

Несмотря на то, что большинство заданий может быть решено с использованием Colab, настройка локального окружения крайне полезна. Ниже мы шаг за шагом объясним, как именно это сделать.

Это руководство может показаться немного сложным, когда вы читаете его в первый раз. Тем не менее, в нем нет ничего страшного. Просто действуйте шаг за шагом. Если вы настроите окружение прямо сейчас, это значительно сэкономит вам время в будущем.

Общий комментарий: всегда читайте вывод консоли, особенно когда она выбрасывает вам предупреждения / ошибки. Если вы хотите найти решение проблемы в сети или задать вопрос, указывайте как введенную команду, так и выходные данные (например, `pip install numpy`). Для форматирования кода в telegram можно использовать знак ``` (например, ``ls``). Многострочные примеры должны быть заключены в `````, например:
...

```
1 | this
2 | is
3 | code
```

...

0. Пользователи Windows

Если вы используете обновленную Windows 10 или 11, вы можете использовать WSL (Windows Subsystem Linux) ([официальная документация](#)).

Также есть полезный [пост на habr](#).

1. Настройка локального окружения

Здесь мы предполагаем, что вы уже пользуетесь окружением Linux. Инструкция для macOS аналогична за исключением нескольких ссылок (например, для загрузки miniconda).

Для работы с Python мы рекомендуем вам использовать `conda`. Лучше использовать Miniconda (~50 мегабайт) вместо Anaconda (~2 ГБ). Ниже доступна инструкция по установке и созданию локального environment.

1. Скачайте Miniconda с [официального веб-сайта](#). Вы можете сделать это непосредственно из командной строки используя `wget`:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

2. Для установки нужно запустить скачанный скрипт

```
bash Miniconda3-latest-Linux-x86_64.sh
```

Он попросит вас ознакомиться с лицензией. Чтобы читать его построчно вы можете написать Return(Enter). Чтобы перейти к концу файла лицензии можете нажать **q**. Чтобы принять условия лицензии, введите **yes**:

```
Do you accept the license terms? [yes|no]
[no] >>> yes
```

Путь установки можно не менять. Далее вас спросят, инициализировать ли Miniconda3. Отвечайте **yes**:

```
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

Рекомендуем прочитать вывод консоли.

3. Закройте терминал и откройте заново. Это наиболее простой способ инициализировать **conda** в текущей сессии терминала.
4. Теперь перед строкой ввода в консоле должно быть написано **(base)**. Чтобы явно проверить, что **conda** установлена, введите:

```
conda --version
```

Актуальная версия значительно превосходит 4.8.3. Установщик теперь можно удалить:

```
rm Miniconda3-latest-Linux-x86_64.sh
```

5. Теперь установим **jupyter**. Его можно устанавливать и в окружение **base**.

```
pip install jupyter
```

6. Используя **conda**, теперь вы можете создавать виртуальные окружения. **Мы рекомендуем вам использовать конкретное окружение для конкретных проектов вместо системного python.** Чтобы создать окружение (environment), можно написать, которую вы можете назвать **conda**.

```
conda create --name py3_study python=3.9
```

Для получения дополнительной информации вы можете обратиться к [документации](#).
Версия python 3.9 достаточна для прохождения данного курса. Окружение называется `py3_study`. Для его активации вызовите:

```
conda activate py3_study
```

Теперь вы находитесь в окружении `py3_study`. Перед строкой ввода в консоли вы должны видеть префикс `(py3_study)`.

6. Установите все необходимые библиотеки. На текущий момент достаточно следующих библиотек:

```
pip install ipykernel numpy matplotlib scikit-learn pandas tabulate tqdm torch
```

7. Теперь создадим ядро (kernel) для *jupyter notebook*:

```
python -m ipykernel install --name py3_study --display-name "Py3 study" --user
```

Теперь данная среда должна быть доступна и из `jupyter`.

8. Проверим, что установка прошла успешно. Введите `python` попробуйте импортировать `numpy`:

```
(py3_study) test_user@newmachine.phystech.edu:~$ python
Python 3.7.9 (default, Aug 31 2020, 12:42:55)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> quit()
(py3_study) test_user@newmachine.phystech.edu:~$
```

9. On the next step we will connect jupyter with this machine via port-forwarding.

3. Использование jupyter

1. Создайте файл `run_jupyter.sh` в той директории, где вы будете использовать `jupyter`. Чтобы добавить в него команду для запуска jupyter, вы можете

использовать `echo`. Изменить его потом можно через любой текстовый редактор. Также необходимо добавить файлу права на запуск:

```
touch run_jupyter.sh
echo "jupyter notebook --no-browser --port 8001 --ip 0.0.0.0">run_jupyter.sh
chmod +x run_jupyter.sh
```

Обратите внимание, в команде выше указывается порт, на котором будет запущен ваш `jupyter`. Если вы его меняете, вам необходимо обращаться к тому порту, который вы указали.

2. Создайте пароль для запуска `jupyter`. Дополнительная информация доступна в [документации](#).

```
(base) test_user@newmachine.phystech.edu:~$ jupyter notebook --generate-
config
Writing default config to:
/home/test_user/.jupyter/jupyter_notebook_config.py
(base) test_user@newmachine.phystech.edu:~$ jupyter notebook password
Enter password:
Verify password:
[NotebookPasswordApp] Wrote hashed password to
/home/test_user/.jupyter/jupyter_notebook_config.json
```

3. Теперь запустим `jupyter`. На удаленной машине рекомендуем использовать `tmux`.

```
./run_jupyter.sh
```

4. Наконец, откройте браузер и введите `localhost:8001` (или тот порт, который вы указали выше). Введите пароль и вы увидите перед собой `jupyter`. Для создания нового ноутбука воспользуйтесь кнопкой в правом верхнем углу.

Поздравляем! На этом установка завершена!

Если вы хотите настроить `jupyter` на удаленной машине, вы можете воспользоваться [расширенной версией данной инструкции](#).