

1.2 Запускаем браузер с помощью Selenium WebDriver

Шаг 1

Содержание урока

- [Установка Python3 в Windows \(/lesson/25969/step/2\)](/lesson/25969/step/2)
- [Установка Python3 в Ubuntu \(/lesson/25969/step/3\)](/lesson/25969/step/3)
- [Установка Python3 в macOS \(/lesson/25969/step/4\)](/lesson/25969/step/4)
- [Задание: интерпретатор Python \(/lesson/25969/step/5\)](/lesson/25969/step/5)
- [Установка Selenium для Python \(/lesson/25969/step/6\)](/lesson/25969/step/6)
- [Установка драйвера для браузера \(/lesson/25969/step/7\)](/lesson/25969/step/7)
- [Установка драйвера для браузера: Windows \(/lesson/25969/step/8\)](/lesson/25969/step/8)
- [Установка драйвера для браузера: Linux \(/lesson/25969/step/9\)](/lesson/25969/step/9)
- [Установка драйвера для браузера: macOS \(/lesson/25969/step/10\)](/lesson/25969/step/10)
- [Запуск браузера и первый скрипт \(/lesson/25969/step/11\)](/lesson/25969/step/11)
- [Задание \(/lesson/25969/step/12\)](/lesson/25969/step/12)

Шаг 2

Установка Python3 в Windows

Если вы используете другую ОС, то смело пропускайте этот шаг и переходите к следующему.

Внимание! В курсе вы будете работать в командной строке (консоли). Если вы никогда ничего подобного не делали, то не нужно бояться — в этом нет ничего страшного. Можете почитать про некоторые полезные команды по этой ссылке — [Работа с командной строкой \(https://cmd.readthedocs.io/cmd.html\)](https://cmd.readthedocs.io/cmd.html).

Для начала вам нужно скачать установочный файл для Python3. Для этого на официальной странице (<https://www.python.org/downloads/windows/>) Python выберите стабильную версию для Windows, и скачайте файл для вашей системы (64-разрядная или 32-разрядная).

Если Python у вас уже есть в системе, убедитесь, что установлена версия 3.6 или выше, чтобы вы могли использовать все современные возможности языка. Обновите версию Python, если у вас более старая версия.

Важно! Во время установки убедитесь в том, что вы поставили галочку в разделе **Add Python 3.x to PATH**. В этом случае вызов интерпретатора Python будет доступен вам из командной строки. (Подробнее о том, что это за переменная такая PATH можно почитать здесь: <http://barancev.github.io/what-is-path-env-var/> (<http://barancev.github.io/what-is-path-env-var/>))

Python Releases for Windows

- [Latest Python 3 Release - Python 3.7.3](#)
- [Latest Python 2 Release - Python 2.7.16](#)

Stable Releases

- [Python 3.7.3 - March 25, 2019](#)
 - Note that Python 3.7.3 cannot be used on Windows XP or earlier.**
 - Download [Windows help file](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 web-based installer](#)
 - Download [Windows x86 embeddable zip file](#)
 - Download [Windows x86 executable installer](#)
 - Download [Windows x86 web-based installer](#)
- [Python 3.4.10 - March 18, 2019](#)
 - No files for this release.
- [Python 3.5.7 - March 18, 2019](#)

www.python.org/ftp/python/3.7.3/python-3.7.3-amd64.exe

Pre-releases

- [Python 3.8.0a3 - March 25, 2019](#)
 - Download [Windows help file](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 web-based installer](#)
 - Download [Windows x86 embeddable zip file](#)
 - Download [Windows x86 executable installer](#)
 - Download [Windows x86 web-based installer](#)
- [Python 3.7.3rc1 - March 12, 2019](#)
 - Download [Windows help file](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 web-based installer](#)

SCREENSHOTER@mail.ru

Запустите скачанный файл и следуйте инструкциям установщика.

Вот и все, через несколько минут у вас будет установлен интерпретатор Python!

Чтобы проверить правильность установки, откройте командную строку Windows (Пуск - Найти программу - cmd.exe и Запустить). В консоли введите `python --version` (на некоторых сборках Windows нужно вводить короткую команду, например, `py --version`):

```
python --version
Python 3.7.3
```

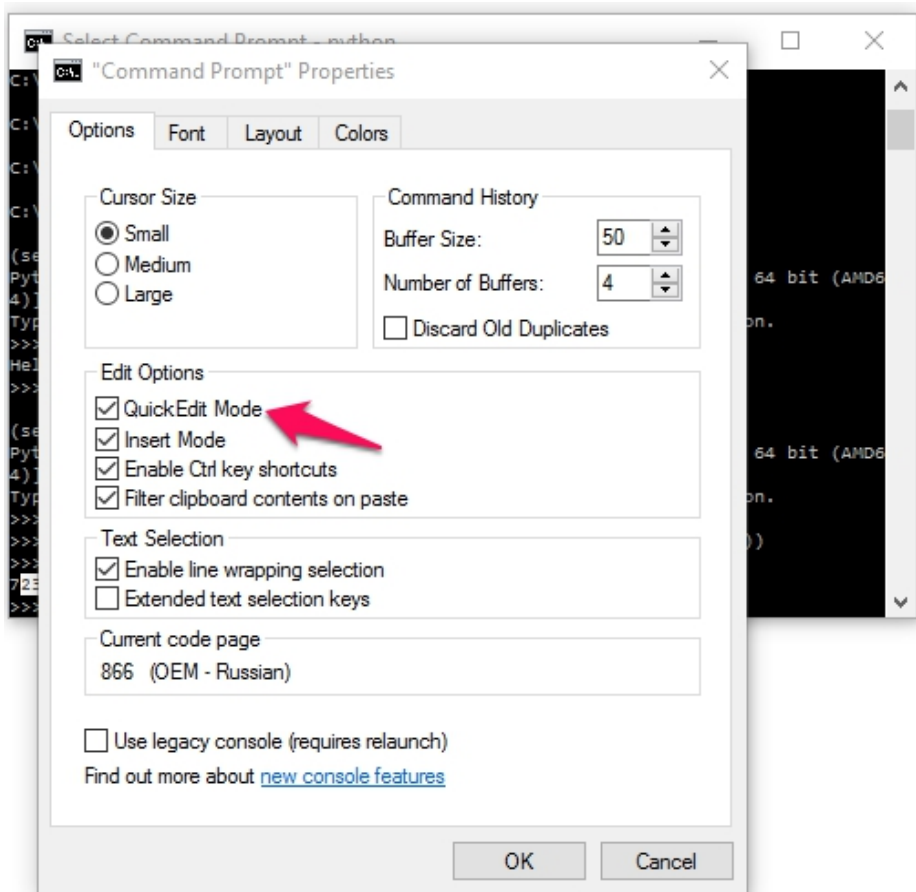
Если Python не найден, то попробуйте переустановить его еще раз или вручную добавить в переменную окружения PATH путь к директории, куда был установлен Python. Как добавлять путь в PATH в разных версиях Windows, вы можете прочитать здесь: <https://www.computerhope.com/issues/ch000549.htm> (<https://www.computerhope.com/issues/ch000549.htm>). После переустановки Python не забудьте перезапустить командную строку, чтобы изменения переменной окружения стали доступны.

Чтобы не засорять наше основное Python-окружение, мы будем работать в виртуальном окружении, которое в дальнейшем можно будет легко удалить или изменить под ваши нужды. Все пакеты для Python, которые мы будем устанавливать в курсе, будут доступны только в этом виртуальном окружении.

Внимание! У консоли Windows специфичные отношения с операциями копировать/вставить. Чтобы разрешить выделение текста мышкой, нужно сделать следующие настройки:

- Правой кнопкой нажать на заголовок окна, и выбрать в меню пункт "Настройки" (Settings)
- В настройках поставить галочку "Разрешить выделение мышью" (QuickEdit Mode). Найти настройку вам также поможет скриншот приведенный ниже.
- Сохранить настройки.

Теперь вы сможете выбрать текст в консоли, зажав левую кнопку мыши. После выделения текста и нажатия Enter текст окажется в буфере обмена. Чтобы вставить текст в консоль, нужно нажать правую кнопку мыши.



Создадим папку, где будут храниться наши виртуальные окружения, и перейдем в неё:

```
mkdir environments  
cd environments
```

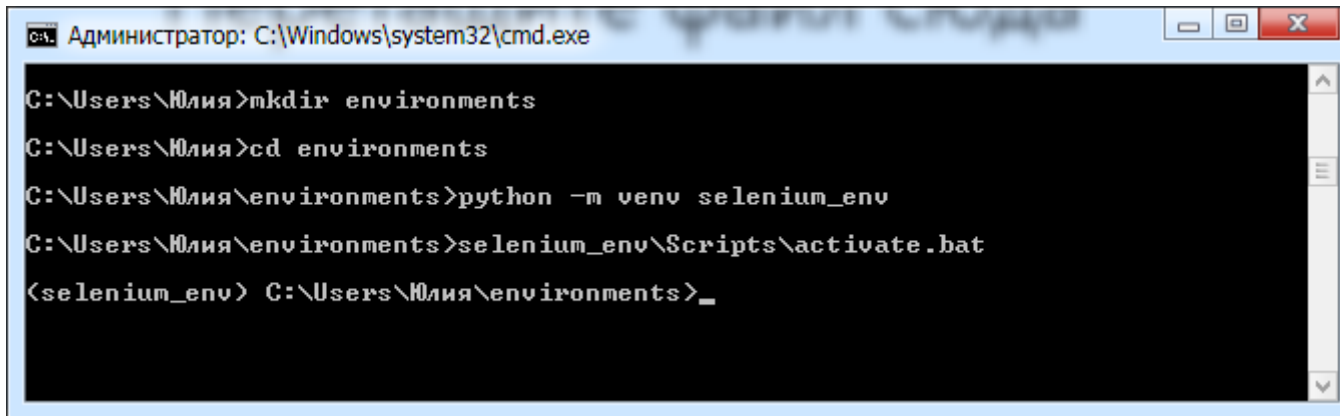
Создадим виртуальное окружение:

```
python -m venv selenium_env
```

Запустим созданный для нас приложением venv файл activate.bat, чтобы активировать окружение:

```
selenium_env\Scripts\activate.bat
```

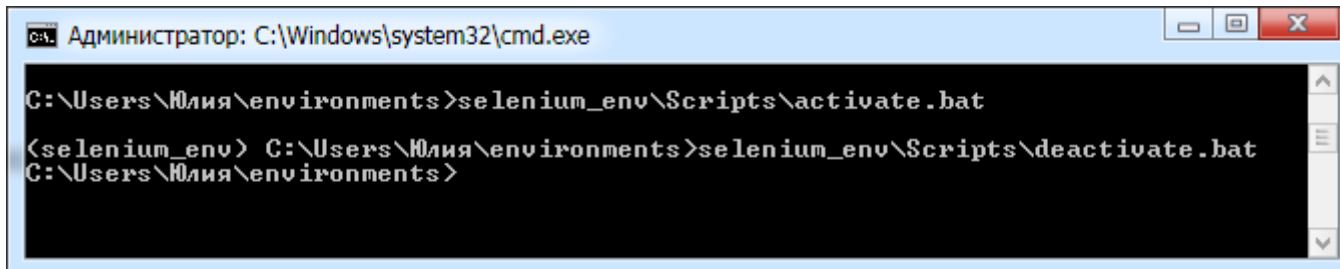
Если окружение активировано, то мы увидим в начале командной строки терминала название окружения в круглых скобках. Теперь мы можем устанавливать нужные нам пакеты и запускать скрипты для тестов, которые мы напишем:



```
Администратор: C:\Windows\system32\cmd.exe

C:\Users\Юлия>mkdir environments
C:\Users\Юлия>cd environments
C:\Users\Юлия\environments>python -m venv selenium_env
C:\Users\Юлия\environments>selenium_env\Scripts\activate.bat
(selenium_env) C:\Users\Юлия\environments>_
```

Если понадобится выйти из нашего окружения, достаточно выполнить команду deactivate:



```
Администратор: C:\Windows\system32\cmd.exe

C:\Users\Юлия\environments>selenium_env\Scripts\activate.bat
(selenium_env) C:\Users\Юлия\environments>selenium_env\Scripts\deactivate.bat
C:\Users\Юлия\environments>
```

Обратите внимание, что после перезапуска терминала необходимо снова активировать нужное окружение.

Проверим, что мы можем теперь использовать Python. Запустим интерпретатор Python и напишем собственную версию HelloWorld:

```
> selenium_env\Scripts\activate.bat

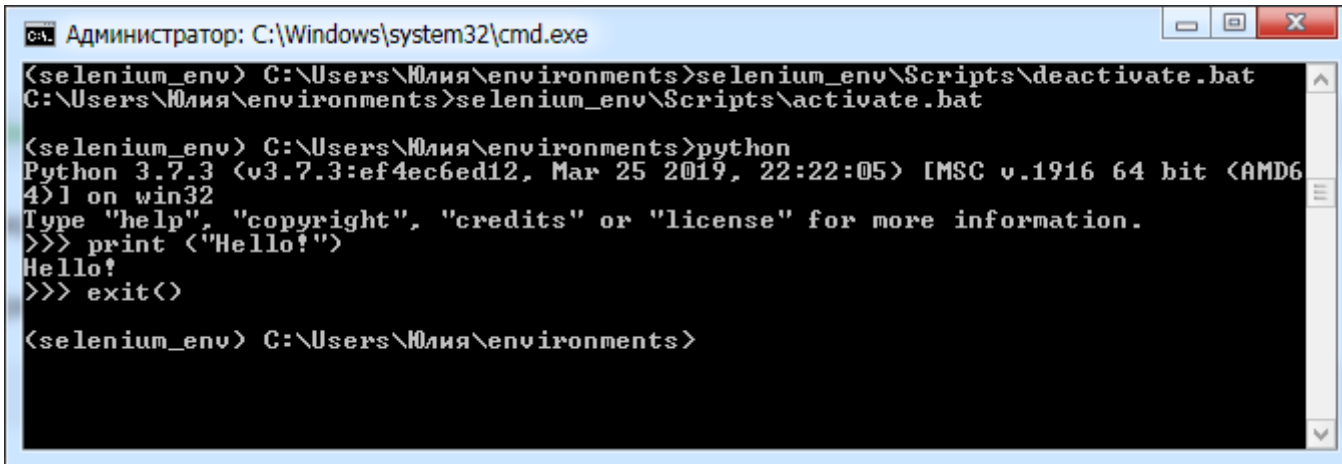
(selenium_env) C:\Users\Юлия\environments> python
```

```
>>> print("Hello, Selenium!")
```

В итоге вы должны увидеть вывод строки "Hello, Selenium! "

Выйдем из интерпретатора:

```
>>> exit()
```



```
Администратор: C:\Windows\system32\cmd.exe
(selenium_env) C:\Users\Юлия\environments>selenium_env\Scripts\deactivate.bat
C:\Users\Юлия\environments>selenium_env\Scripts\activate.bat

(selenium_env) C:\Users\Юлия\environments>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print ("Hello!")
Hello!
>>> exit()

(selenium_env) C:\Users\Юлия\environments>
```

Следующий шаг с настройками Python для Ubuntu вы можете пропустить.

Переходите к пятому шагу этого урока: <https://stepik.org/lesson/25969/step/5> (<https://stepik.org/lesson/25969/step/5>)

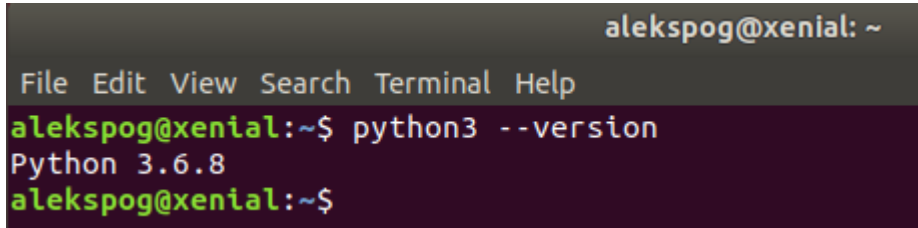
Шаг 3

Установка Python3 в Ubuntu

Если вы используете другую ОС, то смело пропускайте этот шаг и переходите к следующему.

Если вы пользуетесь Ubuntu, то обычно Python2 и Python3 у вас уже установлены. Чтобы проверить, какая версия Python3 у вас в системе, откройте терминал и введите команду:

```
python3 --version
```

A terminal window with a dark background. The prompt is 'alekspog@xenial: ~'. The menu bar shows 'File Edit View Search Terminal Help'. The command 'python3 --version' is entered, and the output 'Python 3.6.8' is displayed. The prompt returns to 'alekspog@xenial:~\$'.

Если вы получили ошибку Command 'python3' not found, то установите Python с помощью инструкции приведенной ниже.

Если Python у вас уже есть в системе, убедитесь, что установлена версия 3.6 или выше, чтобы вы могли использовать все современные возможности языка. Если у вас более старая версия, установите новую версию Python с помощью инструкции приведенной ниже.

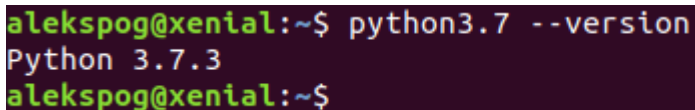
Установка Python3.7

Выполните в терминале:

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install python3.7
```

Python будет установлен в папку /usr/bin/python3.7. Теперь вы можете запускать интерпретатор python командой python3.7, а также посмотреть более детально версию python:

```
python3.7 --version
```

A terminal window with a dark background. The prompt is 'alekspog@xenial:~\$'. The command 'python3.7 --version' is entered, and the output 'Python 3.7.3' is displayed. The prompt returns to 'alekspog@xenial:~\$'.

Если вы хотите, чтобы данная версия Python была доступна по более короткой команде `python3`, то можно прописать `alias`, то есть короткое имя, следующим образом.

Открыть файл с настройками конфигураций для сеанса терминала `.bashrc`:

```
nano ~/.bashrc
```

Добавить в начало файла строчку:

```
alias python3=python3.7
```

Сохранить файл `ctrl+o` и закрыть его `ctrl+x`. В командной строке применить настройки:

```
source ~/.bashrc
```

Теперь данный `alias` должен стать постоянным. Проверим:

```
alekspog@xenial:~$ python3 --version
Python 3.7.3
alekspog@xenial:~$
```

Создание виртуального окружения

Для курса нам понадобятся дополнительные пакеты Python, для их дальнейшего использования установим систему управления пакетами `pip`:

```
python3 -m pip install pip
```

Чтобы не засорять наше основное Python-окружение, мы будем работать в виртуальном окружении, которое в дальнейшем можно будет легко удалить или изменить под ваши нужды. Установим модуль `venv` для управления виртуальными окружениями:

```
sudo apt-get install -y python3.7-venv
```


Отлично! Давайте теперь создадим виртуальное окружение и активируем его. Все пакеты для Python, которые мы будем устанавливать в курсе, будут доступны только в этом виртуальном окружении. Следующие команды мы продолжим выполнять в консоли.

Создадим папку, где будут храниться наши виртуальные окружения, и перейдем в неё.

```
$ mkdir environments  
$ cd environments
```

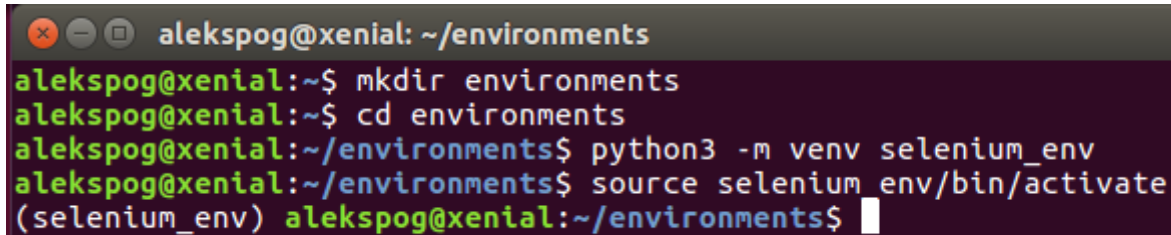
Создадим виртуальное окружение:

```
$ python3 -m venv selenium_env
```

Активируем окружение:

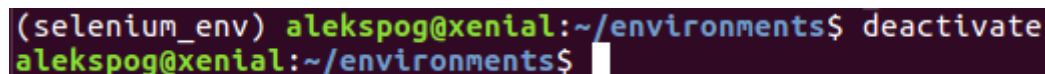
```
$ source selenium_env/bin/activate
```

Если окружение активировано, то мы увидим в начале командной строки терминала название окружения в круглых скобках. Теперь мы можем устанавливать нужные нам пакеты и запускать скрипты для тестов, которые мы напишем:



```
alekspog@xenial: ~/environments  
alekspog@xenial:~$ mkdir environments  
alekspog@xenial:~$ cd environments  
alekspog@xenial:~/environments$ python3 -m venv selenium_env  
alekspog@xenial:~/environments$ source selenium_env/bin/activate  
(selenium_env) alekspog@xenial:~/environments$
```

Если понадобится выйти из нашего окружения, достаточно ввести команду deactivate:



```
(selenium_env) alekspog@xenial:~/environments$ deactivate  
alekspog@xenial:~/environments$
```

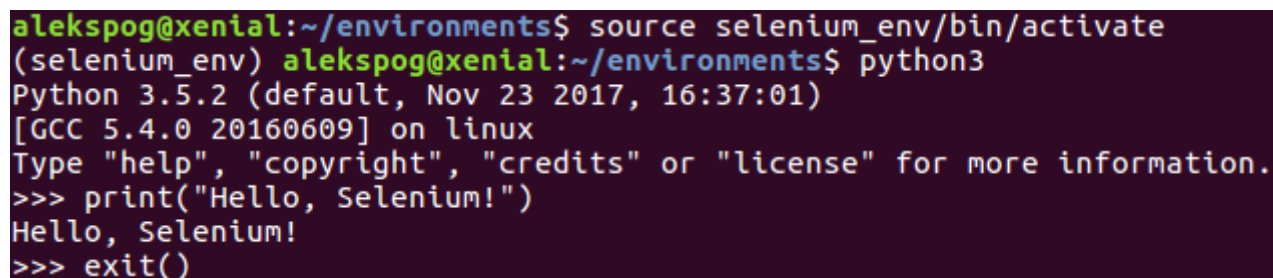
Обратите внимание, что после перезапуска терминала нужно будет снова активировать нужное окружение.

Проверим, что мы можем теперь использовать Python. Запустим интерпретатор Python и напомним собственную версию HelloWorld. В итоге мы должны увидеть вывод строки "Hello, Selenium!":

```
$ source selenium_env/bin/activate  
  
(selenium_env) alekspog@xenial:~/environments$ python  
  
>>> print("Hello, Selenium!")  
  
Hello, Selenium!
```

Теперь выйдем из интерпретатора:

```
>>> exit()
```



```
alekspog@xenial:~/environments$ source selenium_env/bin/activate  
(selenium_env) alekspog@xenial:~/environments$ python3  
Python 3.5.2 (default, Nov 23 2017, 16:37:01)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello, Selenium!")  
Hello, Selenium!  
>>> exit()
```

Будьте внимательны: когда вы открываете новое окно терминала, нужно заново активировать виртуальное окружение.

Теперь можете переходить к пятому шагу этого урока: <https://stepik.org/lesson/25969/step/5>

(<https://stepik.org/lesson/25969/step/5>)

Шаг 4

Установка Python3 в macOS

Если вы используете другую ОС, то смело пропускайте этот шаг и переходите к следующему.

Чтобы проверить установлен ли у вас Python3 в системе, в консоли введите команду `python3 --version`:

```
python3 --version  
Python 3.7.3
```

Если Python у вас уже есть в системе, убедитесь, что установлена версия 3.6 или выше, чтобы вы могли использовать все современные возможности языка. Если у вас более старая версия или вы получили сообщение 'command not found', установите последнюю версию Python с помощью инструкции приведенной ниже.

Для установки Python3 в macOS воспользуйтесь инструкцией <https://wsvincent.com/install-python/#install-python-on-macos> (<https://wsvincent.com/install-python/#install-python-on-macos>). Рекомендуем установить пакетный менеджер Homebrew, как указано в инструкции. С помощью него потом будет удобно устанавливать другие инструменты, нужные для разработки автотестов.

После установки откройте окно терминала и создайте папку, где будут храниться наши виртуальные окружения, затем перейдите в созданную папку:

```
$ mkdir environments  
$ cd environments
```

Создадим виртуальное окружение с помощью команды `python3`:

```
$ python3 -m venv selenium_env
```

Активируем окружение:

```
$ source selenium_env/bin/activate
```

Если окружение активировано, то мы увидим в начале командной строки терминала название окружения в круглых скобках. Теперь мы можем устанавливать нужные нам пакеты и запускать скрипты для тестов, которые мы напишем:

```
alekspog@xenial: ~/environments
alekspog@xenial:~$ mkdir environments
alekspog@xenial:~$ cd environments
alekspog@xenial:~/environments$ python3 -m venv selenium_env
alekspog@xenial:~/environments$ source selenium env/bin/activate
(selenium_env) alekspog@xenial:~/environments$
```

Если понадобится выйти из нашего окружения, достаточно ввести команду deactivate:

```
(selenium_env) alekspog@xenial:~/environments$ deactivate
alekspog@xenial:~/environments$
```

Обратите внимание, что после перезапуска терминала нужно будет снова активировать нужное окружение.

Проверим, что мы можем теперь использовать Python. Запустим интерпретатор Python и напомним собственную версию HelloWorld. В итоге мы должны увидеть вывод строки "Hello, Selenium!":

```
$ source selenium_env/bin/activate

(selenium_env) alekspog@xenial:~/environments$ python

>>> print("Hello, Selenium!")

Hello, Selenium
```

Теперь выйдем из интерпретатора:

```
>>> exit()
```

```
vyafs-MBP:environments alekspog$ source selenium_env/bin/activate  
(selenium_env) vyafs-MBP:environments alekspog$ python3  
Python 3.7.4 (default, Jul 9 2019, 18:13:23)  
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello, Selenium!")  
Hello, Selenium!  
>>> exit()  
(selenium_env) vyafs-MBP:environments alekspog$
```

Будьте внимательны: когда вы открываете новое окно терминала, нужно заново активировать виртуальное окружение.

Теперь можете переходить к следующему шагу этого урока.

Шаг 5

Задание: интерпретатор Python

Чтобы мы точно знали, что вы выполнили все инструкции и Python успешно установлен на ваш компьютер, запустите следующий код в интерпретаторе:

```
import math  
fun = lambda x : 1 if x == 1 else math.ceil(math.sinh(fun (x-1)))  
print(fun(5))
```

Чтобы решить это задание откройте <https://stepik.org/lesson/25969/step/5>

Установка Selenium для Python

Теперь мы почти готовы запустить браузер с помощью Python и Selenium, и выполнить в нём простые действия. Но предварительно нам еще надо установить библиотеку Selenium для Python и драйвер — программу, которая позволяет взаимодействовать с браузером при помощи Selenium.

В нашем виртуальном окружении установим библиотеку Selenium:

```
pip install selenium==4.*
```

Проект Selenium развивается очень быстро и в вашей установленной версии библиотеки могут быть расхождения в методах или в том, что библиотека выводит в терминал при выполнении скриптов. Если вы просите помощи в комментариях, пожалуйста, указывайте версии python-библиотек, которые вы используете. Версии библиотек в вашем виртуальном окружении можно узнать с помощью команды `pip list`.

Проверим, что библиотека действительно установлена:

```
pip list
```

Ubuntu (скриншот приведён для старой версии библиотеки, но вы должны увидеть похожие логи в терминале):

```
(selenium_env) user@unit-652:~/pytest testing$ pip install selenium==3.14.0
Collecting selenium==3.14.0
  Using cached https://files.pythonhosted.org/packages/b8/53/9cafbb616d20c7624ff
31bcabd82e5cc9823206267664e68aa8acdde4629/selenium-3.14.0-py2.py3-none-any.whl
Collecting urllib3 (from selenium==3.14.0)
  Using cached https://files.pythonhosted.org/packages/e6/60/247f23a7121ae632d62
811ba7f273d0e58972d75e58a94d329d51550a47d/urllib3-1.25.3-py2.py3-none-any.whl
Installing collected packages: urllib3, selenium
Successfully installed selenium-3.14.0 urllib3-1.25.3
You are using pip version 18.1, however version 19.2.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(selenium_env) user@unit-652:~/pytest testing$ pip list
Package      Version
-----
pip          18.1
selenium     3.14.0
setuptools   40.6.2
urllib3      1.25.3
You are using pip version 18.1, however version 19.2.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(selenium_env) user@unit-652:~/pytest testing$
```

macOS(скриншот приведён для старой версии библиотеки, но вы должны увидеть похожие логи в терминале):

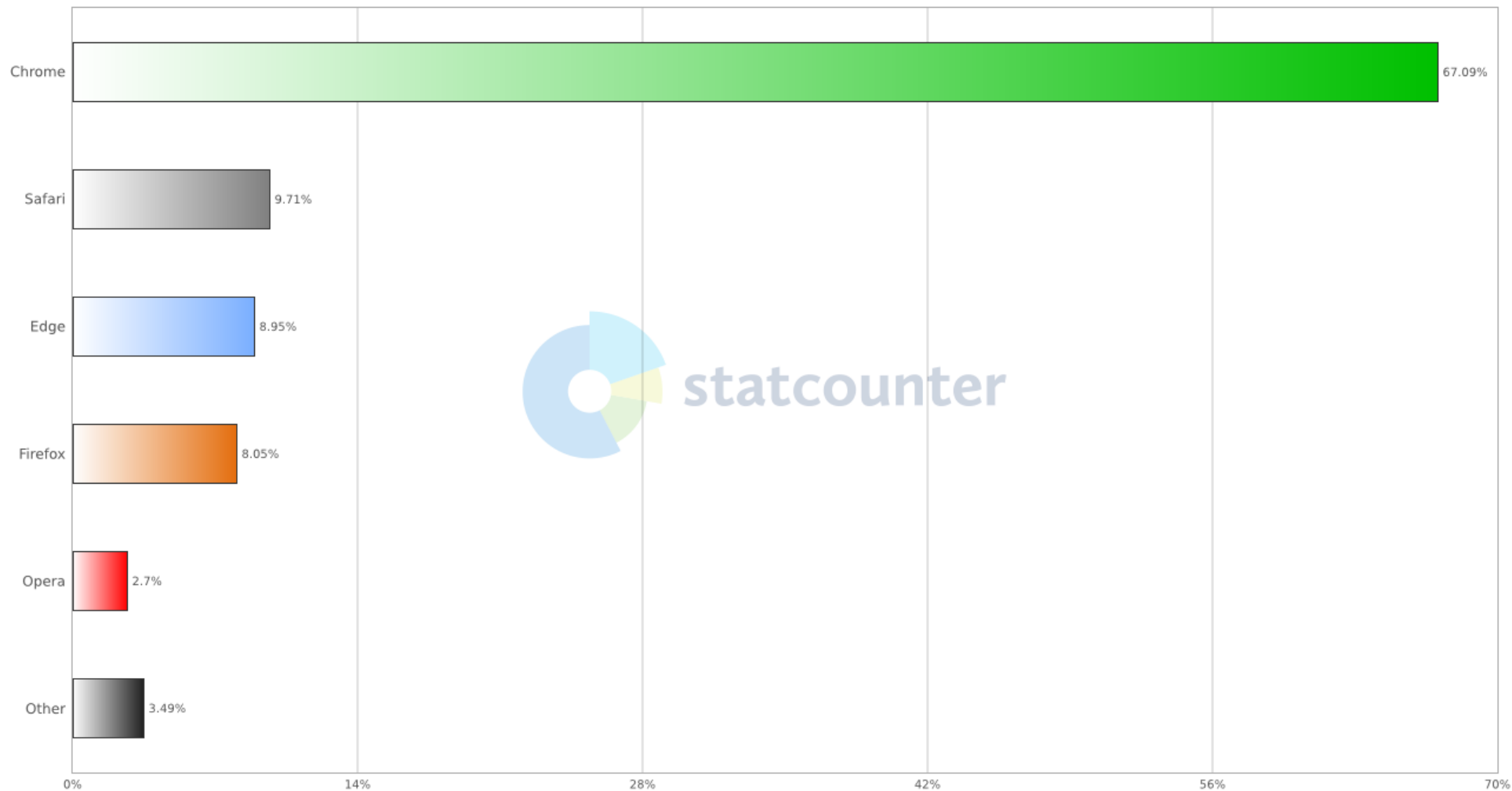
```
(selenium_env) vyafs-MacBook-Pro:environments alekspog$ pip3 install selenium==3.14.0
Collecting selenium==3.14.0
  Using cached https://files.pythonhosted.org/packages/b8/53/9cafbb616d20c7624ff31bcabd82e5cc9823206267664e68aa8acdde4629/selenium-3.14.0-py2.py3-none-any.whl
Collecting urllib3 (from selenium==3.14.0)
  Using cached https://files.pythonhosted.org/packages/e6/60/247f23a7121ae632d62811ba7f273d0e58972d75e58a94d329d51550a47d/urllib3-1.25.3-py2.py3-none-any.whl
Installing collected packages: urllib3, selenium
Successfully installed selenium-3.14.0 urllib3-1.25.3
You are using pip version 19.0.3, however version 19.2.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(selenium_env) vyafs-MacBook-Pro:environments alekspog$ pip3 list
Package      Version
-----
pip          19.0.3
selenium     3.14.0
setuptools   40.8.0
urllib3      1.25.3
You are using pip version 19.0.3, however version 19.2.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(selenium_env) vyafs-MacBook-Pro:environments alekspog$
```

Шаг 7

Установка драйвера для браузера

В этом курсе мы будем работать с драйвером для Chrome, так как на данный момент это самый популярный браузер, и в первую очередь следует убедиться, что веб-приложение работает для большинства пользователей.

StatCounter Global Stats
Desktop Browser Market Share Worldwide from Apr 2021 - Apr 2022



<https://gs.statcounter.com/browser-market-share/desktop/worldwide/> (<https://gs.statcounter.com/browser-market-share/desktop/worldwide/>)

Драйвер для Chrome разрабатывается командой браузера и носит название ChromeDriver. Скачать нужную версию можно с официального сайта по ссылке <https://sites.google.com/chromium.org/driver/downloads> (<https://sites.google.com/chromium.org/driver/downloads>) (старый сайт <https://sites.google.com/a/chromium.org/chromedriver/downloads>).

Более подробные инструкции по установке для разных ОС смотрите в следующих шагах.

Установка драйвера для браузера: Windows

Для установки откройте сайт <https://sites.google.com/chromium.org/driver/> (<https://sites.google.com/chromium.org/driver/>) (старая версия сайта <https://sites.google.com/a/chromium.org/chromedriver/downloads> (<https://sites.google.com/a/chromium.org/chromedriver/downloads>)) и скачайте ту версию ChromeDriver, которая соответствует версии вашего браузера. Чтобы узнать версию браузера, откройте новое окно в Chrome, в поисковой строке наберите: `chrome://version/` и нажмите Enter. В верхней строчке вы увидите информацию про версию браузера.

- Скачайте с сайта <https://sites.google.com/chromium.org/driver/> (<https://sites.google.com/chromium.org/driver/>) (старая версия сайта <https://sites.google.com/a/chromium.org/chromedriver/downloads> (<https://sites.google.com/a/chromium.org/chromedriver/downloads>)) драйвер для вашей версии браузера. Разархивируйте скачанный файл.
- Создайте на диске C: папку `chromedriver` и положите разархивированный ранее файл `chromedriver.exe` в папку `C:\chromedriver`.
- Добавьте в системную переменную `PATH` папку `C:\chromedriver`. Как это сделать в разных версиях Windows, описано здесь: <https://www.computerhope.com/issues/ch000549.htm> (<https://www.computerhope.com/issues/ch000549.htm>).

Пример: как добавить путь в системную переменную `PATH` на Windows10

1. Откройте настройки системы.
2. В настройках откройте вкладку About, затем System info:

- Tablet mode
- Multitasking
- Projecting to this PC
- Apps for websites

About



[Change product key or upgrade your edition of Windows](#)

[Read the Privacy Statement for Windows and Microsoft services](#)

[Read the Microsoft Services Agreement that applies to our services](#)

[Read the Microsoft Software License Terms](#)

Related settings

[Additional administrative tools](#)

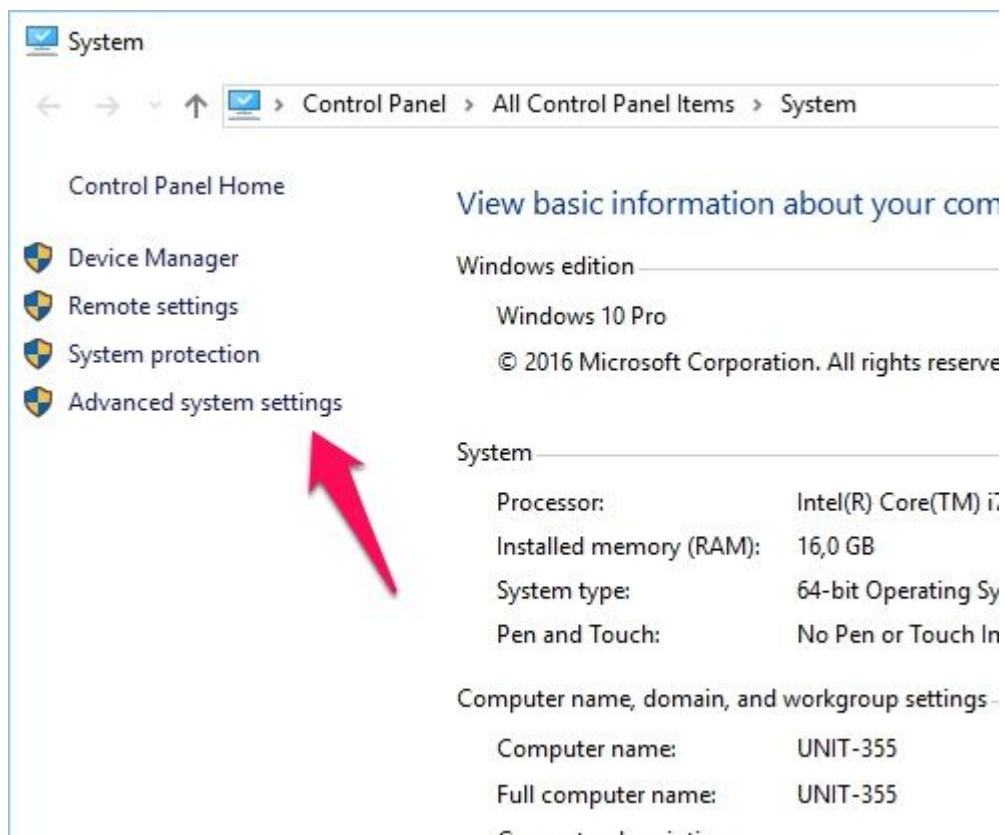
[Bitlocker settings](#)

[Device manager](#)

[System info](#)



3. Выберите Advanced system settings:

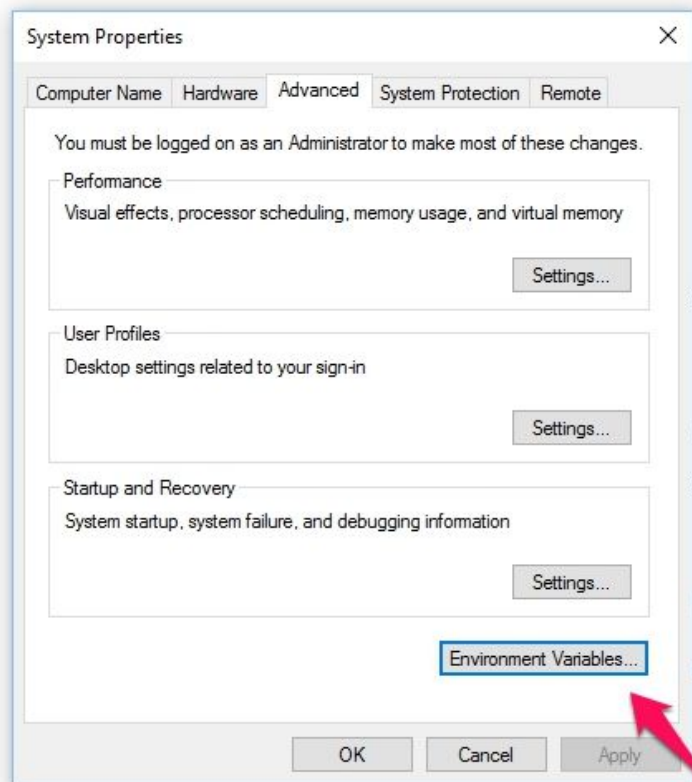


4. Выберите Environment Variables:

Rename PC

Organization

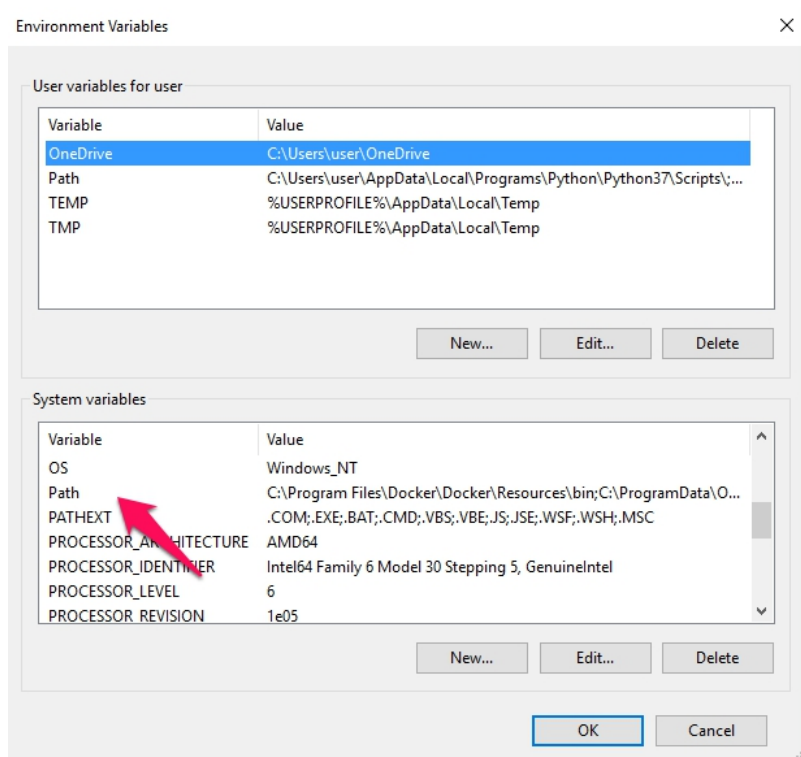
LABS



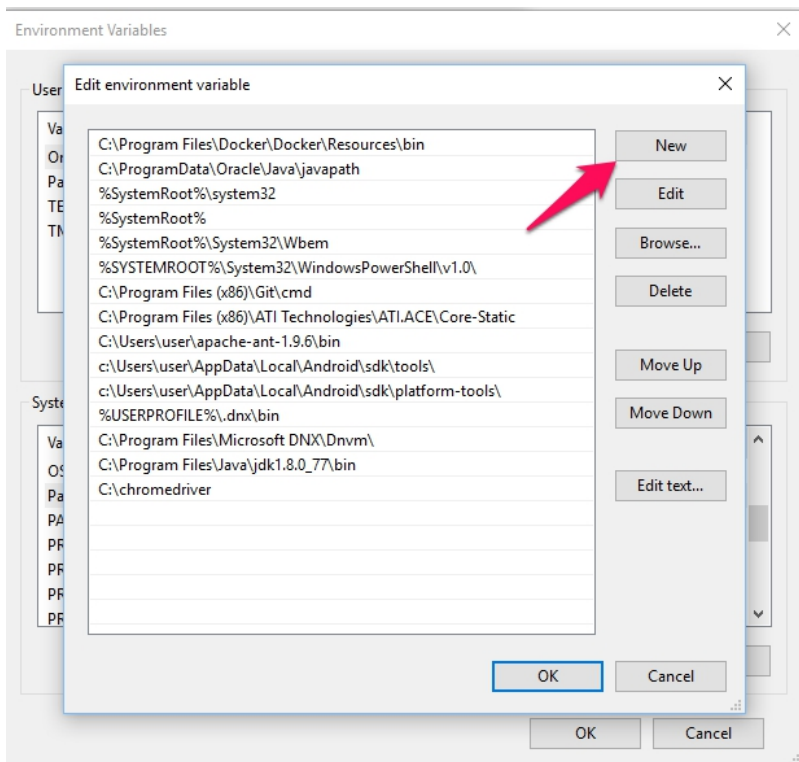
Related settings

[Additional administrative tools](#)

5. Кликните два раза на строчке Path в System variables:



6. Нажмите кнопку New. Введите в новую строку путь к ChromeDriver — C:\chromedriver. Нажмите Enter. У вас должна появиться строка с указанным путем:



7. Если у вас была открыта командная строка Windows, не забудьте ее закрыть. Затем откройте новую командную строку, чтобы изменения переменной окружения стали доступны. Активируйте снова виртуальное окружение `selenium_env`, которое мы создали в предыдущих шагах.

Давайте убедимся в том, что вебдрайвер установлен правильно.

Для начала проверим содержимое переменной `path`, для этого наберем в командной строке **Path**:

```
Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Юлия>environments\selenium_env\Scripts\activate.bat

(selenium_env) C:\Users\Юлия>path
PATH=C:\Users\Юлия\environments\selenium_env\Scripts;D:\chromedriver.exe;C:\Users\Юлия\AppData\Local\Programs\Python\Python37\Scripts\;C:\Users\Юлия\AppData\Local\Programs\Python\Python37\;C:\chromedriver
(selenium_env) C:\Users\Юлия>
```

Ура, там есть папка с chromedriver! Попробуем вызвать его напрямую из командной строки:

```
<selenium_env> C:\Users\Юлия>chromedriver
Starting ChromeDriver 2.46.628402 (536cd7adbad73a3783fdc2cab92ab2ba7ec361e1) on
port 9515
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent
access by malicious code.
```

Магия переменной path: хотя программа chromedriver находится где-то в другом каталоге, мы можем напрямую открывать её, используя имя chromedriver. Чтобы завершить процесс в консоли, нажмите Ctrl+C.

Знак на этом этапе, что пошло что-то не так:

```
C:\Users\Юлия>chromedriver
"chromedriver" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
```

В таком случае попробуйте перезапустить консоль, перезапустить компьютер, перепроверить и добавить заново по инструкциям папку с chromedriver в переменную path.

Когда все получится, переходите к шагу [https://stepik.org/lesson/25969/step/11 \(/lesson/25969/step/11\)](https://stepik.org/lesson/25969/step/11 (/lesson/25969/step/11)), в котором мы запустим браузер с помощью Selenium WebDriver и выполним простые команды.

Шаг 9

Установка драйвера для браузера: Linux

Давайте установим и настроим ChromeDriver с помощью команд в терминале. Укажем нужную нам версию ChromeDriver для загрузки. Для получения ссылки перейдите в браузере на нужную вам версию драйвера по ссылке на <https://sites.google.com/chromium.org/driver/> (<https://sites.google.com/chromium.org/driver/>). На открывшейся странице нажмите на файле для Linux правой кнопкой и скопируйте путь к файлу. Замените в примере ниже путь к файлу для команды wget вашей ссылкой:


```
wget https://chromedriver.storage.googleapis.com/102.0.5005.61/chromedriver_linux64.zip
unzip chromedriver_linux64.zip
```

Переместите разархивированный файл с ChromeDriver в нужную папку и разрешите запускать chromedriver как исполняемый файл:

```
sudo mv chromedriver /usr/local/bin/chromedriver
sudo chown root:root /usr/local/bin/chromedriver
sudo chmod +x /usr/local/bin/chromedriver
```

Проверьте, что chromedriver доступен, выполнив команду chromedriver в терминале, вы должны получить сообщение о том, что процесс успешно запущен:

```
> chromedriver
Starting ChromeDriver 102.0.5005.61 (0e59bcc00cc4985ce39ad31c150065f159d95ad3-re
fs/branch-heads/5005@{#819}) on port 9515
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggest
ions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

Если вы получили сообщение "Command 'chromedriver' not found":

```
stephen@stephen-Aspire-A315-53: ~
stephen@stephen-Aspire-A315-53:~$ chromedriver
chromedriver: command not found
stephen@stephen-Aspire-A315-53:~$
```

То повторите установку драйвера по инструкциям выше.

Настройки для Linux готовы. Переходите к шагу [https://stepik.org/lesson/25969/step/11 \(/lesson/25969/step/11\)](https://stepik.org/lesson/25969/step/11 (/lesson/25969/step/11)), в котором мы запустим браузер с помощью Selenium WebDriver и выполним простые команды.

Шаг 10

Установка драйвера для браузера: macOS

При установке Python вы уже, скорее всего, установили пакетный менеджер Homebrew. Если нет, то рекомендуем сделать это сейчас, а затем с его помощью установить программу `wget` для загрузки файлов по сети.

```
brew install wget
```

Для установки драйвера откройте сайт <https://sites.google.com/chromium.org/driver/> (<https://sites.google.com/chromium.org/driver/>) и скопируйте ссылку на ту версию ChromeDriver, которая соответствует версии вашего браузера. Чтобы узнать версию браузера, откройте новое окно в Chrome, в поисковой строке наберите: `chrome://version/` — и нажмите Enter. В верхней строке вы увидите информацию про версию браузера.

```
cd ~/Downloads  
wget https://chromedriver.storage.googleapis.com/76.0.3809.68/chromedriver_mac64.zip
```

Разархивируйте скачанный файл и переместите его в папку `/usr/local/bin`, чтобы он был глобально доступен в вашей системе.

```
unzip chromedriver_mac64.zip  
sudo mv chromedriver /usr/local/bin
```

Проверим, что нужная версия `chromedriver` установлена.

```
chromedriver --version
```

Мы должны увидеть ответ системы:

```
ChromeDriver 76.0.3809.68 (420c9498db8ce8fcd190a954d51297672c1515d5-refs/branch-heads/3809@{#864})
```

В этом случае все хорошо, и можно переходить к следующему шагу.

Если мы вдруг видим что-то вроде:

```
-bash: chromedriver: command not found
```

то необходимо проверить папку /usr/local/bin на наличие файла chromedriver. Если файла там нет, то нужно повторить команды, описанные выше.

Переходите к шагу [https://stepik.org/lesson/25969/step/11 \(/lesson/25969/step/11\)](https://stepik.org/lesson/25969/step/11 (/lesson/25969/step/11)), в котором мы запустим браузер с помощью Selenium WebDriver и выполним простые команды.

Шаг 11

Запуск браузера и первый скрипт

Для всех ОС:

В нашем виртуальном окружении запустим интерпретатор python:

```
~/environments$ python
```

Теперь напишем скрипт, который откроет простую веб-страницу и заполнит форму. Рекомендуем выполнять команды последовательно по одной, чтобы лучше понять, что происходит в каждой строчке. Не пугайтесь, если не понимаете, что делают те или иные команды, — подробнее мы их разберем в следующих уроках.

```
import time

# webdriver это и есть набор команд для управления браузером
from selenium import webdriver

# импортируем класс By, который позволяет выбрать способ поиска элемента
from selenium.webdriver.common.by import By

# инициализируем драйвер браузера. После этой команды вы должны увидеть новое открытое окно браузера
driver = webdriver.Chrome()

# команда time.sleep устанавливает паузу в 5 секунд, чтобы мы успели увидеть, что происходит в браузере
time.sleep(5)

# Метод get сообщает браузеру, что нужно открыть сайт по указанной ссылке
driver.get("https://suninjuly.github.io/text_input_task.html")
time.sleep(5)

# Метод find_element позволяет найти нужный элемент на сайте, указав путь к нему. Способы поиска элементов
мы обсудим позже
# Метод принимает в качестве аргументов способ поиска и значение, по которому мы будем искать
# Ищем поле для ввода текста
textarea = driver.find_element(By.CSS_SELECTOR, ".textarea")

# Напишем текст ответа в найденное поле
textarea.send_keys("get()")
time.sleep(5)

# Найдем кнопку, которая отправляет введенное решение
submit_button = driver.find_element(By.CSS_SELECTOR, ".submit-submission")

# Скажем драйверу, что нужно нажать на кнопку. После этой команды мы должны увидеть сообщение о правильном
ответе
submit_button.click()
time.sleep(5)

# После выполнения всех действий мы должны не забыть закрыть окно браузера
driver.quit()
```

Вы можете запустить этот же скрипт из файла. Скачайте файл `first_script.py` (https://stepik.org/media/attachments/lesson/25969/first_script.py) к себе в папку и запустите его. Для этого создадим папку `selenium_course` и скопируем в нее скачанный файл. Проверим линк (он должен быть таким, как в уроке). Затем запустим файл в нашем виртуальном окружении.
(может появиться предупреждение, но пока можно на него не обращать внимание)

Пример для Windows

(Вместо `user` везде подставляйте имя пользователя в системе)

При выполнении следующих команд не забудьте изменить путь к вашей пользовательской папке вместо `C:\Users\user` и к папке с загруженным файлом.

Выполните следующие команды в командной строке Windows:

```
c:\Users\user> mkdir selenium_course  
  
C:\Users\user> copy C:\Users\user\Downloads\first_script.py c:\Users\user\selenium_course  
  
C:\Users\user> environments\selenium_env\Scripts\activate.bat  
  
(selenium_env) C:\Users\user>python c:\Users\user\selenium_course\first_script.py
```

Пример для Ubuntu и macOS

Выполните следующие команды в консоли:

```
mkdir ~/selenium_course  
mv ~/Downloads/first_script.py ~/selenium_course  
python ~/selenium_course/first_script.py
```

Задание

Какой метод используется в Selenium WebDriver, чтобы открыть веб-страницу в браузере?

Чтобы решить это задание откройте <https://stepik.org/lesson/25969/step/12>