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.

Запустите скачанный файл и следуйте инструкциям установщика.

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. 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.

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

mkdir environments

cd environments

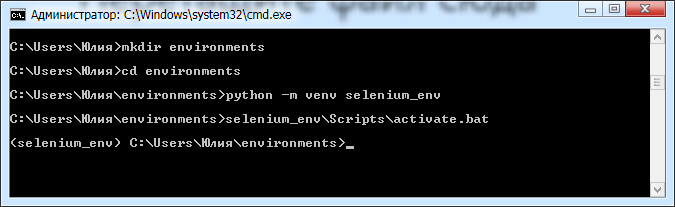
Создадим виртуальное окружение:

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:



Если понадобится выйти из нашего окружения, достаточно выполнить команду deactivate.4 selenium\_env\Scripts\deactivate.bat

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) С:\Users\Юлия\environments> python

 >>> print("Hello, Selenium!")

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

Выйдем из интерпретатора:

>>> exit()

**Установка Selenium для Python**

В нашем виртуальном окружении установим библиотеку Selenium:

pip install selenium==4.\*

Проверим, что библиотека действительно установлена:

pip list

## 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.

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 в виртуальном окружении, указав путь к файлу..

*(Вместо user подставляйте имя пользователя в системе)*

(selenium\_env) C:\Users\user>python c:\Users\user\selenium\_course\first\_script.py