Лабораторная работа №1

Установка Python, Jupyter Notebook и необходимых библиотек. Знакомство со средой обработки больших данных

Оглавление

[**Теоретические сведения 2**](#_Toc144587757)

[**Обзор программного обеспечения 2**](#_Toc144587758)

[**Дополнительные ссылки по установке: 4**](#_Toc144587759)

[**Ход работы: 5**](#_Toc144587760)

[**Запуск Jupyter Notebook 5**](#_Toc144587761)

[**Переменные и типы данных 9**](#_Toc144587762)

[**Выполнение математических операций и функций 11**](#_Toc144587763)

[**Основные языковые конструкции (операторы Python) 12**](#_Toc144587764)

[**Создание функций 14**](#_Toc144587765)

[**Дополнительные возможности среды Jyputer Notebook 15**](#_Toc144587766)

[**Задание: 17**](#_Toc144587767)

[**Контрольные вопросы: 17**](#_Toc144587768)

**Цель работы: установить и познакомиться со средой работы с большими данными. Освоить основные приемы и команды интерфейс Jupyter Notebook.**

**Результаты**

* Установка Python, Jupyter Notebook
* Запуск Jupyter Notebook
* Написание переменных, функций и циклов на Python
* Выполнение основных манипуляций с данными в Python

# Теоретические сведения

## Обзор программного обеспечения

**Почему Python**

**Python** — это простой язык сценариев, который упрощает взаимодействие с данными. Более того, Python имеет широкий спектр пакетов, которые упрощают начало работы и создание приложений, от самых простых до самых сложных. Python широко используется в промышленности и становится де-факто языком науки о данных в промышленности. (R — еще один альтернативный язык. Однако R имеет тенденцию быть значительно менее масштабируемым и имеет очень мало инструментов развертывания, поэтому он редко используется для промышленного кода в промышленности.)



**Python** — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

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

**IPython** — интерактивная оболочка для языка программирования Python, которая предоставляет расширенную интроспекцию, дополнительный командный синтаксис, подсветку кода и автоматическое дополнение. Является компонентом пакетов программ SciPy и Anaconda.

**Jupyter** **Notebook** — интерактивный блокнот, первоначально являвшийся веб-реализацией и развитием IPython, ставший самостоятельным проектом, ориентированным на работу со множеством сред выполнения — не только Python, но и R, Julia, Scala и ряда других. Создан в 2014 году Фернандо Пересом и Брайном Гренджером.

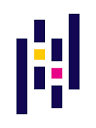
<https://jupyter.org/>

 **Anaconda** — дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения.

<https://www.anaconda.com/download>

** PyCharm** — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик.

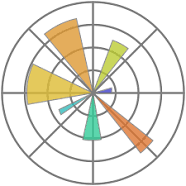
<https://www.jetbrains.com/ru-ru/pycharm/download/?section=windows>

 **pandas** — программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временны́ми рядами.

<https://pandas.pydata.org/>

** NumPy** — библиотека с открытым исходным кодом для языка программирования Python. Возможности: поддержка многомерных массивов; поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

<https://numpy.org/>

**Matplotlib** — библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой. Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях. Matplotlib написан и поддерживался в основном Джоном Хантером и распространяется на условиях BSD-подобной лицензии.

<https://matplotlib.org/>

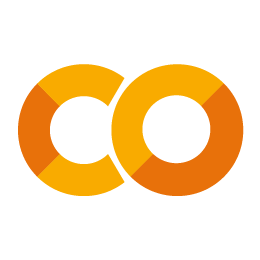
**scikit-learn** — библиотека, предназначенная для машинного обучения, написанная на языке программирования Python и распространяемая в виде свободного программного обеспечения.

<https://scikit-learn.org/stable/>

****

**SciPy** — библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов.

<https://scipy.org/>

**Блокнот Colab** — это бесплатная интерактивная облачная среда для работы с кодом на языке Python от Google в браузере.

<https://colab.research.google.com/>

В нашем курсе мы будет использовать библиотеки Python, которые представлены на рис.1.



*рис. 1 Библиотеки Python*

### Дополнительные ссылки по установке:

**Jupyter Notebook**

<https://www.youtube.com/watch?v=bfCfBuIhlB8>

<https://www.youtube.com/watch?v=HW29067qVWk>

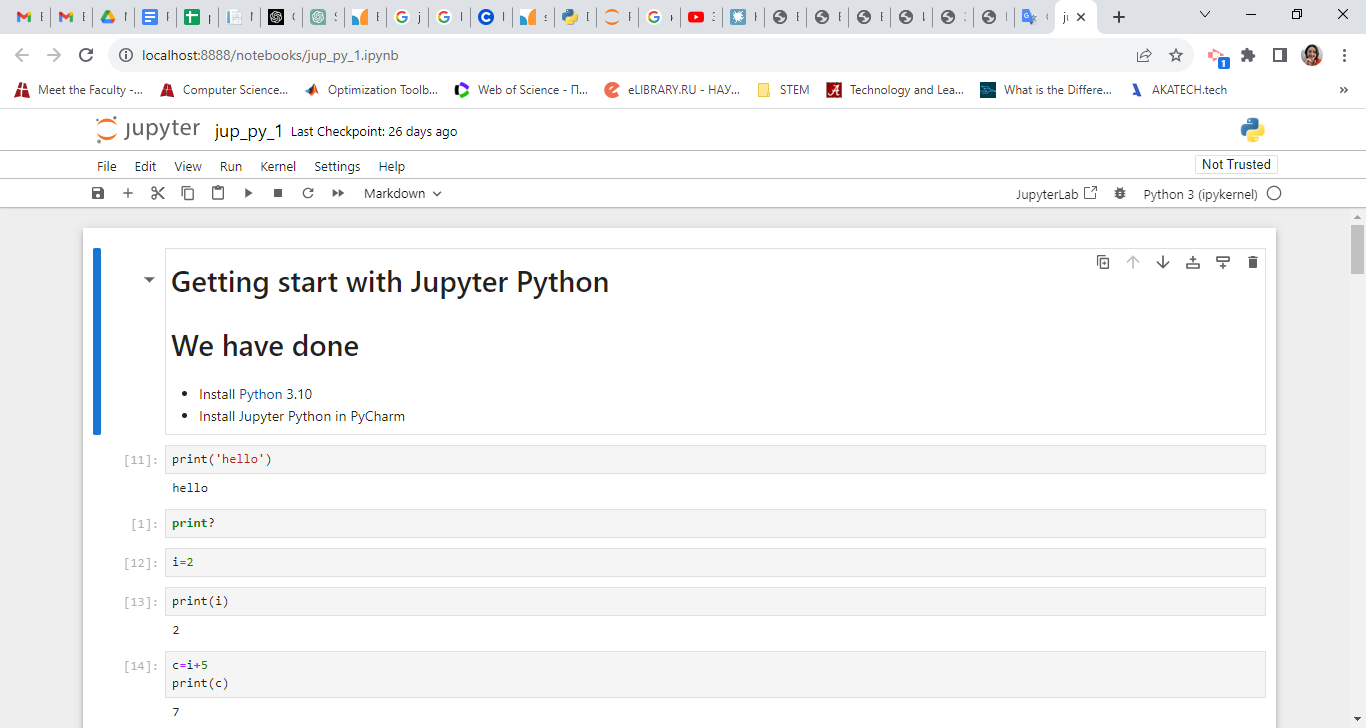
**Установка библиотек в PyCharm**

<https://www.youtube.com/watch?v=BXYsYLYGKEc&t=156s>

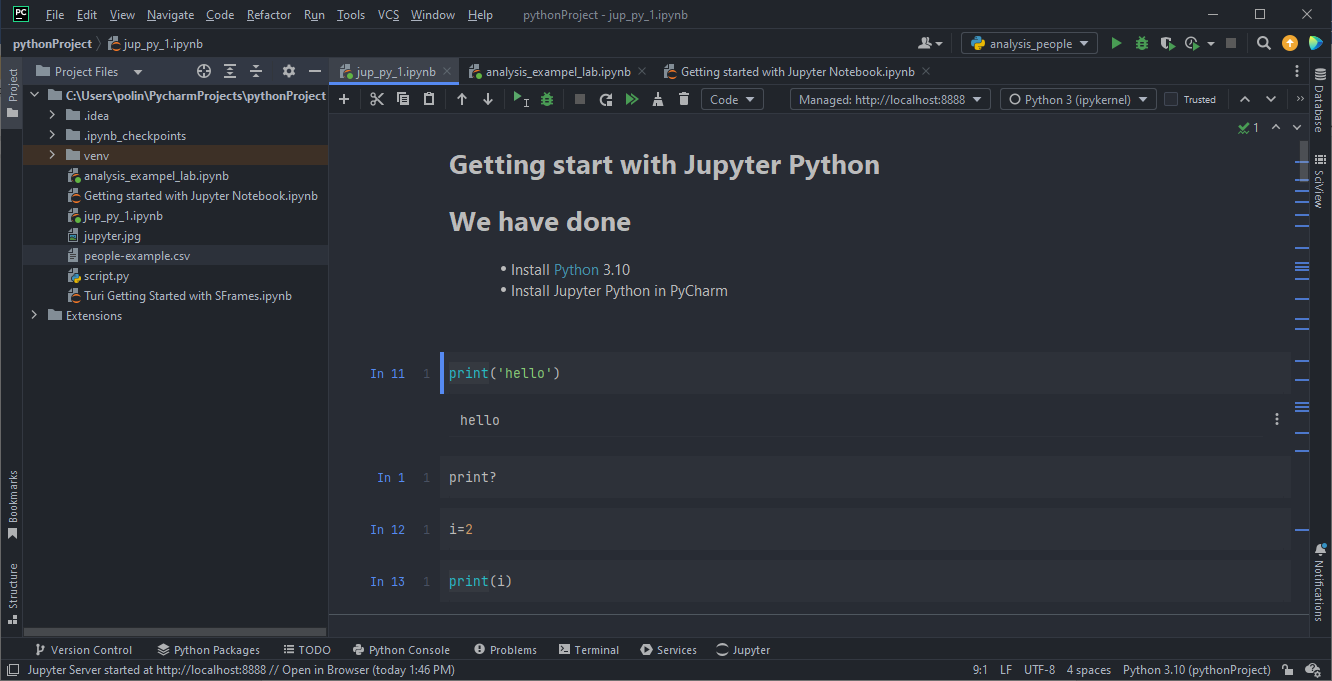
# Ход работы:

## Запуск Jupyter Notebook

После установки программного обеспечения открывается следующий интерфейс в зависимости какую версию вы установили онлайн (browser) или десктоп (PyCharm). См. рис. 2,3

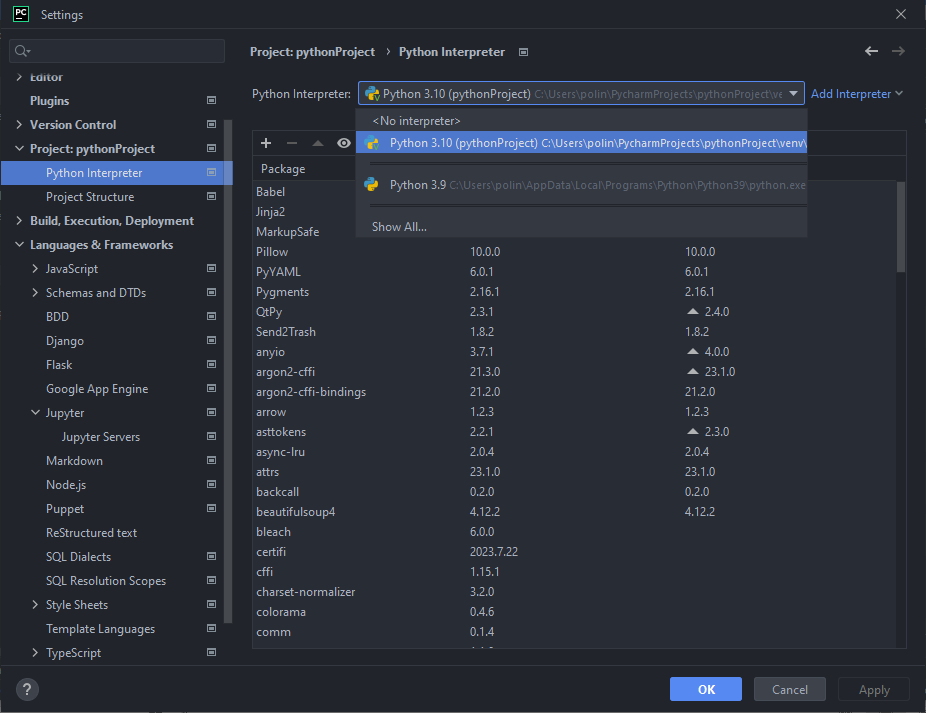


*Рис. 2 Веб-интерфейс Jupyter Notebook*



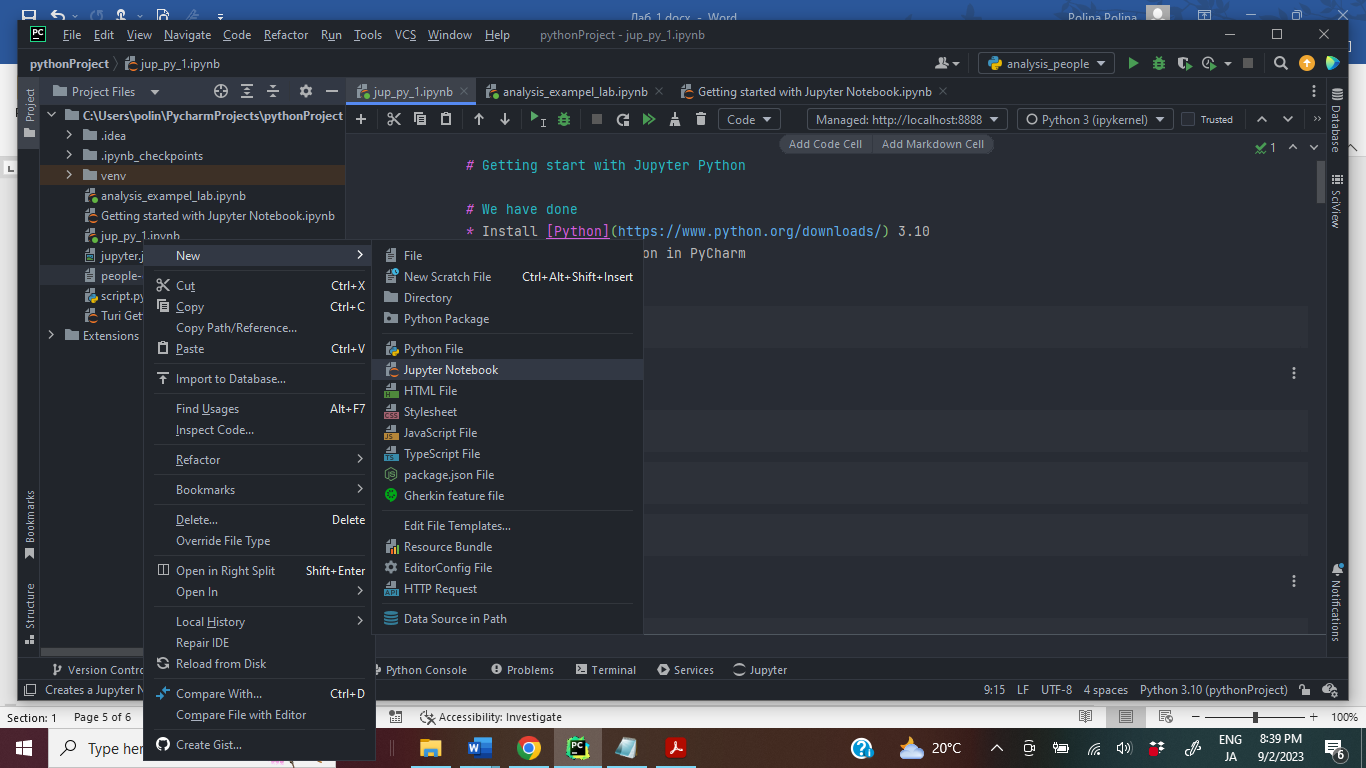
*Рис. 3 Окно IDE PyCharm с установкой проекта Jupyter Notebook*

Но PyCharm имеет команду перехода к просмотру файла в браузере. Также в PyCharm необходимо настроить интерпретатор Python : **File- Settings – Python Interpretator (рис. 4)**



*Рис. 4 Настройка установленной версии интерпретатора Python*

Приступая к работе, необходимо создать файл с расширением **.ipynb,** как показано на рис. 5.



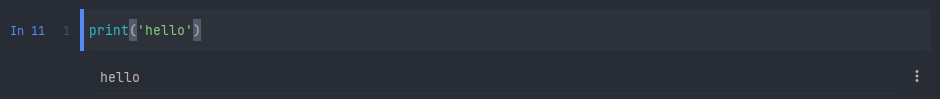
*Рис. 5 Создание нового Notebook (IDE PyCharm)*

**File – New…-Jupyter Notebook**

В среде разработки Notebook исходные данные и компилируемые результаты будут продемонстрированы последовательно в одном **интерактивном файле. Для выполнения команд запускается сервер**

**Jupyter Server started at** [**http://localhost:8888**](http://localhost:8888)

Данные и результаты вставляются в ячейки со сквозной нумерацией **In** и **Out (рис. 6).**

****

*Рис. 6 Ввод и вывод данных в Notebook*

Панель команд, содержит интуитивно понятные инструменты для вставки ячеек, перемещения, запуска на исполнение, вставки типов данных, конфигурации сервера и типа ядра Python (см. рис. 7)



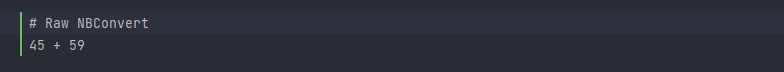
*Рис. 7 Панель инструментов Jupyter Notebook*

В ячейку можно вставить данные типа Code, Markdown и Raw NBConvert.

**Code** – команды Python

**Markdown** – текстовые данные, например, заголовки, текстовые абзацы, гиперссылки и т.д.

**Raw NBConvert** – не исполняемые данные.

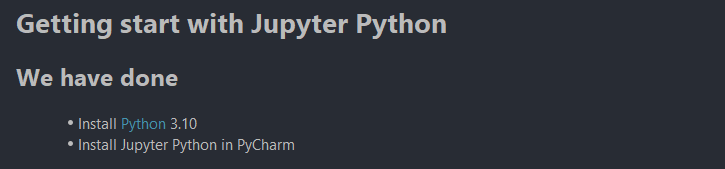


Пример:



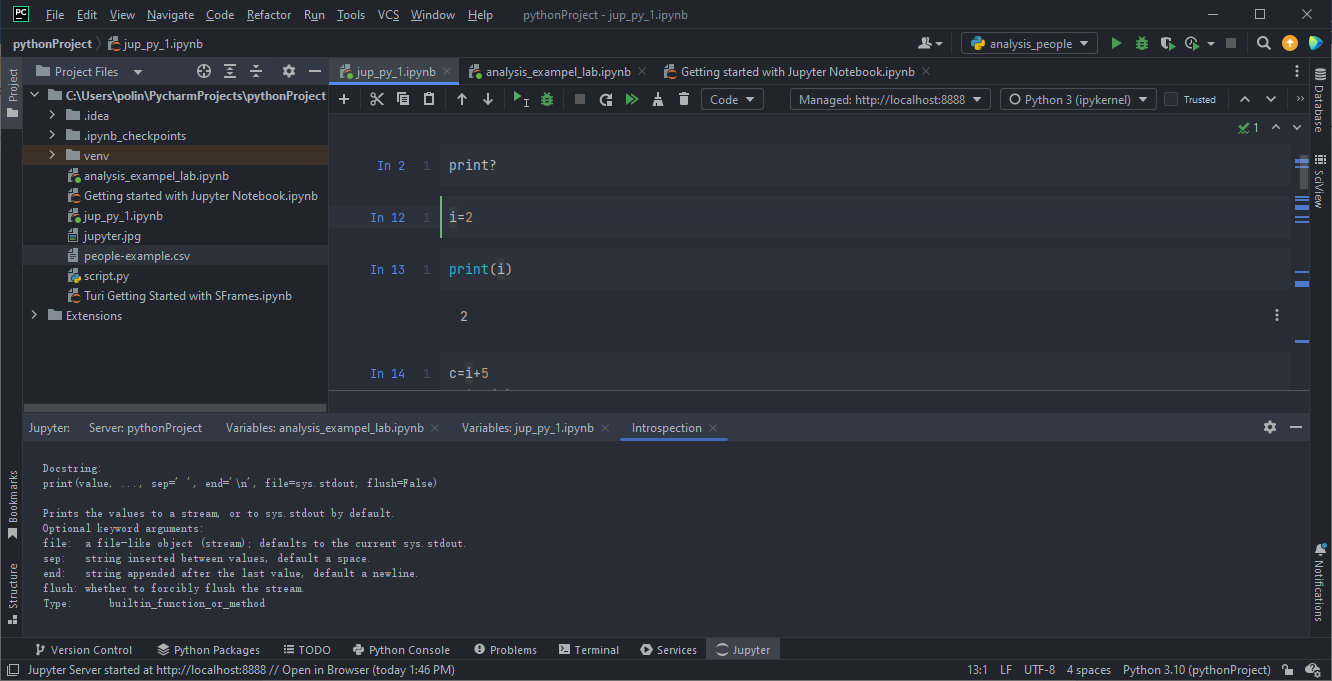
*Рис. 8 Данные типа Markdown*

Результат их исполнения – Заголовок (первого уровня один символ #, второго уровня - ##), список (знак - \*), гиперссылка (Текст указывается в квадратных скобках, URL – в круглых скобках)



Запуск подсказки осуществляется через команду **help()** или вопросительный знак **?**

Пример:

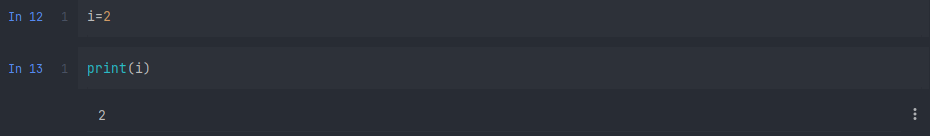


*Рис. 9 Вывод подсказки*

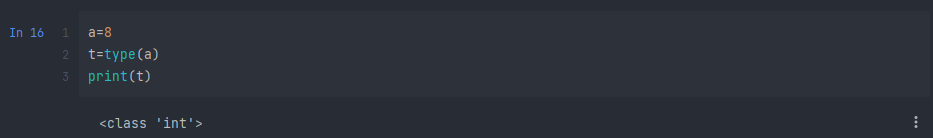
## Переменные и типы данных

Создание переменных происходит в момент присваивания идентификаторам необходимых значений (рис. 10). Тип переменной определяем командой **type(),** см. рис. 11

Пример:



*Рис. 10 Создание переменной i*



*Рис. 11 Определение типа переменной*

Примеры создания переменных в Python (рис. 12, 13)

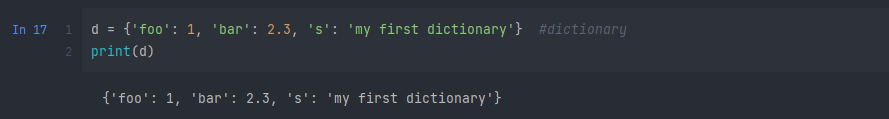


*Рис. 12 Создание переменных в Python*



*Рис. 13 Создание переменных в Python*

Создание типа данных **dictionary**, показано на рис. 14

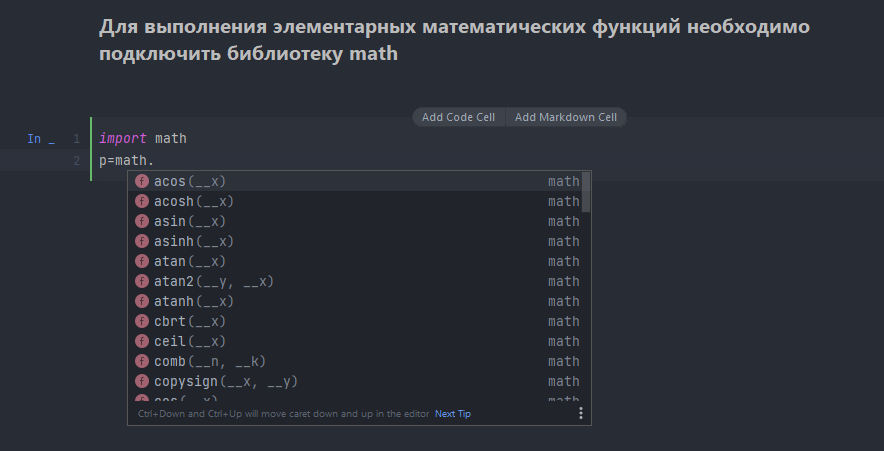


*Рис. 14 Создание переменной типа данных Словарь*

## Выполнение математических операций и функций

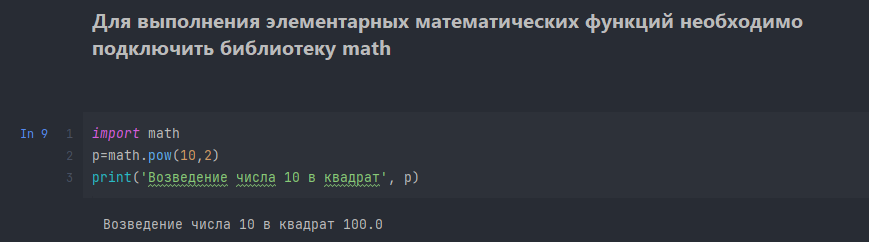
Для выполнения основных арифметических действий работают соответствующие операторы: + - / \*

**Вызов основных математических функций представлен на рис. 15**



*Рис. 15 Подключение библиотеки math*

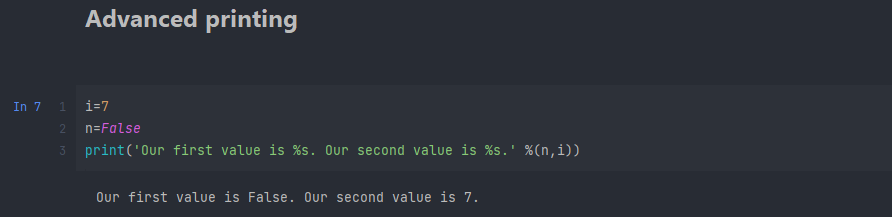
Пример:



*Рис. 16 Выполнение команды «Возведение в степень»*

***Самостоятельно изучите команды класса math. Приведите не менее трех примеров на использование команд для вычисления элементарных математических функций.***

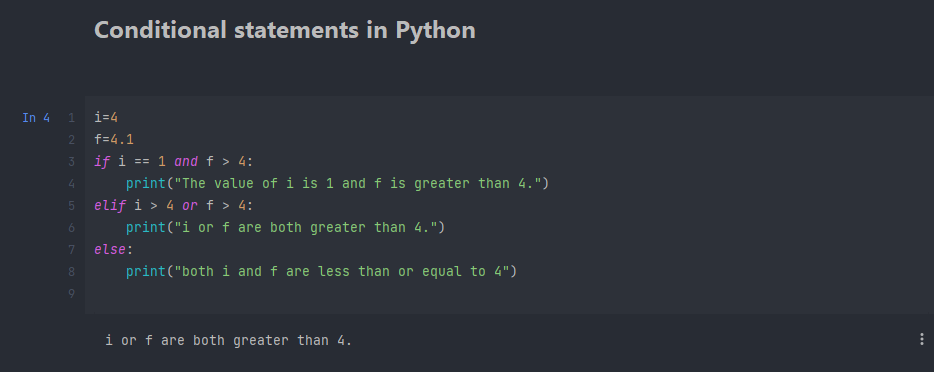
Развёрнутый вывод результатов представлен на рис. 17



*Рис. 17 Применение команды print() с дополнительными аргументами для вывода текста.*

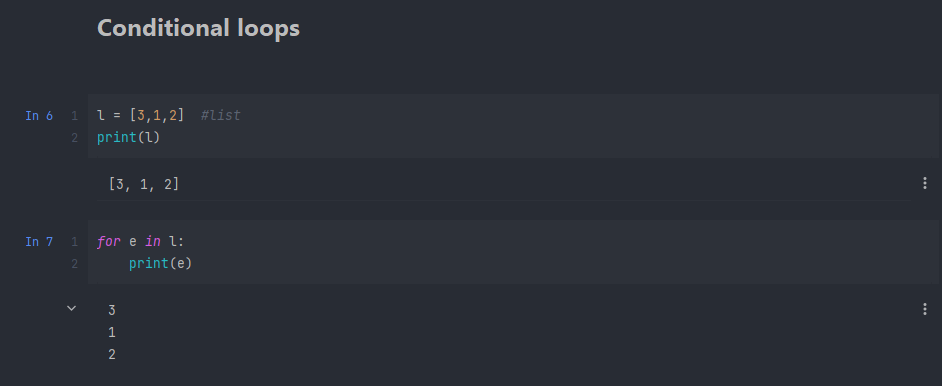
## Основные языковые конструкции (операторы Python)

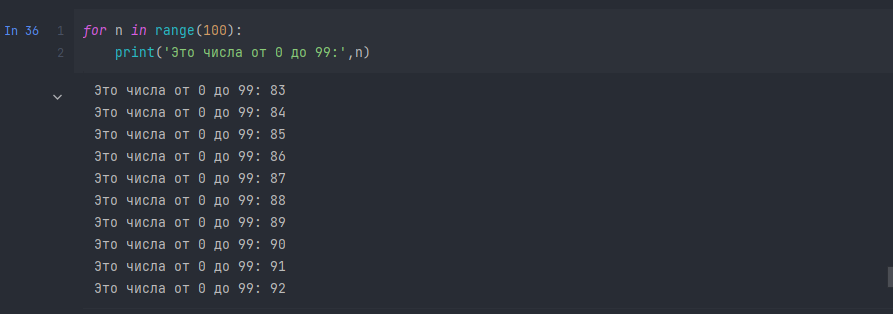
Использование логических операторов **if … elif … else** представлено на рис. 18



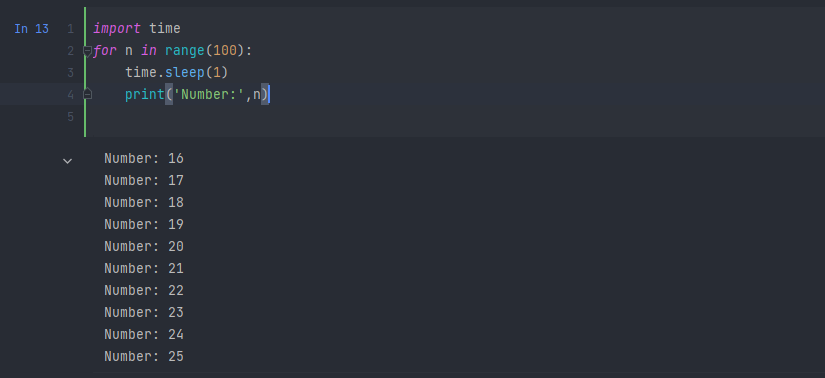
*Рис. 18 Применение условного оператора if …else*

При применении оператора for…in происходит перебор или считывание значений списка в новую переменную, как показано на рис. 19





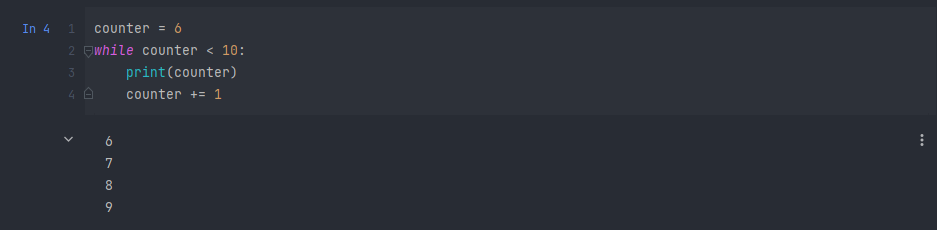
*Рис. 19 Оператор for … in*

**

*Рис. 19а Вывод натуральных чисел с задержкой в 1 сек*

Обратите внимание, что в Python мы не используем {} или другие маркеры для обозначения части цикла, которая повторяется. Вместо этого мы просто делаем отступ и выравниваем каждый повторяемый оператор с помощью пробелов или табуляции. (Вы можете использовать столько, сколько хотите, при условии, что линии выровнены.)

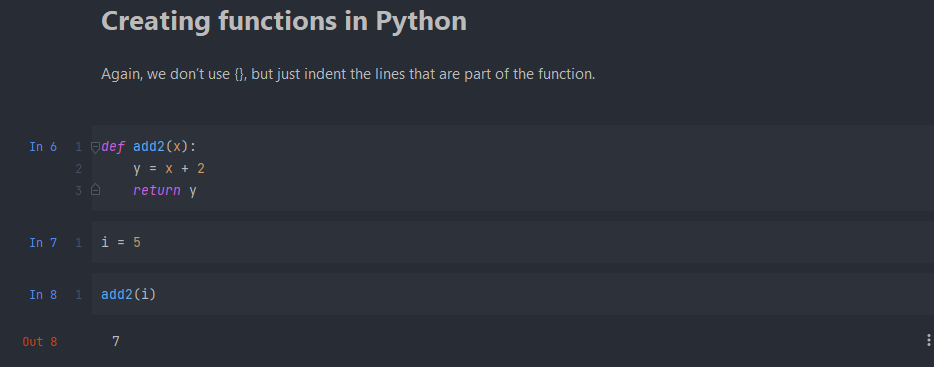
Применение цикла с предусловием **while** представлено на рис. 20



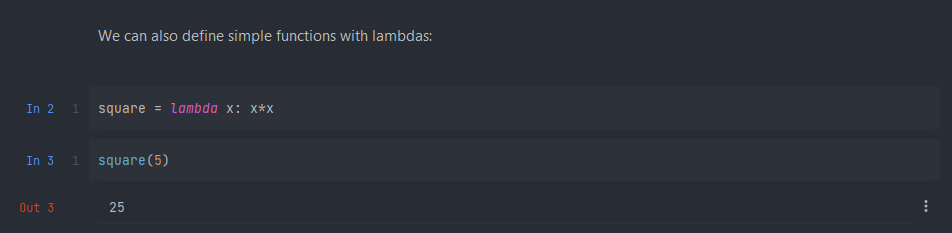
*Рис. 20 Выполнение цикла w**hil*Создание функций

Создание функций в Python, используется ключевое слово **def** . Функция должна возвращать результат **return**

На рис. 21, 22 представлен код создания функции, возвращающей увеличение значения на 2.



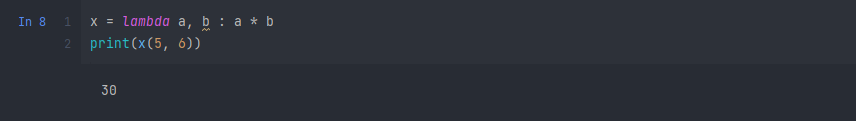
*Рис. 21 Создание функции Python*

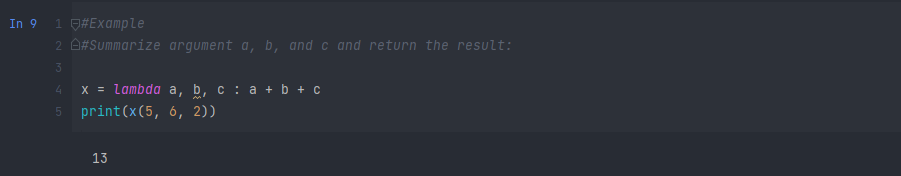


*Рис. 22 Создание функции Python*

Лямбда-функция — это небольшая анонимная функция.

Лямбда-функция может принимать любое количество аргументов, но иметь только одно выражение.





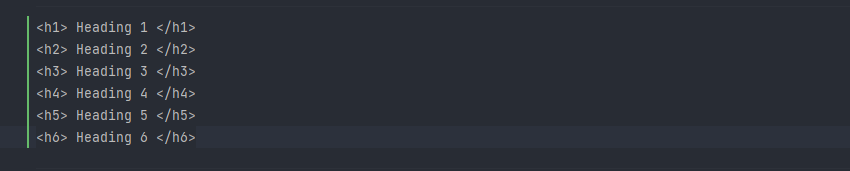
*Рис. 23 Создание анонимной функции в Python*

## Дополнительные возможности среды Jyputer Notebook

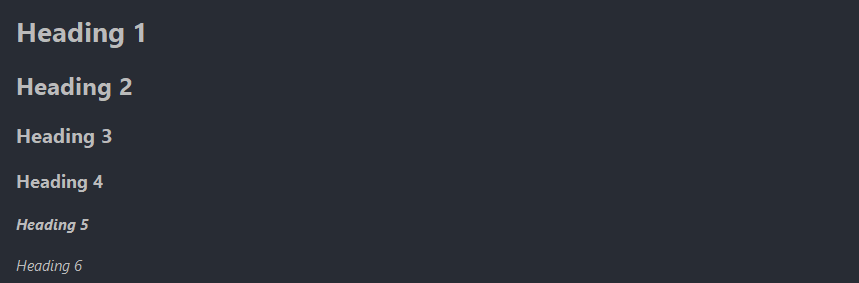
В **Jyputer Notebook** можно внедрять теги языка разметки html и назначать им стили.

Например:

1. Заголовки h1-h6



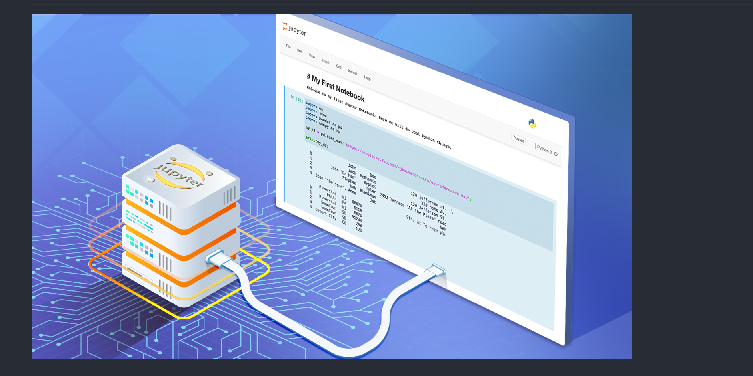
Исполнение



1. Вставка изображений

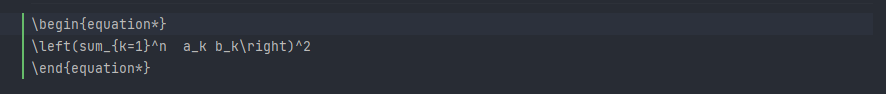


Исполнение

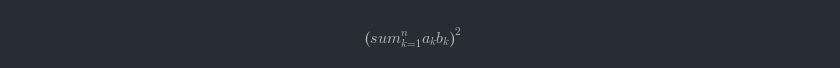


1. Вставка математических формул с помощью встроенного редактора TeX

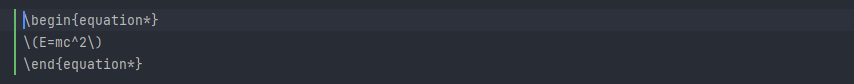
Пример 1



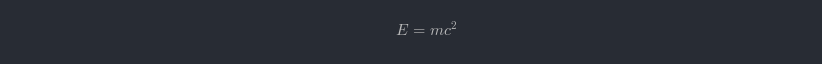
Исполнение



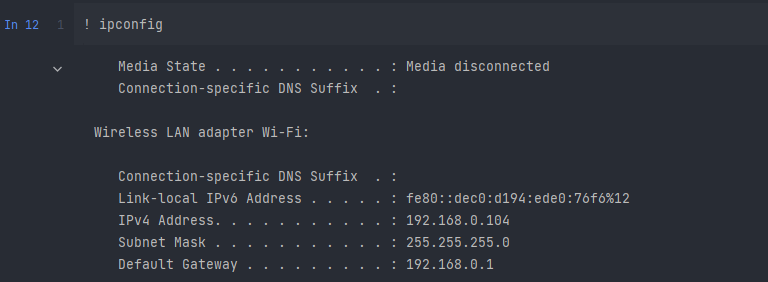
Пример 2

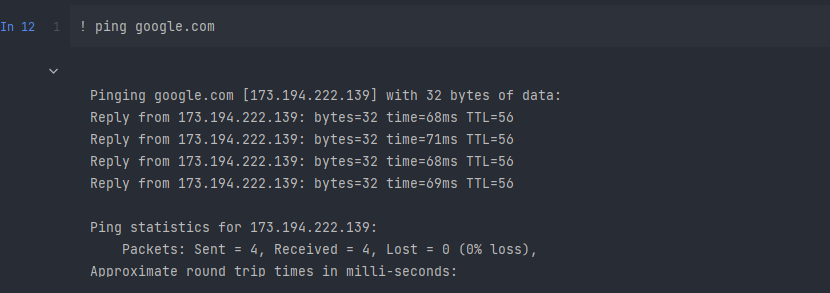


Исполнение (формула Эйнштейна)



1. Выполнение команд консоли





# Задание:

1. Установите Python и Jupyter Notebook
2. Создайте блокнот с именем Lab1\_SName.ipynb (где SName – ваша фамилия)
3. Выполните все примеры, описанные в ходе работы, необходимо использовать свои данные для примеров.
4. Оформите файл Lab1\_SName.ipynb в виде отчета с советующими разделами, комментариями и т.п.

## Контрольные вопросы:

1. Почему для работы с большими данными и для машинного обучения выбирают язык Python? Перечислите не менее трех причин.
2. В чем особенности среды Jupyter Notebook, почему ее выбирают специалисты по Data Science?
3. Перечислите основные библиотеки Python для машинного обучения.
4. Какие типы данных Python вы знаете?
5. Назовите основные языковые конструкции Python.
6. Какое назначение у оператора if … else?
7. Когда заканчивается выполнение оператора while?
8. Какие способы задания функций есть в Python?
9. Что такое Лямбда-функция?
10. Какая библиотека Python отвечает за выполнение основным математических функций?
11. Назовите дополнительные возможности использования среды Jupyter Notebook.