

*School of
Computer
Science*

Работа с Jupyter Notebook

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы



ЧТО ТАКОЕ JUPYTER NOTEBOOK

Jupyter Notebook – интерактивная оболочка для языка программирования Python



Основные компоненты:

- **Веб-приложение**, запускаемое в браузере
- **Блокноты** – файлы **.ipynb**, объединяющие код и его вывод в единый документ, который объединяет визуализацию, повествовательный текст, математические уравнения и другие мультимедиа

Jupyter используется для интерактивных вычислений, анализа данных, статистического моделирования и машинного обучения. Студенты могут использовать его для оформления отчётов о лабораторных работах



ОСНОВНЫЕ ВОЗМОЖНОСТИ

- Редактирование Python-кода в браузере, с подсветкой синтаксиса, автоотступами и автодополнением
- Запуск кода в браузере
- Отображение результатов вычислений с медиа представлением (схемы, графики)
- Работа с языками разметки Markdown и LaTeX
- Экспорт блокнотов в PDF и HTML





Первый вариант (для data-сайентистов):

Установить дистрибутив **Anaconda**, включающий популярные библиотеки для работы с данными (NumPy, Pandas и Matplotlib), с официального сайта:

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

Второй вариант:

использовать *pip* для установки Jupyter Notebooks:

```
pip install jupyter
```

В последнем случае все необходимые дополнительные пакеты Python нужно будет установить также вручную



ИСПОЛЬЗОВАНИЕ WEB-ВЕРСИЙ

- Можно работать с Jupyter без установки дополнительного софта на своем компьютере
- Смотрите по этим ссылкам:

1. <https://jupyter.org/try>

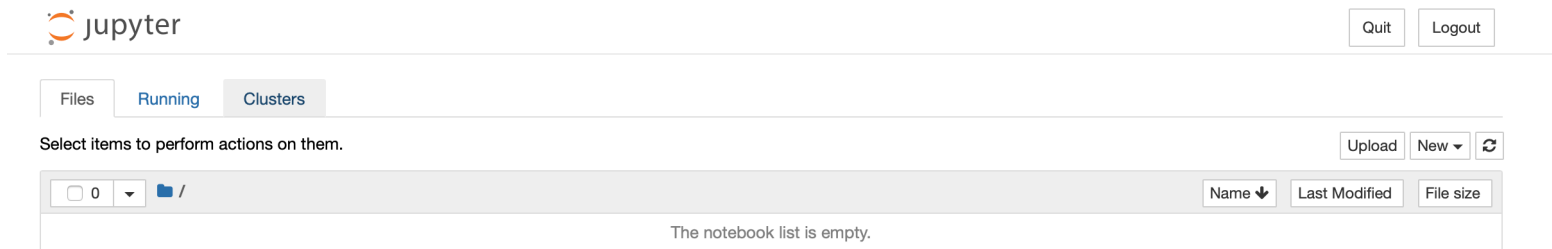
2. <https://colab.research.google.com/>



Если Jupyter был установлен вместе с Anaconda, то он запускается с помощью ярлыка, который Anaconda добавляет в ваше меню «Пуск»

Если для установки использовался менеджер пакетов `pip3`, то Jupyter запускается из командной строки командой `jupyter notebook`

После запуска откроется новая вкладка в вашем браузере по умолчанию:

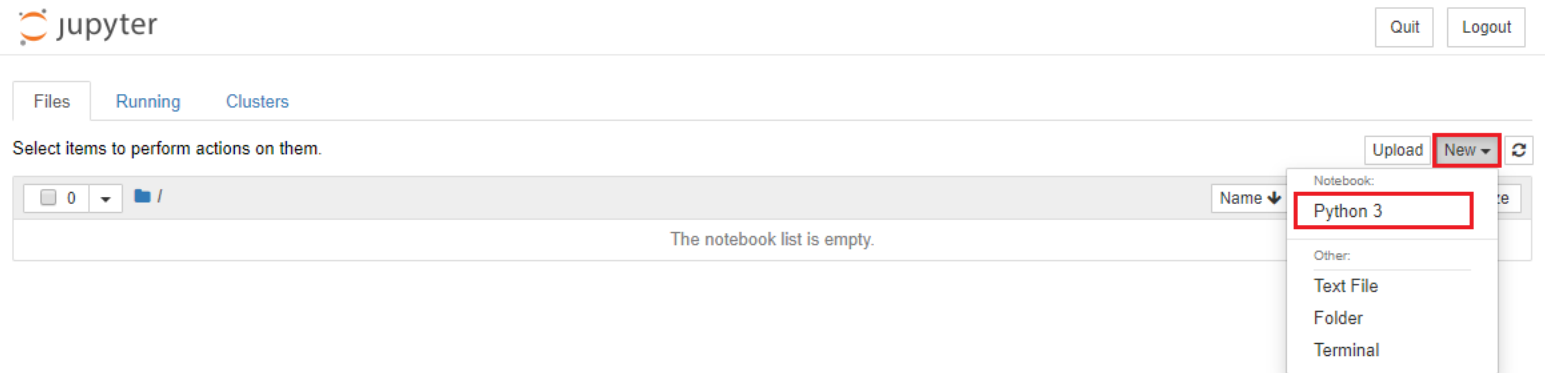


Рабочий каталог может уже содержать файлы и директории. В данном случае он пустой



СОЗДАНИЕ БЛОКНОТА

Чтобы создать новый файл .ipynb, на главной странице Jupyter Notebook в выпадающем меню New выберите Python 3:



Чтобы загрузить новый файл, на главной странице Jupyter Notebook нажмите Upload и выберите файлы для загрузки на своём компьютере



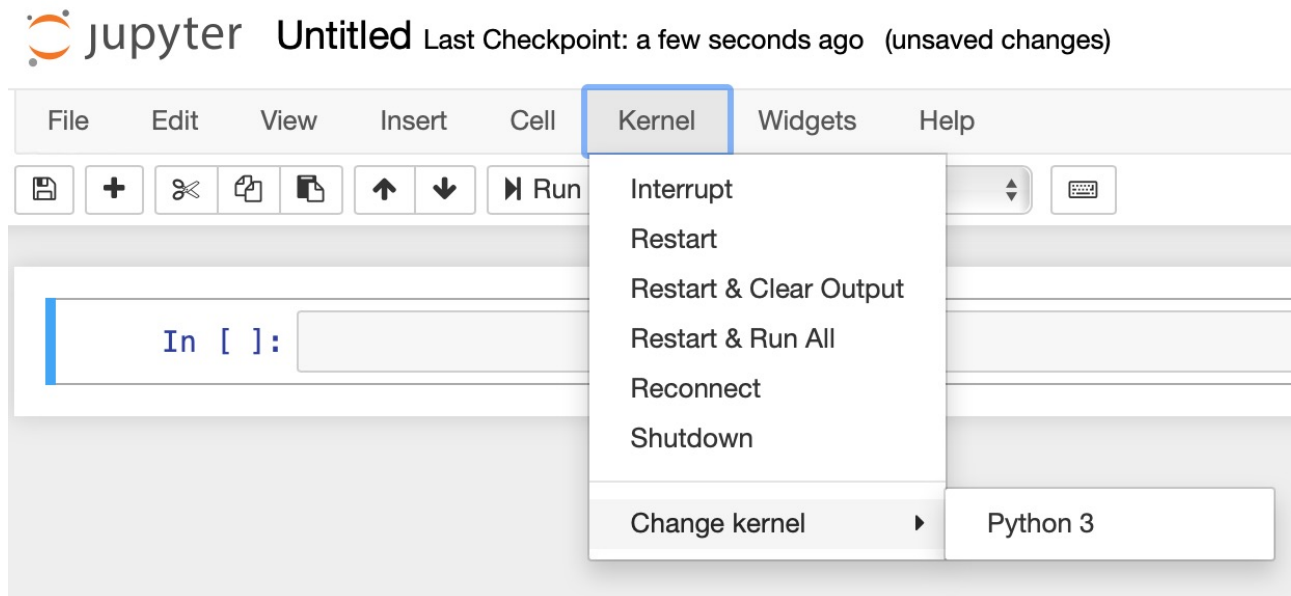
Самыми значимыми компонентами являются:

- **Kernel** (Ядро) – это «вычислительный движок», который выполняет код, содержащийся в документе блокнота
- **Cell** (Ячейка) – это контейнер для текста, который будет отображаться в блокноте, или код, который будет выполняться ядром блокнота



За каждым ноутбуком работает ядро. Когда вы запускаете ячейку кода, этот код выполняется в ядре, и любой вывод возвращается обратно в ячейку для отображения. Состояние ядра сохраняется во времени и между ячейками – оно относится к документу в целом, а не к отдельным ячейкам

Управление ядром находится в пункте меню Kernel:





Для управления работой ядра доступны следующие команды:

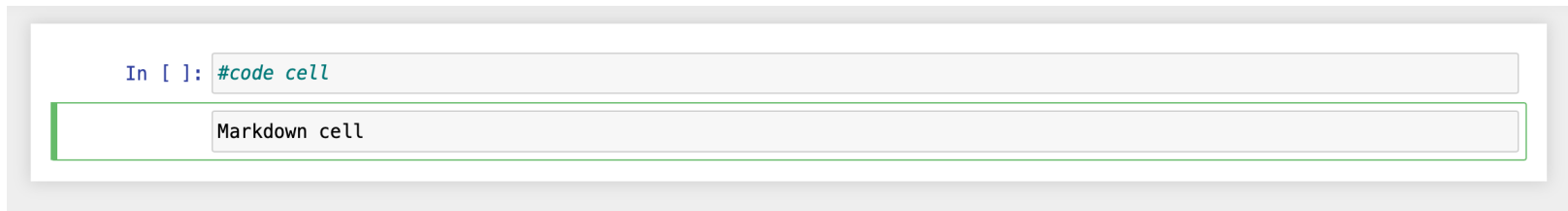
- **Interrupt** – останавливает ядро в случае его зависания во время вычислений
- **Restart** – перезапускает ядро, таким образом очищая все переменные и т.д., которые были определены
- **Restart & Clear Output** – то же, что и выше, но также стирает вывод, отображаемый под ячейками кода
- **Restart & Run All** – то же, что и выше, но также будет запускать все ваши ячейки в порядке от первого до последнего
- **Reconnect** – заново устанавливает соединение с ядром
- **Shutdown** – полностью выключает ядро, удаляет все переменные



Есть два основных типа ячеек:

- **Ячейка кода** содержит код, который должен быть выполнен в ядре, и отображает его вывод ниже:
- **Ячейка Markdown** содержит текст, отформатированный с использованием Markdown, и отображает его вывод на месте при запуске:

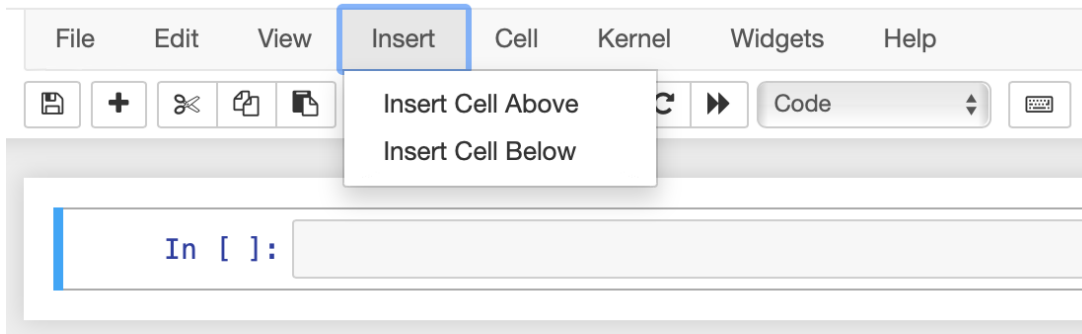
По умолчанию первая ячейка в новом ноутбуке всегда является ячейкой кода. Ячейки кода имеют слева метку `In []`



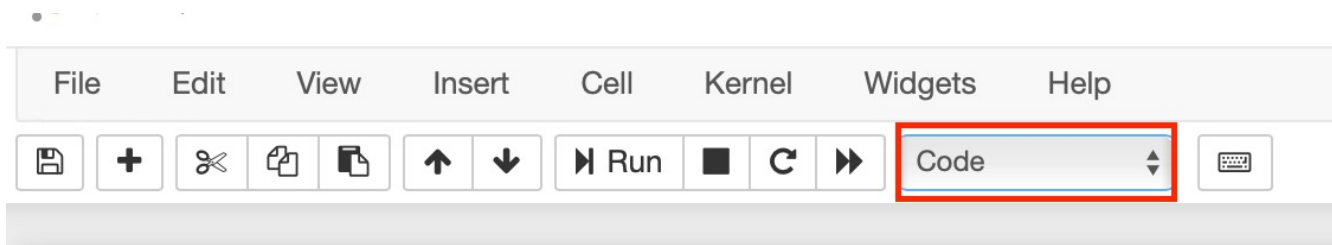


УПРАВЛЕНИЕ ЯЧЕЙКАМИ

Для вставки новой ячейки используется пункт меню Insert. Insert Cell Above вставляет ячейку перед текущей, Insert Cell Below после текущей



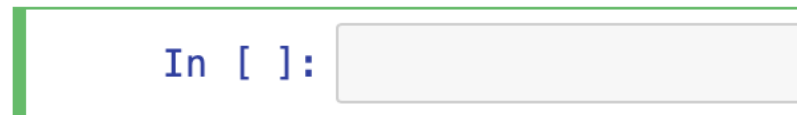
Тип ячейки устанавливается на панели инструментов. Используются только типы Code и Markdown, остальные считаются устаревшими





Активная ячейка выделяется рамкой, цвет которой обозначает ее текущий режим, где зеленый означает edit mode (режим редактирования), а синий – command mode (командный режим)

Эта ячейка находится в режиме редактирования:



А эта в командном режиме:



Переключение между режимом редактирования и командным режимом производится с помощью Esc и Enter соответственно



КОМАНДЫ ЯЧЕЕК

В любом режиме доступны команды запуска ячейки:

- **Ctrl+Enter** – выполнить ячейку
- **Shift+Enter** – выполнить ячейку и создать новую, если текущая ячейка является последней, иначе выполнить и перейти к следующей
- **Alt+Enter** – выполнить текущую ячейку и вставить после неё новую

В командном режиме доступны команды:

- **M** – преобразовать активную ячейку в ячейку Markdown.
- **Y** – установить активную ячейку в кодовую ячейку
- **D + D (D дважды)** – удалить активную ячейку

Список всех команд доступен в пункте меню Help, разделе Keyboard Shortcuts



В ячейке можно писать код на Python так же, как в обычном редакторе

Во время выполнения метка меняется на In [*]. После выполнения в метке In [] появится номер, который указывает, какой по счёту ячейка была выполнена в ядре

Под выполненной ячейкой появится область вывода, содержащая результат работы программы:

```
In [2]: print('Hello, world!')
```

```
Hello, world!
```

```
In [ ]:
```



ЯЧЕЙКА MARKDOWN. ИЗОБРАЖЕНИЯ

Markdown — язык текстовой разметки, предназначенный для создания красиво оформленных текстов, пригодных для преобразования в языки для продвинутых публикаций (HTML, Rich Text и других)

Ячейка Markdown меняет отображение при запуске. Markdown позволяет отображать изображения, это можно сделать следующими способами:

![Альтернативный текст](/путь/к/изображению.jpg)

- Вставить URL для изображения в Интернете.
- Вставить локальный URL-адрес изображения, которое хранится рядом с ноутбуком
- Добавить вложение через «Edit> Insert Image» – это преобразует изображение в строку и сохранит его в файле .ipynb ноутбука



MARKDOWN КОМАНДЫ

Заголовок первого уровня.
Заголовок второго уровня.
Заголовок третьего уровня.

Курсив обозначается *звездочками* или подчеркиванием.
Полужирный шрифт – двойными **звездочками** или подчеркиванием.
Для зачеркнутого текста используются две тильды ~~Уберите это.~~

Список:

1. Первый пункт нумерованного списка.
 2. Второй пункт.
 1. Пункт вложенного списка.
- * Ненумерованный список можно размечать звездочками.

[Ссылка в строке](<https://www.google.com>)

> С помощью цитат очень удобно в письме обозначать исходный текст.
> Эта строка - часть той же цитаты.





LaTeX – это система набора и вёрстки для обмена и публикации научных документов.

В Jupyter LaTeX используется для вставки формул. Чтобы использовать LaTeX-код, необходимо заключить его в символы \$ в ячейке Markdown.

Например, ячейка, содержащая код

```
$e^{i\pi} + 1 = 0$
```

после выполнения выглядит так:

$$e^{i\pi} + 1 = 0$$

Посмотрите пример [*TestCells.ipynb*](#)



СОВЕТ ПО ОФОРМЛЕНИЮ

Хорошей практикой является написание логически разных частей программы в отдельных ячейках. Например, обычно начинают с ячейки кода, специально предназначенной для импорта и настройки, поэтому, если вы решите добавить или изменить что-либо, вы можете просто отредактировать и повторно запустить ячейку, не вызывая побочных эффектов:

```
In [ ]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```



ВЫВОД ГРАФИКОВ

Печать изображений может пригодиться в том случае, если вы используете библиотеку `matplotlib` для построения графиков. По умолчанию, графики не выводятся в рабочее поле ноутбука. Для того, чтобы графики отображались, необходимо ввести и выполнить следующую команду:

```
%matplotlib inline
```

```
In [7]: import matplotlib.pyplot as plt  
%matplotlib inline
```



ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Это всё, что вам нужно знать об использовании Jupyter Notebook. Для дальнейшей работы вам понадобится освежить знания по работе с *numpy* и *matplotlib* (см. предыдущий урок)

Рассмотрите файл, **Mandelbrot.ipynb**, содержащий информацию об интересном объекте – фрактале. В файле приведена общая информация о фракталах и разобран пример фрактала Мандельброта

Дополните файл информацией об одном из фракталов на выбор:

- **треугольник Серпинского,**
- **кривая Коха,**
- **множество Жюлиа,**
- **дерево Пифагора.**

В markdown-блоках опишите теоретические основы вашего фрактала, используйте Markdown и LaTeX

В code-блоках запрограммируйте создание выбранного фрактала и его отрисовку с помощью *numpy* и *matplotlib*

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?



*School of
Computer
Science*