

Git. Базовый курс

# Урок 2

## Работа с командной строкой

[Командная строка](#)

[Запуск командной строки](#)

[Windows](#)

[MacOS](#)

[Ubuntu](#)

[Текстовый редактор Sublime Text](#)

[Работа с файловой системой в консоли](#)

[Пути к файлам](#)

[Особенности навигации в Windows](#)

[Используемые источники](#)

# Командная строка

Если вы хорошо знакомы с командной строкой, постоянно используете ее в работе, этот урок можно смело пропустить.

Командная строка — один из самых первых способов работы с компьютером, вы вводите команду и в ответ получаете результат:

```
$ cal
      А в г у с т 2019
в с п н в т с р ч т п т с б
      1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

Большинство современных пользователей компьютеров знакомы только с графическим интерфейсом. Интерфейсом командной строки пользуются в основном администраторы и программисты.

К достоинствам командной строки можно отнести:

- Скорость набора и выполнения команд, вы не отвлекаетесь на операции мыши.
- Возможность быстрой автоматизации действий за счет комбинации команд.
- Работа с компьютером на более низком уровне, использование возможностей, для которых не реализован графический интерфейс.

Конечно, у командной строки есть недостатки, иначе бы вместо графического интерфейса все использовали терминал:

- Командная строка не так наглядна, как графический интерфейс.
- Для работы с ней требуется запомнить большой набор команд, фактически, выучить мини-язык программирования.
- В связи с этим при работе в командной строке очень легко ошибиться. В большинстве случаев опечатка или ошибка приводит к тому, что компьютер отказывается выполнять бессмысленные с его точки зрения действия. Однако, возможны и фатальные ошибки, которые могут приводить к деструктивным и необратимым действиям.

Тем не менее, для программистов и системных администраторов умение работать в командной строке — обязательное требование.

Командная строка не так страшна и довольно удобна, особенно в UNIX-подобных операционных системах, где она доведена практически до совершенства. Создание графического интерфейса — довольно трудоемкая задача, которая оправдывает себя для массовых действий. Однако для единичных операций или эксперимента — это непозволительная роскошь.

Навыки и знания в программировании довольно быстро устаревают. Навыки работы в командной строке составляют приятное исключение, они остаются практически неизменными уже 50 лет.

Для многих операций, особенно массовых, в командной строке можно добиться скорости, на порядок превышающей аналогичные действия в графической оболочке.

Кроме того, многие операции по развертыванию, деплою и компиляции приложений осуществляются консольными командами. Поэтому в группе разработки обязательно должно быть минимум два человека, которые владеют командной строкой и способны автоматизировать свои действия.

Именно поэтому данный курс практически полностью пройдет в командной строке. Возможно, вы именно тот человек, которому предстоит настроить деплой приложения на сервер или исправить ошибку слияния в системе контроля версий, разрешить ситуацию, перед которой бессильна программа с графическим интерфейсом.

## Запуск командной строки

Итак, давайте попробуем использовать командную строку в вашей операционной системе. Устанавливать ее не требуется, все операционные системы имеют встроенные программы для работы в командной строке.

### Windows

В Windows запустить командную строку можно несколькими способами: либо выбрав в меню, в стандартных настройках пункт «Командная строка», либо в строке поиска, набрав команду `cmd`.

Первое, что вас ждет в командной строке — приглашение к вводу, оно появляется всякий раз, когда консоль готова к вводу команд.

Если набрать какую-нибудь команду, например, `DIR`, будет получен ответ, в данном случае: список файлов текущего каталога. После выполнения и вывода снова можно видеть приглашение, ожидающее следующей команды.

```
> DIR
```

Символ `>` тут и далее набирать не нужно, это условное обозначение приглашения командной строки в Windows.

Командная строка в Windows довольно сильно отличается от UNIX-консоли, например, в UNIX для вывода списка файлов вместо `dir` используется команда `ls`. Для вывода содержимого файла в Windows используется команда `TYPE`, в UNIX — команда `cat` и так практически с каждой операцией, почти все команды в Windows и UNIX отличаются.

Если вы работаете с UNIX-технологиями, например, разворачиваете веб-приложения на UNIX-серверах, осваиваете базы данных, защиту информации или занимаетесь системным администрированием, лучше сразу ориентироваться на более богатую возможностями командную строку UNIX.

Поэтому в Windows часто воссоздают UNIX-консоль. Один из таких способов — воспользоваться подсистемой UNIX. Активация Linux-подсистемы подробно описывается в официальной документации по ссылке <https://docs.microsoft.com/ru-ru/windows/wsl/install-win10>.

## MacOS

В MacOS-и для запуска консоли можно выбрать Launchpad, папку «Другие» и выбрать программу «Терминал». Другой способ — воспользоваться Spotlight: в нем можно набрать ключевое слово «Терминал».

Для получения списка файлов текущего каталога можно воспользоваться командой **ls**:

```
$ ls
```

Символ **\$** тут и далее, набирать не нужно, это условное обозначение приглашения командной строки в UNIX-подобных операционных системах.

MacOS — это коммерческая UNIX-среда, здесь будут действовать все привычные UNIX-команды. Пользователи MacOS часто прибегают к установке программы iTerm, которая является альтернативой штатному терминалу.

Многие команды могут принимать дополнительные параметры, например, вывести дополнительные атрибуты каждого файла и каталога можно, добавив параметр **-l** (параметры предваряются знаком минус):

```
$ ls -l
```

Если мы добавим параметр **-a**, будут выведены в том числе скрытые файлы, которые в UNIX-подобных операционных системах начинаются с точки:

```
$ ls -l -a
```

Часто, чтобы сделать команду еще более короткой, параметры можно объединять:

```
$ ls -la
```

## Ubuntu

Чтобы добраться до командной строки в linux-дистрибутиве Ubuntu, можно перейти в меню: **Стандартные->Терминал**. Также можно вызывать окно запуска приложений при помощи **Alt+F2** и ввести в нем название программы **gnome-terminal**. В случае, если вы используете KDE, программа будет называться **konsole**.

Можем воспользоваться командой **ls** для получения списка файлов текущего каталога

```
$ ls -la
```

## Текстовый редактор Sublime Text

При работе в консоли мы будем добавлять и изменять файлы, а чтобы за изменениями было проще следить, будем использовать текстовый редактор с возможностью навигации по файловой системе.

Для наших целей подойдет абсолютно любой редактор, на протяжении курса мы будем использовать Sublime Text.

Редактор платный, однако допускается его неограниченное использование без лицензии. В этом случае время от времени будет появляться диалоговое окно, призывающее купить лицензию. В остальном функционал редактора не ограничен. Впрочем, вы можете использовать любой другой текстовый редактор.

## Работа с файловой системой в консоли

При работе в командной строке важно сразу освоиться с навигацией по файловой системе, так как мы не сможем воспользоваться графическим интерфейсом, чтобы перейти в ту или иную папку.

Сразу после открытия терминала вы оказываетесь в домашнем каталоге. Узнать точный путь до текущей папки можно при помощи команды **pwd**:

Мы находимся в домашнем каталоге, узнать точный путь к нему можно при помощи команды **pwd**:

```
$ pwd
```

Для перемещения между каталогами используется команда **cd**:

```
$ cd ~/www/geekbrains/git
```

Создать файл можно при помощи команды **touch**:

```
$ touch hello.txt
```

Получить список файлов можно при помощи команды **ls -la** (в Windows это будет команда **dir**):

```
$ ls -la
```

Для создания каталога можно воспользоваться командой **mkdir**:

```
$ mkdir test
```

Вывести фразу в консоль можно при помощи команды **echo**:

```
$ echo 'Hello, world!'
```

Можно перенаправить такой вывод в файл, для этого можно воспользоваться символом больше **>**, после которого указывается имя файла:

```
$ echo 'Hello, world!' > hello.txt
```

Содержимое файла мы можем прочитать при помощи команды **cat**:

```
$ cat hello.txt
```

Следующая команда перезаписывает содержимое файла фразой 'Hello, Git!':

```
$ echo 'Hello, Git!' > hello.txt  
$ cat hello.txt
```

Чтобы дозаписать какую-то информацию в конец файла, не перезаписывая его содержимое, можно воспользоваться последовательностью два знака больше **>>**:

```
$ echo 'Hello, world!' >> hello.txt
```

В результате в файле `hello.txt` к фразе `'Hello, Git!'` в конец добавится фраза `'Hello, world!'`.

Удалить файл `hello.txt` можно при помощи команды **unlink**:

```
$ unlink hello.txt  
$ ls -la
```

Можно использовать клавишу «Вверх» для навигации по введенным ранее командам, это позволяет избежать их повторного ввода.

## Пути к файлам

В командах часто требуется указывать путь к файлу, поэтому остановимся на правилах формирования пути к ним отдельно.

Каждый файл имеет имя и расположен в каком-либо каталоге, при этом каталоги можно вкладывать друг в друга. Полное имя файла определяется именами вложенных каталогов от корня диска. Такая последовательность называется путем к файлу. В различных операционных системах используются разные разделители между каталогами и файлами.

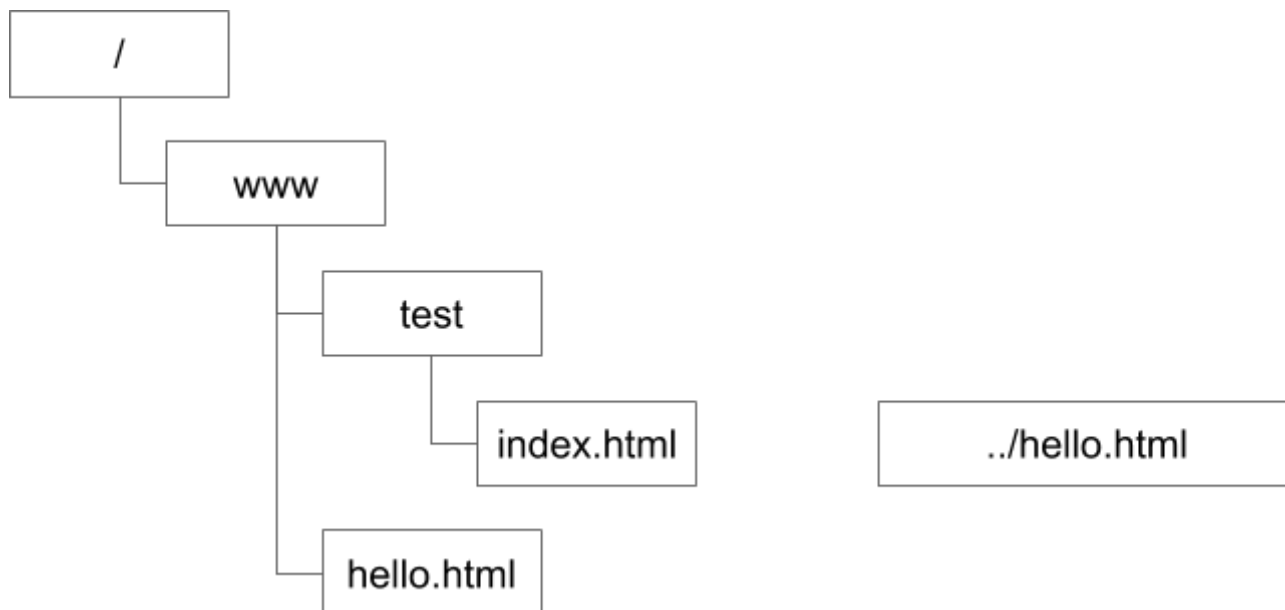
В UNIX-подобных операционных системах в качестве разделителя используется прямой слеш — `/`, в Windows — обратный `\`.

	абсолютный	относительный
UNIX	<code>/var/home/www/htdocs/text.txt</code>	<code>htdocs/text.txt</code>
Windows	<code>C:\www\htdocs\text.txt</code>	<code>htdocs\text.txt</code>

Слева на рисунке приводятся пути к файлу от корня диска — такие пути называются абсолютными. Справа приведен пример пути относительно каталога `www`, такие пути называются относительными.

В операционной системе Windows разделы обозначаются символами английского алфавита. В начале абсолютного пути указывается символ диска. В UNIX-подобных операционных системах любой путь начинается с корневого раздела или, как еще говорят, `root`-раздела, который

обозначается слешем `/`. В UNIX физические разделы, как правило, монтируются к каталогам файловой системы. Таким образом, если в Windows может быть несколько корневых точек, обозначенных буквами, в UNIX-подобных операционных системах такая точка всегда одна.



Относительный путь выстраивается относительно текущего каталога. Следует отметить, что относительный путь не предваряет ни буква раздела, ни символ корневого раздела `/`, он начинается либо с имени каталога, либо с имени файла.

Чтобы из каталога `test` добраться до файла `hello.html`, необходимо подняться на один уровень выше. Для формирования таких путей используется последовательность из двух точек `..`, обозначающая родительский каталог. Помимо родительского каталога, обозначаемого двумя точками (`..`), большинство файловых систем поддерживает текущий каталог, который обозначается одной точкой.

## Особенности навигации в Windows

В Windows для переключения между дисками можно указать имя диска и двоеточие. Перемещаться между папками можно при помощи команды `cd`. Вывод команд, как и в UNIX-подобных операционных системах, можно перенаправлять при помощи знака больше `>`:

```
$ DIR > text.txt
```

Однако для просмотра содержимого файла вместо команды `cat` используется команда `TYPE`:

```
$ TYPE text.txt
```



Если мы сейчас переключимся в редактор и посмотрим содержимое файла `text.txt`, убедимся, что он содержит вывод команды **`dir`**.

## Используемые источники

Для подготовки методического пособия мы использовали эти ресурсы:

- Командная строка Linux. Уильям Шоттс
- Сценарии командной оболочки Linux, OS X и UNIX. Дейв Тейлор, Брендон Перри
- Командная строка Linux и сценарии. Р. Блум, К. Бреснахэн
- Официальная документация Git <https://git-scm.com/book/ru/v2>.