

Привет,

Я решил поступить следующим образом: не смотря ни на что, методичку я буду отправлять вам только в воскресенье. Между вторниками - неделя, в течение этой недели вы можете что-то пытаться делать. Непонятно чего - пишите мне [Тут](#) я отвечаю и пытаюсь помочь.

Здесь и далее будет небольшой ликбез по установке и настройке среды для программирования на языке Python.

Прежде чем начнём, я поделюсь некоторыми ресурсами, которые помогли мне в той или иной мере приобщиться к программированию как таковому, ну и его немножко развить:

Самое большое место со всякими вопросами касательно того, "как это закодить" stackoverflow.com

Или, если английский пока сложен для восприятия: ru.stackoverflow.com

Конечно, я буду настоятельно вам рекомендовать, и давать, примеры на английском языке. Делать это буду банально потому, что большинство технических решений, представленных в библиотеках, о которых мы будем дальше говорить (специализированных) - на английском.

Помимо этих двух ресурсов, где можно искать ответы на вопросы, которые у вас уже возникли, можно проходить какие-нибудь курсы на условном [степике](#)

Но всё это неинтересно. Единственная действительно интересная парадигма изучения это не обкатка абстрактных заданий, а оптимизация каких-то учебных рутин, например, расчёт чего-то в Excel(😡) на языке Python. Вы вроде бы и видите реальное применение, вроде бы и понимаете что и как. Но при этом и учитесь программировать. Все зайцы мертвы. Успех безоговорочный !

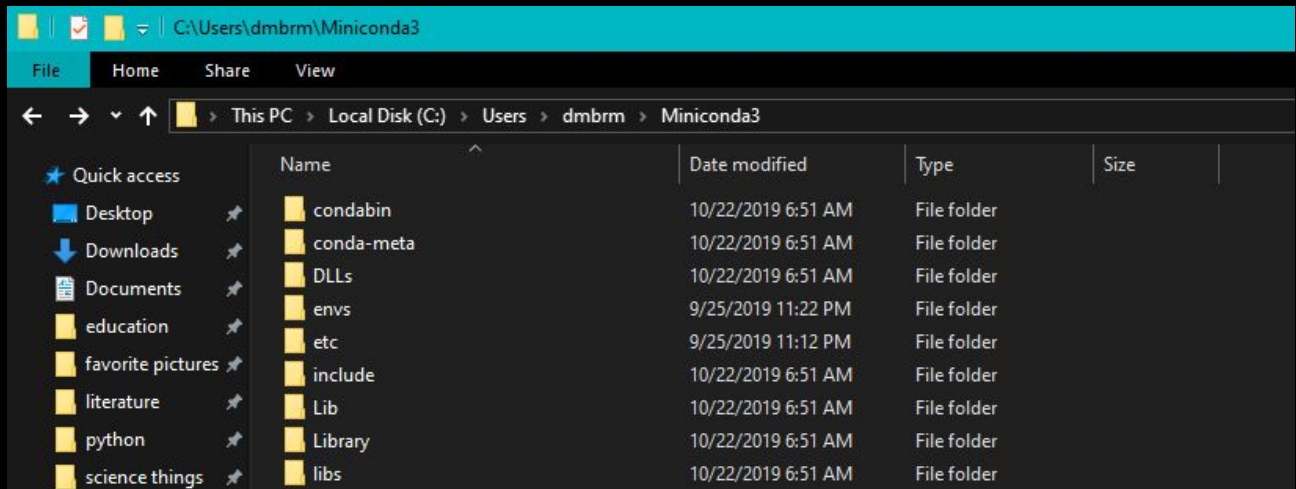
На этом небольшое вступление заканчивается и переходим непосредственно к инструкции, которая должна позволить вам настроить рабочую среду для следующего занятия.

Нам будет нужно:

- Настроенная виртуальная среда
- Jupyter lab
- Numpy, pandas, matplotlib, plotly, seaborn (может ещё что)

Приступим !

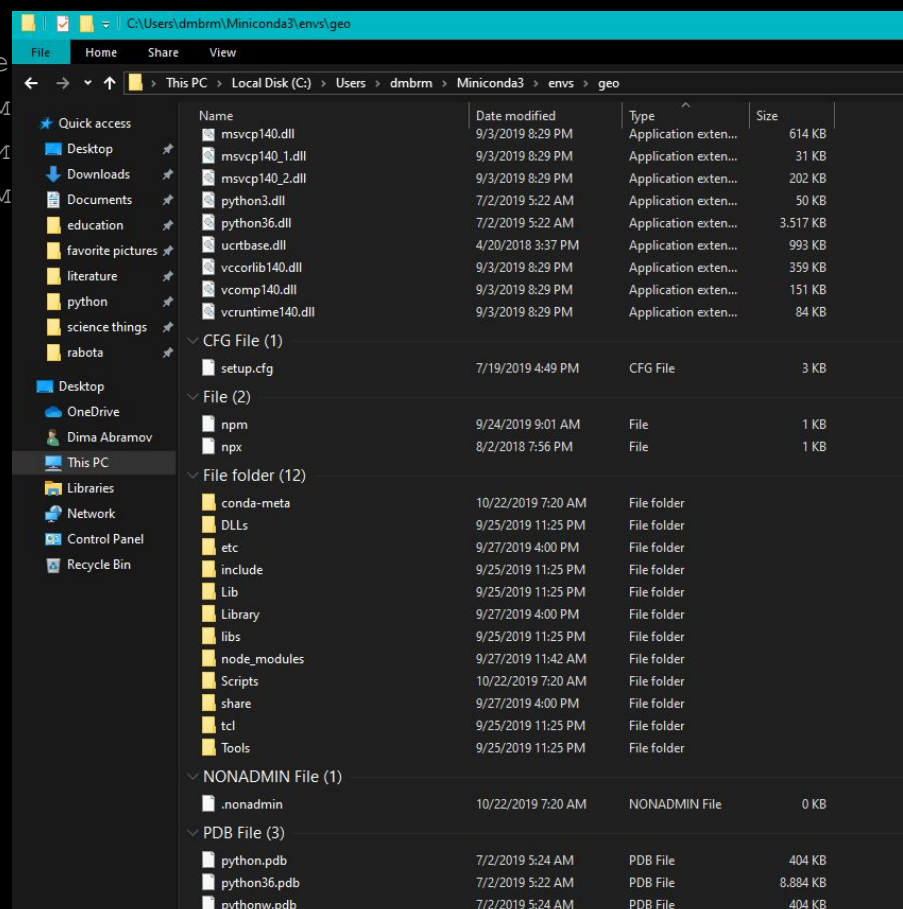
Первоначально давайте создадим для нашей работы опосредованную виртуальную среду. Сама по себе виртуальная среда представляет из себя просто отдельную папку, которую мы заполняем нужными нам инструментами (библиотеками / пакетами)



Внутри папки **envs** находятся все среды, которые мы устанавливаем. Если взглянуть внутрь, то, в принципе, можем понять, что там находится всё тоже самое, что и в корневой папке.

Делается это для того, чтобы пакеты не вступали в конфликт между собой. Некоторые библиотеки требуют определённые дополнительные пакеты для реализации внутри расположенных алгоритмов, которые при нахождении в одном месте ломаются и не работают.

Чтобы такого не произошло – создаём виртуальную среду и насыщаем её нужными нам инструментами.



В предположении, что вы установили Mini Conda сразу перейдём к установке и настройке наших классных штук. По сути своей - это просто консоль Windows, для которой работает надстройка по взаимодействию с Python.

Для начала давайте познакомимся с командой, которая является индикатором хорошего тона. Если всё свеженькое, то, скорее всего, всё работать будет.

Эта команда **conda update**
Любое взаимодействие с консолью **conda** начинается с ключевого слова **conda**. Дальше пишется та или иная команда, ну а после - её аргументы.

conda

(base) C:\Users\dmbrm>conda update --all

conda - conda update --all

```
(base) C:\Users\dmbrm>conda update --all
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: C:\Users\dmbrm\Miniconda3
```

```
The following packages will be downloaded:
```

package	build		
cffi-1.13.1	py37hb32ad35_0	220 KB	conda-forge
Total:		220 KB	

```
The following packages will be UPDATED:
```

```
cffi 1.13.0-py37hb32ad35_0 --> 1.13.1-py37hb32ad35_0
```

```
Proceed ([y]/n)? y
```

В моём случае оказалось не так уж много "устаревших" пакетов. Подтверждаем установку буквой **Y** и наслаждаемся тратой трафика.

В данном разделе мы работали в основной среде. Определить то, где мы находимся можно по слову в скобках, которое находится перед путём расположения папки пользователя. (base) C:\Users\dmbrm>

Чтобы создать новую виртуальную среду нам нужно воспользоваться следующей командой: (base) C:\Users\dmbrm>conda create --name geo
Вместо **geo** может быть использована абсолютно любая переменная имени. Но только на латинице. Кириллицу **Python** не любит. После того, как мы прожали кнопки и среду создали, мы можем в неё "войти". Для этого пишем:

(base) C:\Users\dmbrm>conda activate geo

Как мы можем заметить, префикс перед папкой пользователя сменился. **(geo) C:\Users\dmbrm>**

Отлично ! Давайте теперь установим сюда наш первый пакет/библиотеку/инструмент - **Jupyter Lab**

(geo) C:\Users\dmbrm>conda install jupyterlab Для этого вводим следующую команду:

Важно при этом находиться в той среде, в которой вы будете далее работать. Иначе всё это тщетно и бесполезно. И установите вы не туда. Так что следите за префиксом.

После установки давайте подключим ещё один канал для загрузок библиотек - **Conda Forge**. Делается это весьма тривиально.

Поочерёдно введенные следующие команды помогут нам в этом:

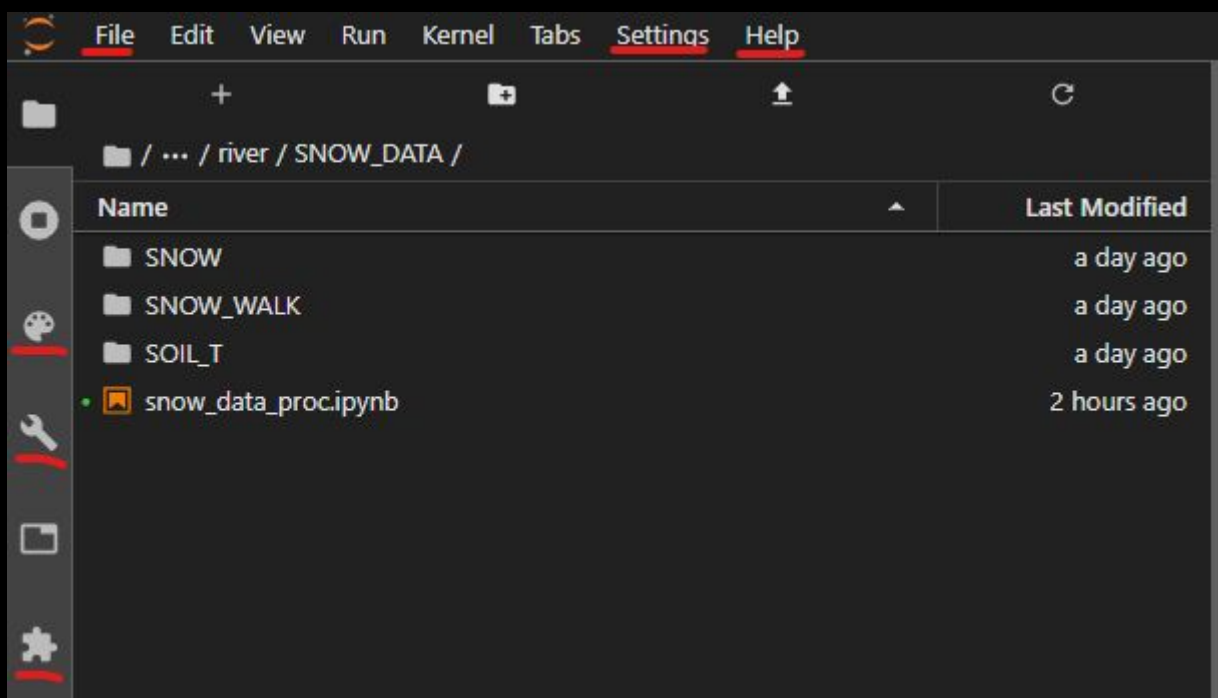
```
(geo) C:\Users\dmbrm>conda config --add channels conda-forge
```

```
(geo) C:\Users\dmbrm>conda config --set channel_priority strict
```

Теперь мы подключили дополнительный канал, по которому можем загружать какие-нибудь пакеты. Сделали мы это банально потому, что некоторые специализированные из них как раз таки только по этому каналу и могут быть получены.

Теперь повторяем нашу первую команду и всё "апдейтим".

После этого запускаем наш **Jupyter Lab**. Пред нами предстает тот инструмент, который будет дальше нам помогать во многом. Писать, смотреть, возможно даже считать. Давайте познакомимся с ним поближе (попытайте на штуки, которые я подчеркнул. Посмотрите настройки. Кастомизируйте среду под себя. Это не сложно и интуитивно понятно. Надеюсь)



Как вы можете заметить, то у вас нету "пазлика". Активировать его можно следующим образом: **Settings -> Enable Extension Manager**

После того, как пазлик появился давайте установим весьма полезное расширение: **Variable Inspector**

Как вы наверное могли заметить, вам вылетела ошибка 😊. Ничего страшного. Читаем что написано и делаем то, что просят. Скорее всего вам не достаёт двух пакетов: **nodejs** и **npm**

Давайте их установим. Сделать можно это двумя способами:

- Открыть дополнительную консоль **conda**, зайти в вашу среду вновь и воспользоваться командой **conda install "имя пакета"**
- Или воспользоваться ячейкой **Jupyter Lab**

```
[*]: !conda install nodejs
```

Ждём, пока звёздочка превратится в циферку. И пробуем вновь. (Аналогично поступите и для другого пакета). Можно устанавливать сразу несколько пакетов, просто записывая их через пробел.

```
[*]: !conda install pandas numpy
```

Неважно где - в notebook или консоли - логика одна. Только отличается за тем, что в консоль не нужно вводить восклицательный знак перед выполнением команды.

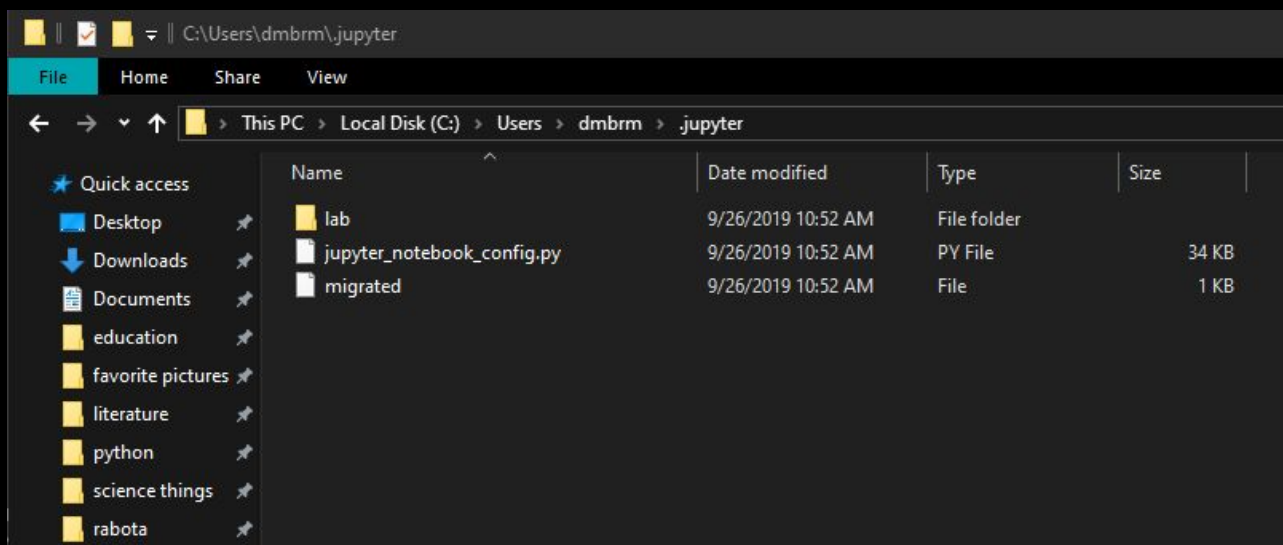
Стоит также обратить внимание, что наша корневая папка, которая открывается в **Jupyter Lab** это папка с нашим пользователем на диске **C**. Не удобно. Наши файлы и документы для работы могут жить в другом месте (лучше так и делать, чтобы организовывать логичные структуры внутри своего компьютера)

Давайте "переедем".

Для этого в консоль вводим следующую команду:

```
(geo) C:\Users\dmbrm>jupyter notebook --generate-config
```

После этого в папке пользователя, расположенной на диске **"C"** появится папка с конфигом (консоль вас об этом оповестит)

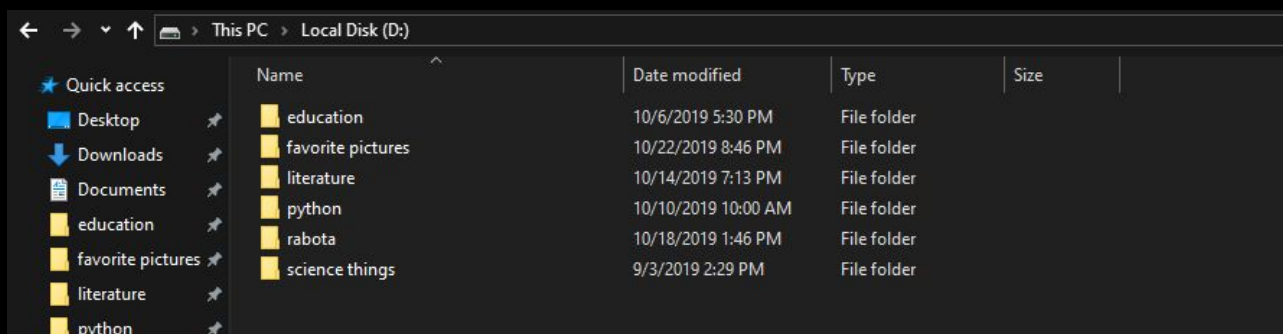
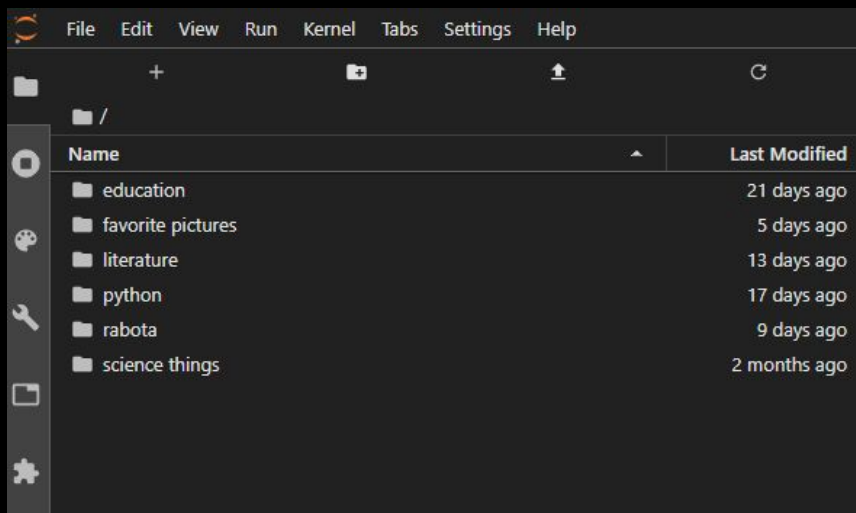


Открываем файл с расширением `".py"`. Воспользуемся для этого блокнотом. С помощью поиска по файлу (**Alt + F**) находим следующую строку:

```
## The directory to use for notebooks and kernels.  
c.NotebookApp.notebook_dir = 'D:/'
```

Убираем символ комментария (`#`). И в кавычках пишем желаемую директорию. Сохраняем файл (**Ctrl + S**). Закрываем файл (**Alt + F4**)

Запускаем вновь **Jupyter Lab** и наслаждаемся нашим мини-успехом. Как видно теперь мы связали папку на компьютере с интерфейсом в браузере



PS: так же я вам рекомендую пробежать по тем файлам **pdf** которые вы можете найти на **GitHub** в папке задания номер 3.

Помимо этого вам нужно установить библиотеки: **pandas matplotlib numpy seaborn glob geopandas osgeo plotly**

Успехов !