы

Для *сохранения без выхода* из программы используйте команду `:w`.

Чтобы *выйти без сохранения*, используйте команду `:q!`.

Чтобы *сохранить изменения и выйти* из Vi, совместите команды `w` и `q`, то есть `:wq`.

# Git

# Git

git — это система контроля версий, которая отслеживает и фиксирует изменения в проекте или файле. Git не делает копии (дубликаты) файлов, она только запоминает изменения.

- вернуться к предыдущим версиям файла.
- вести четкую историю изменений.
- сотрудничать с другими людьми над одним и тем же проектом.

# Службы git-хостинга

GitHub, GitLab и Bitbucket, предоставляют удаленный репозиторий, в котором вы можете хранить код и получать к нему доступ из любого места.

- Совместная работа.
- Резервное копирование и восстановление.
- Безопасность (аутентификация, авторизация).
- Видимость проекта для людей всего мира.
- Сообщество разработчиков и пользователей.



# Git команды 1

- `git init` — инициализация репозитория (проекта) на вашем локальном компьютере.
- `git clone` — клонирование удалённого репозитория на ваш локальный компьютер.
- `git add` — добавление изменений в промежуточную область перед их фиксацией в репозитории.
- `git commit` — сохранение изменений в локальном репозитории.
- `git push` — отправка изменений из локального репозитория в удаленный репозиторий.

# Git команды 2

- `git pull` — извлечение изменений из удаленного репозитория в локальный репозиторий.
- `git merge` — объединение изменений из нескольких веток в одну ветку.
- `git status` — проверки состояния репозитория и просмотра того, какие файлы были изменены.
- `git log` — просмотр истории фиксаций репозитория.

# Удачи!

**Best of luck!**