

Погружение в JupyterLab. Знакомство с Python

Цель работы: Приобретение начальных навыков работы в интерактивной среде разработки JupyterLab. Работа с электронным документом блокнотом Jupyter Notebook. Встроенные типы данных Python. Модули math, pyplot (пакет matplotlib), расширение numpy, модуль constants (пакет расширений scipy), модуль sympy языка Python.

Порядок выполнения работы

1. Создание документа и формирование его структуры

Задание 1.1

Ознакомьтесь с интерфейсом среды разработки JupyterLab, научитесь создавать, сохранять, закрывать и открывать рабочие документы.

Выполнение задания 1.1.

JupyterLab – это интерактивная среда разработки для работы с блокнотами, кодом и данными. Проект Jupyter существует для разработки программного обеспечения с открытым исходным кодом, открытых стандартов и сервисов для интерактивных и воспроизводимых вычислений.

- 1) Запустите JupyterLab с помощью ярлыка Anaconda Navigator, меню Пуск. Альтернативным способом запуска JupyterLab является запуск ярлыка Anaconda Prompt, меню Пуск и дальнейшее выполнение команды `python -m jupyterlab` в командной строке. Изучите представленную панель управления блокнотами Jupyter.
- 2) Домашней папкой JupyterLab является `C:/Users/Username`. Переместитесь из домашней папки в папку Documents и создайте в ней новую папку, например, km2 для сохранения Ваших рабочих документов. Сделайте созданную папку текущей.
- 3) Создайте новый документ: используйте кнопку File|New|Notebook и выберите Python [conda env:base]*. Система создаст файл с именем Untitledxx.ipynb и разместит его в текущей папке.
- 4) Ознакомьтесь с пунктами меню среды разработки JupyterLab и командами, представленными в этих пунктах.
- 5) Переименуйте созданный документ. Один из способов – выберите команду File|Close and Shutdown Notebook. Далее укажите правой

кнопкой мыши блокнот, который был закрыт, и в появившемся контекстном меню выберите **Rename**. Именуйте документ следующим образом (латинская раскладка клавиатуры): `KM2_Lb01_Surname.ipynb`.

Как только Вы переименовали рабочий документ, откройте его снова. Открыть документ можно, дважды щелкнув левой кнопкой мыши по его имени. Еще один способ переименования документа – открыть контекстное меню документа, кликнув правой кнопкой мыши по его закладке, и указать команду **Rename Notebook...**

6) Теперь, когда мы начали работать с проектом, лучше регулярно сохранять документ. Нажатие **Ctrl + S** сохранит Ваш документ.

Следует отметить, что каждый раз, когда создается новый блокнот, создается также файл контрольной точки этого блокнота. Он располагается в скрытом подкаталоге места сохранения блокнота с именем `.ipynb_checkpoints`, и также является файлом `.ipynb`. По умолчанию JupyterLab периодически автоматически сохраняет блокнот в этот файл контрольной точки, не изменяя основной файл блокнота. Таким образом, контрольная точка позволяет вам восстановить несохраненную работу в случае непредвиденной проблемы. Вы можете вернуться к контрольной точке из меню через **File | Revert to Checkpoint**.

Задание 1.2

Ознакомьтесь с типами ячеек блокнота Jupyter Notebook и режимами работы с ними. Для отработки навыков создайте структуру документа, аналогичную структуре документа `2025_KM2_Lb1.pdf`.

Выполнение задания 1.2.

Ячейки являются основными объектами блокнота Jupyter Notebook. Различают два типа ячеек:

- ячейка кода (**Code type**) – ее содержимое обрабатывается ядром и результат вычисления отображается ниже ячейки;
- текстовая ячейка (**Markdown type**) – ее содержимое форматируется посредством языка разметки текста **Markdown**, и результат отображается в этой же ячейке.

Различают два режима для работы с ячейками:

- командный режим (**Command mode**) – для вычисления ячейки;
- режим редактирования (**Edit mode**) – для ввода и редактирования данных.

Любая вновь созданная ячейка имеет тип **Code**, режим **Edit**. Первая ячейка создается по умолчанию, при создании нового документа. Последующие ячейки можно создавать выше, ниже существующих или между ними. Для этого используют пункты меню **Edit | Paste Cells (Bellow/Above)** или горячие клавиши, представленные в табл.1.1.

В каждый момент времени документ имеет активную ячейку. Рядом с активной ячейкой отображается синяя вертикальная черта. Сделать другую

ячейку активной можно или щелчком левой кнопки мыши, или курсорными клавишами вверх/вниз. Активными также можно сделать несколько расположенных рядом ячеек, см. табл.1.1.

Таблица 1.1
Работа с ячейками в командном режиме

Действия с ячейками	Горячие клавиши
Отменить последнее действие с ячейками	Z
Создать новую ячейку выше активной	B
Создать новую ячейку ниже активной	A
Удалить активную ячейку	D + D
Копировать активную ячейку в буфер обмена	C
Вставить скопированную ячейку из буфера	V
Вырезать активную ячейку	X
Установить для активной ячейки тип Code	Y
Установить для активной ячейки тип Markdown	M
Изменить режим редактирования на командный режим	Shift + Enter
Изменить командный режим на режим редактирования	двойной щелчок мыши
Сделать активными несколько подряд стоящих ячеек	Shift + вверх/вниз
Объединить активные ячейки	Shift + M
Разделить активную ячейку по курсору	Ctrl +Shift -

- 1) Создайте в Вашем документе текстовые ячейки, введите в каждую текст, представленный в правом столбце табл. 1.2 (тип Markdown, режим редактирования Edit)

Таблица 1.2
Структура документа

Номер ячейки	Содержимое текстовой ячейки
1	Лабораторная работа 1 Задача о падении тела. Построение графиков функций
2	ФИО исполнителя, дата
3	Задание 1.1. Задача о падении тела
4	Задание 1.2. Определение момента времени касания телом земли
5	Задание 1.3. Построение секущей, касательной и нормали к графику функции
6	Задание 1.4. Построение графиков функций

- 2) Измените режим редактирования созданных ячеек на командный режим, используйте горячие клавиши, см. табл. 1.1.

Задание 1.3

Освойте форматирование простого текста посредством языка разметки Markdown.

Для этого каждую из шести созданных ранее текстовых ячеек документа приведите к виду, который имеет соответствующая ячейка в документе 2025_KM2_Lb1.pdf.

Выполнение задания 1.3.

Markdown – язык разметки для форматирования простого текста, его основные управляющие символы представлены в табл. 1.3.

Чтобы узнать больше, обратитесь к официальному руководству от создателя Markdown, Джона Грубера, на его веб-сайте.

<https://daringfireball.net/projects/markdown/syntax>

Таблица 1.3
Разметка для форматирования текста

Тип форматирования	Содержимое текстовой ячейки
Создать заголовок первого уровня	символ решетки # перед текстом заголовка
Создать заголовок второго уровня	два символа решетки ## перед текстом заголовка
Выделить текст курсивом	звездочка * перед и после выделяемого текста
Выделить текст жирным шрифтом	две звездочки ** перед и после выделяемого текста
Создать нумерованный список	символ тире с пробелом – , или плюс с пробелом + , или звездочка с пробелом *

Задание 1.4

Ознакомьтесь с основами языка разметки для форматирования математического текста *Latex*.

Для этого введите в новые ячейки тексты заданий 1.1 – 1.4 из документа 2025_KM2_Lb1.pdf, располагая новые ячейки в соответствующих местах.

Выполнение задания 1.4.

Выполните самостоятельно, принимая во внимание представленную ниже разметку текста из первых абзацев задания 1.1 из документа 2025_KM2_Lb1.pdf.

****Описание процесса**.**

Тело запущено под углом α к горизонту на высоте h_{start} и через время $t=T$ достигает другое тело, расположенное на расстоянии s_{end} и высоте h_{end} .

****Предположения**.**

- Тело будем считать материальной точкой массы m .
- Движение запущенного тела осуществляется только под действием силы тяжести $\mathbf{F} = m(0, -g)$, где m -- масса тела, g -- ускорение свободного падения.
- Соппротивлением среды пренебрегаем.

2. Задача о падении тела

Выполните задание 1.1, представленное в документе 2025_KM2_Lb1.pdf. Рекомендации по его реализации смотрите там же. Выполнение сопровождайте лаконичными комментариями.

3. Время нахождения тела в воздухе

Выполните задание 1.2, представленное в документе 2025_KM2_Lb1.pdf. Рекомендации по реализации задания 1.2 смотрите там же. Выполнение сопровождайте лаконичными комментариями.

4. Секущая, касательная и нормаль к графику функции

Выполните задание 1.3, представленное в документе 2025_KM2_Lb1.pdf. Рекомендации по реализации задания 1.3 смотрите там же. Выполнение сопровождайте лаконичными комментариями.

5. Графики функций в декартовой и полярной системах координат

Выполните Ваш вариант задания 1.4, представленного в документе 2025_KM2_Lb1.pdf. Рекомендации по реализации задания 1.5 смотрите там же. Выполнение сопровождайте лаконичными комментариями.

6. Контрольные вопросы

ЛБ01 Часть 1. Введение в язык программирования Python

1. Расскажите, кто является основателем языка Python, когда и где он начал разрабатываться, каковы его позиции в рейтинге использования языков программирования на сегодняшний день.
2. Перечислите не менее восьми преимущественных качеств языка Python.
3. Объясните фразу «Python поддерживает несколько парадигм программирования», кратко характеризуйте каждую парадигму.
4. Объясните фразу «Python является интерпретируемым языком программирования».
5. Сформулируйте основной недостаток языка Python.
6. Укажите области применения языка Python. Перечислите реальные, наиболее известные продукты, которые созданы с преимущественным использованием языка Python.
7. Опишите общую структуру программы на языке Python.
8. Расскажите об особенностях синтаксиса языка Python.
9. Что значит аббревиатура PEP 8 и какие основные соглашения прописаны в этом документе?
10. Объясните фразу «JupyterLab – свободное ПО с открытым кодом из проекта Jupyter».

ЛБ01 Часть 2. Работа с ячейками в документе .ipynb

1. Дайте определения двум сущностям документа .ipynb приложения Jupyter Notebook: ячейке кода и ячейке редактирования. Используйте шаблон следующего формата: «Ячейка XXX – это сущность документа .ipynb, которая содержит XXX, предназначена для XXX, результат выполнения ячейки отображается XXX».
2. Перечислите действия с ячейками, которые можно выполнять в командном режиме работы Jupyter Notebook.

3. Укажите два языка разметки, которые используют для форматирования текста в документе `.ipynb` приложения Jupyter Notebook, сформулируйте назначение каждого.

А) Ячейки документа. Укажите горячие клавиши для следующих действий:

1. сделать ячейку документа активной;
2. создать ячейку выше активной;
3. создать ячейку ниже активной;
4. установить для активной ячейки тип Code;
5. установить для активной ячейки тип Markdown;
6. активную ячейку перевести в режим редактирования;
7. активную ячейку перевести в командный режим;
8. копировать активную ячейку в буфер обмена;
9. вставить скопированную ячейку из буфера;
10. вырезать активную ячейку;
11. удалить активную ячейку;
12. сделать активными несколько подряд стоящих ячеек;
13. объединить активные ячейки;
14. разделить активную ячейку по курсору;
15. отменить последнее действие с ячейками.

Б) Элементы языка разметки для форматирования простого текста (Markdown). Укажите, какие управляющие символы и каким образом используют для

1. создания заголовка первого уровня;
2. создания заголовка второго уровня;
3. создания заголовка уровня n ;
4. создания нумерованного списка;
5. выделения текста курсивом;
6. выделения текста жирным шрифтом;
7. создания математического выражения без перехода на новую строку;
8. создания математического выражения с переходом на новую строку;
9. отображения последовательности символом указанным цветом.

ЛБ01 Часть 3. Использование некоторых модулей и расширений

1. Укажите имя модуля, который содержит математические функции, и напишите, каким образом его следует подключить.
2. Приведите пример использования некоторой элементарной математической функции, начиная с подключения необходимого модуля.
3. Напишите, как подключить модуль для работы с графикой и как создать при этом псевдоним имени модуля.
4. Укажите имя расширения, которое предназначено для работы со структурой данных массив.
5. Укажите имя модуля, в котором содержатся значения физических констант, и имя пакета, в котором находится этот модуль.

6. Напишите, как определить величину ускорения свободного падения с использованием модуля, который содержит значения физических констант. Начать с подключения этого модуля, указывая соответствующий пакет, его содержащий.
7. Укажите назначение модуля `sympy`.
8. Напишите пример решения некоторого алгебраического уравнения с одним неизвестным, используя при этом функции модуля `sympy`.

ЛБ01 Часть 4. Использование функций графики

1. Напишите спецификацию функции `range`.
2. Напишите спецификацию функции `numpy.arange` для создания одномерного массива.
3. Приведите пример построения графика функции в прямоугольной декартовой системе координат по заданным спискам абсцисс `x_list` и соответствующих ординат `y_list` абсцисс точек плоскости. Используйте функции для установки пределов по осям, для именования осей, для задания заголовка графика.
4. Приведите пример построения графика функции в полярной системе координат, создавая предварительно массивы значений полярных углов поворота и соответствующих полярных радиусов точек плоскости. Используйте функции для установки пределов по осям, для именования осей, для задания заголовка графика.

7. Литература

1. *Лутц М. Изучаем Python, том 1* (5-е изд.) — СПб.: ООО "Диалектика", 2019.
2. Matplotlib Reference <https://matplotlib.org/>
3. *Rougier N., Scientific Visualization: Python + Matplotlib*, 2021.
<https://github.com/rougier/scientific-visualization-book>
4. *Devpractice Team. Библиотека Matplotlib.* — devpractice.ru, 2019.