**Модуль 1  
 Установка Python, Anaconda, Pycharm**

**(в различных операционных системах)**

<https://www.python.org/downloads/>

Пошаговая инструкция: <https://devpractice.ru/python-lesson-1-install/>

**Установка Python, Anaconda**

<https://lectureswww.readthedocs.io/999.additions/python/install/index.html>

* [Установка Python в ОС Linux](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html)
  + [Сборка из исходников (UNIX)](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#unix)
    - [Скачиваем](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#id1)
    - [Собираем](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#id3)
    - [virtualenv](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#virtualenv)
  + [Linux](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#linux)
    - [Установка интерпретатора CPython](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#cpython)
    - [Пакетный менеджер pip](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#pip)
    - [Виртуальное окружение Virtualenv](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#id4)
    - [Компиляция пакетов](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#id5)
    - [Установка git](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#git)
    - [Пример](https://lectureswww.readthedocs.io/999.additions/python/install/linux.html#id6)
* [Установка Python в ОС MacOS](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html)
  + [Установка интерпретатора CPython](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html#cpython)
  + [Пакетный менеджер pip](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html#pip)
  + [Виртуальное окружение Virtualenv](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html#virtualenv)
  + [Компиляция пакетов](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html#id1)
  + [Установка git](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html#git)
  + [Пример](https://lectureswww.readthedocs.io/999.additions/python/install/macos.html#id2)
* [Установка Python в ОС Windows](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html)
  + [Установка интерпретатора CPython](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html#cpython)
  + [Пакетный менеджер pip](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html#pip)
  + [Виртуальное окружение Virtualenv](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html#virtualenv)
  + [Компиляция пакетов](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html#id1)
  + [Установка git](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html#git)
  + [Пример](https://lectureswww.readthedocs.io/999.additions/python/install/windows.html#id2)
* [Установка Anaconda в Windows](https://lectureswww.readthedocs.io/999.additions/python/install/anaconda.html)
  + [Пакетный менеджер conda](https://lectureswww.readthedocs.io/999.additions/python/install/anaconda.html#conda)
  + [Виртуальное окружение Conda](https://lectureswww.readthedocs.io/999.additions/python/install/anaconda.html#id1)
  + [Пакетный менеджер pip](https://lectureswww.readthedocs.io/999.additions/python/install/anaconda.html#pip)

**Настройка среды Anaconda (**<https://python.ivan-shamaev.ru/guide-conda-environments-anaconda-python-data-science-platform/> **)**

* Создание новой среды в Anaconda Navigator
* Как начать работу в новой среде Conda?
* Настройка среды для Spyder
* Как открыть Jupyter Notebook в новой среде MyNewEnvironmentName
* Установка новой библиотеки (пакета) в среду
* Jupyter Notebook: цифровая лабораторная тетрадь
* Команды Conda
* Управление средами — Managing Environments

 Работа с IPython и Jupyter Notebook:

Для установки Jupyter Notebook необходимо установить его командой:

**python3 -m pip install --upgrade pip python3 -m pip install jupyter**

После установки Jupiter Notebook можно запустить командой:

**jupyter notebook**

Руководство по работе с Jupyter Notebook:

<https://devpractice.ru/python-lesson-6-work-in-jupyter-notebook/>

**Модуль 2**

репозиторий git

Лекции: [Python. Ввод вывод данных.ipynb](http://localhost:8889/notebooks/Softline/softline-course/Python.%20%D0%92%D0%B2%D0%BE%D0%B4%20%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85.ipynb) ,

[Python. Конструкции языка.ipynb](http://localhost:8889/notebooks/Softline/softline-course/Python.%20%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D0%B8%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0.ipynb) ,

[Python. Функции.ipynb](http://localhost:8889/notebooks/Softline/softline-course/Python.%20%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8.ipynb) ,

[Python. Коллекции.ipynb](http://localhost:8889/notebooks/Softline/softline-course/Python.%20%D0%9A%D0%BE%D0%BB%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B8.ipynb)

[Python\_ООП.ipynb](http://localhost:8888/notebooks/Softline/softline-course/Python_%D0%9E%D0%9E%D0%9F.ipynb)

**Методы списков**

|  |  |  |
| --- | --- | --- |
| **Операция** | **Описание** | **Пример** |
| x **in** a | Проверка, что x содержится в а | 5 in [2, 3, 5] |
| x **not in** a | Проверка, что x не содержится в а То же, что и not (x in a) | 5 not in [2, 3, 6] |
| a + a2 | Конкатенация списков, то есть новый список, в котором сначала идут все элементы a, а затем все элементы a2 | [2, 4] + [5, 3] == [2, 4, 5, 3] |
| a \* k | Список a, повторенный k раз | [2, 3] \* 3 == [2, 3, 2, 3, 2, 3] |
| a[n] | n-й элемент списка, отрицательные n — для отсчета с конца | [2, 3, 7][0] == 2 [2, 3, 7][-1] == 7 |
| a[start:stop:step] | Срез списка | [2, 3, 7][:2] == [2, 3] |
| len(a) | Длина списка | len([2, 3, 7]) == 3 |
| max(a) | Максимальный элемент списка | max([2, 3, 7]) == 7 |
| min(a) | Минимальный элемент списка | min([2, 3, 7]) == 2 |
| sum(a) | Сумма элементов списка | sum([2, 3, 7]) == 12 |
| a.index(x) | Индекс первого вхождения x в a (вызовет ошибку, если x not in a, то есть если х отсутствует в а) | [2, 3, 7].index(7) == 2 |
| a.count(x) | Количество вхождений x в a | [2, 7, 3, 7].count(7) == 2 |
| a.append(x) | Добавить x в конец a | a = [2, 3, 7] a.append(8) a == [2, 3, 7, 8] |
| a.extend(a2) | Добавить элементы коллекции a2 в конец a | a = [2, 3, 7] a.extend([8, 4, 5]) a == [2, 3, 7, 8, 4, 5] |
| **del** a[n] | Удалить n-й элемент списка | a = [2, 3, 7] del a[1] a == [2, 7] |
| **del** a[start:stop:step] | Удалить из a все элементы, попавшие в срез | a = [2, 3, 7] del a[:2] a == [7] |
| a.clear() | Удалить из a все элементы (то же, что del a[:]) | a.clear() |
| a.copy() | Копия a (то же, что и полный срез a[:]) | b = a.copy() |
| a += a2 a \*= k | Заменить содержимое списка на a + a2 и a \* k соответственно |  |
| a.insert(n, x) | Вставить x в a на позицию n, подвинув последующую часть дальше | a = [2, 3, 7] a.insert(0, 8) a == [8, 2, 3, 7] |
| a.pop(n) | Получить n-й элемент списка и одновременно удалить его из списка. Вызов метода без аргументов равносилен удалению последнего элемента: a.pop() == a.pop(-1) | a = [2, 3, 7] a.pop(1) == 3 a == [2, 7] |
| a.remove(x) | Удалить первое вхождение x в a, в случае x not in a — ошибка | a = [2, 3, 7] a.remove(3) a == [2, 7] |
| a.reverse() | Изменить порядок элементов в a на обратный (перевернуть список) | a = [2, 3, 7] a.reverse() a == [7, 3, 2] |
| a.sort() | Отсортировать список по возрастанию | a = [3, 2, 7] a.sort() a == [2, 3, 7] |
| a.sort(reverse=**True**) | Отсортировать список по убыванию | a = [3, 2, 7] a.sort(reverse = True) a == [7, 3, 2] |
| bool(a) | Один из способов проверить список на пустоту (возвращает True, если список непустой, и False в противном случае) |  |

**Методы строк**

|  |  |  |
| --- | --- | --- |
| **Операция** | **Описание** | **Пример** |
| s2 **in** s | Проверка, что подстрока s2 содержится в s | 'm' in 'team' |
| s2 **not in** s | Проверка, что подстрока s2 не содержится в s то же, что not (s2 in s) | 'I' not in 'team' |
| s + s2 | Конкатенация (склейка) строк, то есть строка, в которой сначала идут все символы из s, а затем все символы из s2 | 'tea' + 'm' == 'team' |
| s \* k | Строка s, повторенная k раз | 'ha' \* 3 == 'hahaha' |
| s[n] | n-й элемент строки, отрицательные n — для отсчета с конца | 'team'[2] == 'a' 'team'[-1] == 'm' |
| s[start:stop:step] | Срез строки | 'mama'[:2] == 'ma' |
| len(s) | Длина строки | len('abracadabra') == 11 |
| s.find(s2)  s.rfind(s2) | Индекс начала первого или последнего вхождения подстроки s2 в s (вернет -1, если s2 not in s) | s = 'abracadabra' s.find('ab') == 0 s.rfind('ab') == 7 s.find('x') == -1 |
| s.count(s2) | Количество неперекрывающихся вхождений s2 в s | 'abracadabra'.count('a') == 5 |
| s.startswith(s2)  s.endswith(s2) | Проверка, что s начинается с s2 или оканчивается на s2 | 'abracadabra'.startswith('abra') |
| s += s2  s \*= k | Заменить содержимое строки на s + s2 и s \* k соответственно |  |
| s.isdigit()  s.isalpha()  s.isalnum() | Проверка, что в строке s все символы — цифры, буквы (включая кириллические), цифры или буквы соответственно | '100'.isdigit() 'abc'.isalpha() 'E315'.isalnum() |
| s.islower()   s.isupper() | Проверка, что в строке s не встречаются большие буквы, маленькие буквы. Обратите внимание, что для обеих этих функций знаки препинания и цифры дают True | 'hello!'.islower() '123PYTHON'.isupper() |
| s.lower()  s.upper() | Строка s, в которой все буквы (включая кириллические) приведены к верхнему или нижнему регистру, т. е. заменены на строчные (маленькие) или заглавные (большие) | 'Привет!'.lower() == 'привет!' 'Привет!'.upper() == 'ПРИВЕТ!' |
| s.capitalize() | Строка s, в которой первая буква — заглавная | 'привет'.capitalize() == 'Привет' |
| s.lstrip()  s.rstrip()  s.strip() | Строка s, у которой удалены символы пустого пространства (пробелы, табуляции) в начале, в конце или с обеих сторон | ' Привет! '.strip() == 'Привет!' |
| s.ljust(k, c)  s.rjust(k, c) | Добавляет справа или слева нужное количество символов c, чтобы длина s достигла k | 'Привет'.ljust(8, '!') == 'Привет!!' |
| s.join(a) | Склеивает строки из списка a через символ s | '+'.join(['Вася', 'Маша']) == 'Вася+Маша' |
| s.split(s2) | Список всех слов строки s (подстрок, разделенных строками s2) | 'Раз два три!'.split('а') == ['Р', 'з дв', ' три!'] |
| s.replace(s2, s3) | Cтрока s, в которой все неперекрывающиеся вхождения s2 заменены на s3 Есть необязательный третий параметр, с помощью которого можно указать, сколько раз производить замену | 'Раз два три!'.replace('а', 'я') =='Ряз двя три!' 'Раз два три!'.replace('а', 'я', 1) == 'Ряз два три!' |
| list(s) | Список символов из строки строки s | list('Привет') == ['П', 'р', 'и', 'в', 'е', 'т'] |
| bool(s) | Проверка, что строка не пустая (возвращает True, если не пустая, и False в противном случае) |  |
| int(s)  float(s) | Если в строке s записано целое (дробное) число, получить это число, иначе — ошибка | int('25') == 25 |
| str(x) | Представить любой объект x в виде строки | str(25) == '25' |

**Практика:**

Ветвления:

Задачи: Високосный год, Минимум из трех чисел (<https://pythontutor.ru/lessons/ifelse/> )

Цикл For:

Задачи: Сумма факториалов, Лесенка (<https://pythontutor.ru/lessons/for_loop/> )

Строки:

Задачи: Переставить два слова, Замена подстроки (<https://pythontutor.ru/lessons/str/> )

Цикл While:

Задачи: Числа Фибоначчи, Максимальное число идущих подряд равных элементов (<https://pythontutor.ru/lessons/while/> )

Списки:

Задачи: Уникальные элементы, Количество совпадающих пар (<https://pythontutor.ru/lessons/lists/> )

Функции и рекурсия:

Задачи: Возведение в степень, Числа Фибоначчи (<https://pythontutor.ru/lessons/functions/> )

Двумерные массивы:

Задачи: Диагонали, параллельные главной, Поменять столбцы (<https://pythontutor.ru/lessons/2d_arrays/> )

Словари:

Задачи: Частотный анализ, продажи (<https://pythontutor.ru/lessons/dicts/> )

**Модуль 3**

[Numpy для начинающих](https://pythonworld.ru/numpy" \t "_self)

[Серия уроков по Numpy](https://devpractice.ru/category/machine-learning-and-data-analysis/numpy/" \t "_self)

репозиторий git :

Лекции: [Numpy\_1 (создание, slicing).ipynb](http://localhost:8889/notebooks/Softline/softline-course/Numpy_1%20(%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%2C%20slicing).ipynb) , [Numpy\_2 (Математические операции, линейная алгебра).ipynb](http://localhost:8889/notebooks/Softline/softline-course/Numpy_2%20(%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%2C%20%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F%20%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0).ipynb)

Практика, Лабораторная : [Numpy (практика, лабораторная).ipynb](http://localhost:8889/notebooks/Softline/softline-course/Numpy%20(%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B0%2C%20%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F).ipynb)

**Модуль 4**

[Matplotlib уроки](https://devpractice.ru/matplotlib-lesson-4-4-imshow-pcolormesh/)

[Документация](https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html" \t "_self)

**Полезная статья о визуализации -**[**https://m.habr.com/ru/post/468295/**](https://m.habr.com/ru/post/468295/)

[5 простых способов визуализации данных на Python](https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html)

[Тьюториал](https://www.tutorialspoint.com/scipy/index.htm) по разным модулям scipy (см меню слева)

<https://ru.coursera.org/lecture/mathematics-and-python/piervoie-znakomstvo-numpy-scipy-i-matplotlib-gHnLw>  Cursesa МФТИ

[Корреляция в scipy](https://realpython.com/numpy-scipy-pandas-correlation-python/#example-scipy-correlation-calculation)

репозиторий git :

Лекция: [Matplotlib.ipynb](http://localhost:8889/notebooks/Softline/softline-course/Matplotlib.ipynb)

Практика, лабораторная: [Matplotlib (практика, лабораторная).ipynb](http://localhost:8889/notebooks/Softline/softline-course/Matplotlib%20(%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B0%2C%20%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F).ipynb)

**Модуль 5**

[Pandas уроки](https://devpractice.ru/pandas-indexing-part3/)

[Работа с пропусками данных](https://devpractice.ru/pandas-work-with-nan-part4/" \t "_self)

[Анализ данных с помощью Pandas](https://pythonworld.ru/obrabotka-dannyx/pandas-cookbook-1-csv-reading.html)

**О исследовании дата фрейма с pandas -**[**https://habr.com/ru/company/ods/blog/322626/**](https://habr.com/ru/company/ods/blog/322626/)

репозиторий git :

Лекция: [Pandas.ipynb](http://localhost:8889/notebooks/Softline/softline-course/Pandas.ipynb)

Практика, лабораторная:

[Pandas (практика лабораторная).ipynb](http://localhost:8888/notebooks/Softline/softline-course/Pandas%20(%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B0%20%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F).ipynb)

**Модуль 6**

Пример анализа реальных данных: поток велосипедистов через мост Fremont Bridge, Сиэтл, США.

(<https://jakevdp.github.io/blog/2014/06/10/is-seattle-really-seeing-an-uptick-in-cycling/> )

репозиторий git :

Лекции:

[ПОЛИТИКА И ПОЛИЦИЯ. (Стэндфордский университет).ipynb](http://localhost:8888/notebooks/Softline/softline-course/%D0%9F%D0%9E%D0%9B%D0%98%D0%A2%D0%98%D0%9A%D0%90%20%D0%98%20%D0%9F%D0%9E%D0%9B%D0%98%D0%A6%D0%98%D0%AF.%20(%D0%A1%D1%82%D1%8D%D0%BD%D0%B4%D1%84%D0%BE%D1%80%D0%B4%D1%81%D0%BA%D0%B8%D0%B9%20%D1%83%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82).ipynb)

[Поток велосипедистов через мост Fremont Bridge.ipynb](http://localhost:8888/notebooks/Softline/softline-course/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%20%D0%B2%D0%B5%D0%BB%D0%BE%D1%81%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%81%D1%82%D0%BE%D0%B2%20%D1%87%D0%B5%D1%80%D0%B5%D0%B7%20%D0%BC%D0%BE%D1%81%D1%82%20Fremont%20Bridge.ipynb)

Практика, лабораторная:

Загрузите данные по штату Массачусетс, используя <https://stacks.stanford.edu/file/druid:py883nd2578/MA-clean.csv.gz>

Проведите анализ подобно анализу для штата Вермонт, представленному лекции.

**Модуль 7**

Регулярные выражения: Шаблоны, соответствующие одному символу

<https://habr.com/ru/post/349860/>

[Справочное руководство](https://medium.com/nuances-of-programming/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE-%D0%BF%D0%BE-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8E-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8F-%D1%80%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D1%8B%D1%85-%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9-python-774fb2b62218" \t "_self)

Beautiful Soup. Уроки:

<https://habr.com/ru/sandbox/132503/>

<https://code.tutsplus.com/ru/tutorials/scraping-webpages-in-python-with-beautiful-soup-the-basics--cms-28211>

Полная документация по модулю JSON для Python 3 на русском: <https://pythonru.com/moduli/modul-json-python-dlja-raboty-s-formatom-json>

Уроки по работе с JSON

<https://pythonru.com/uroki/modul-json-uroki-po-python-dlja-nachinajushhih>