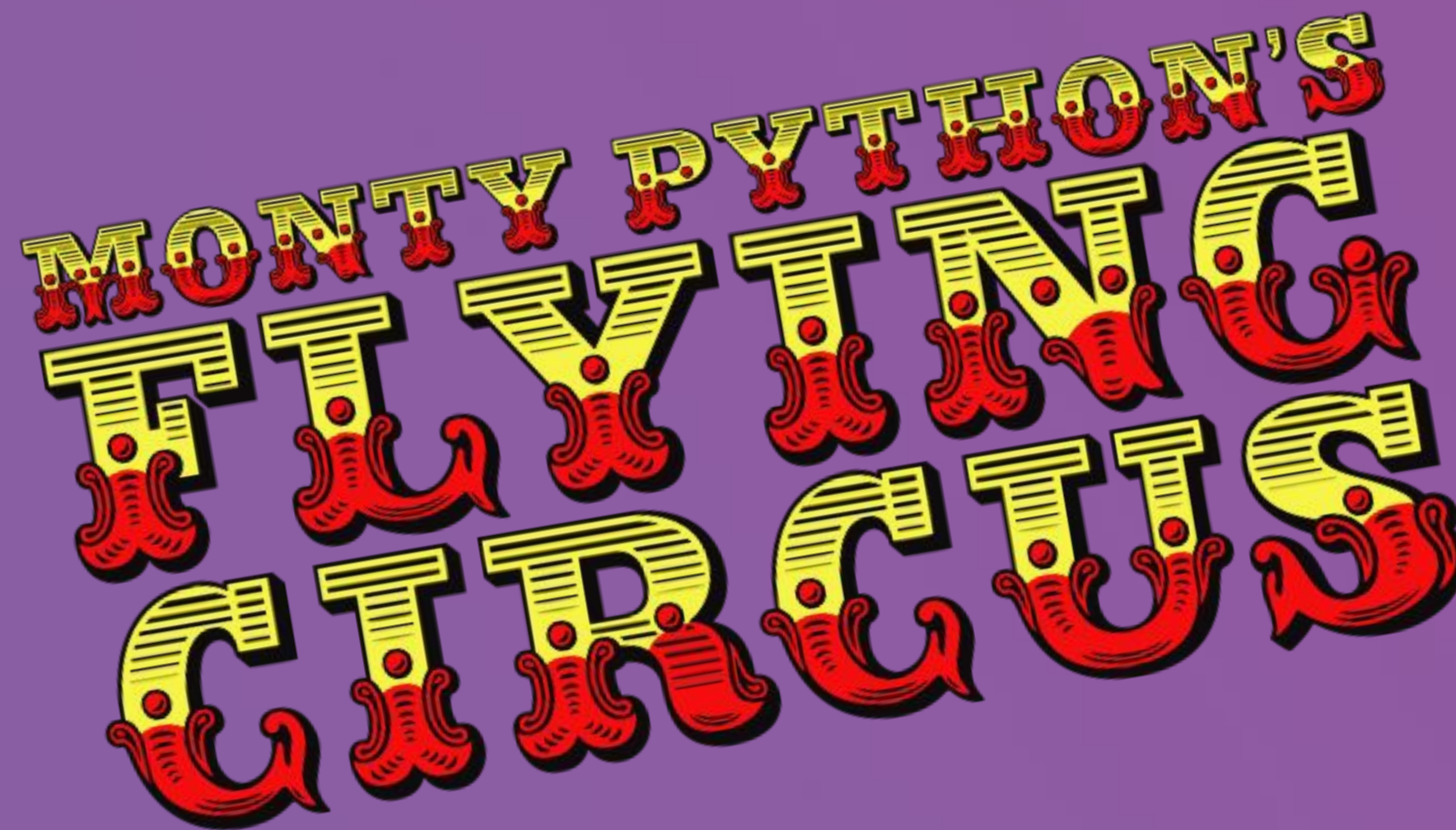


ЗАНЯТИЕ 1

PYTHON



КОНСТАНТИН БАШЕВОЙ

Яндекс

с

Habr

@kpi_maker



О ЧЁМ ПОГОВОРИМ
И ЧТО СДЕЛАЕМ

1. Jupyter notebook: его возможности для команд;
2. Почему все так любят Python?
3. Немного рекомендаций;
4. Простой код и упражнения;
5. Про Git;

ЦЕЛИ ЗАНЯТИЯ

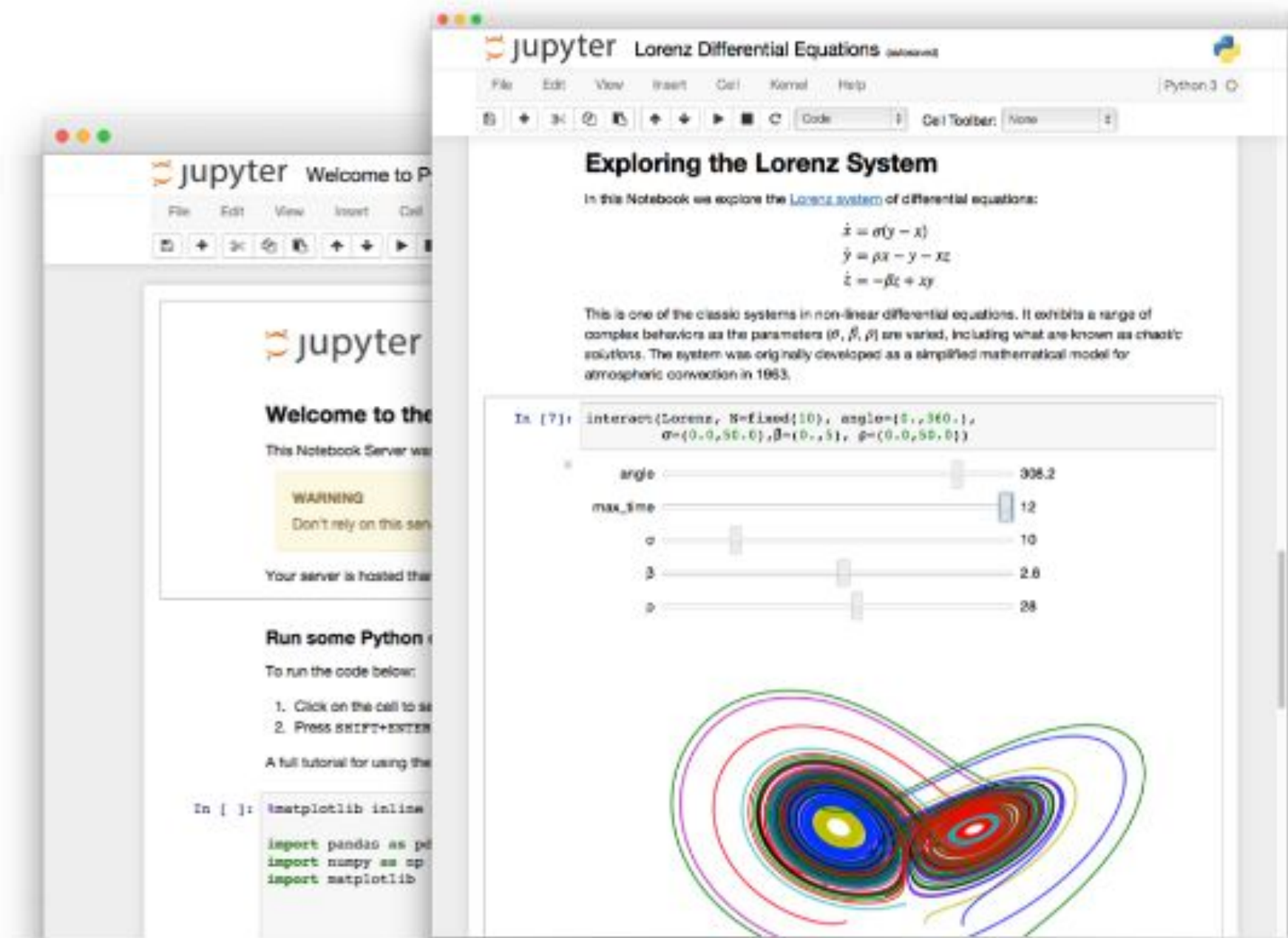
В КОНЦЕ ЗАНЯТИЯ ВЫ СМОЖЕТЕ:

- использовать Jupyter Notebook, чтобы запускать и форматировать код
- проводить арифметические операции в Python
- работать со строками
- выкладывать свой код на Git

JUPYTER NOTEBOOK


JUPYTER NOTEBOOK

По умолчанию установлен в Anaconda. [Установка на удалённом сервере](#)

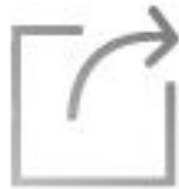


The Jupyter Notebook


The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.




Language of choice



Share notebooks



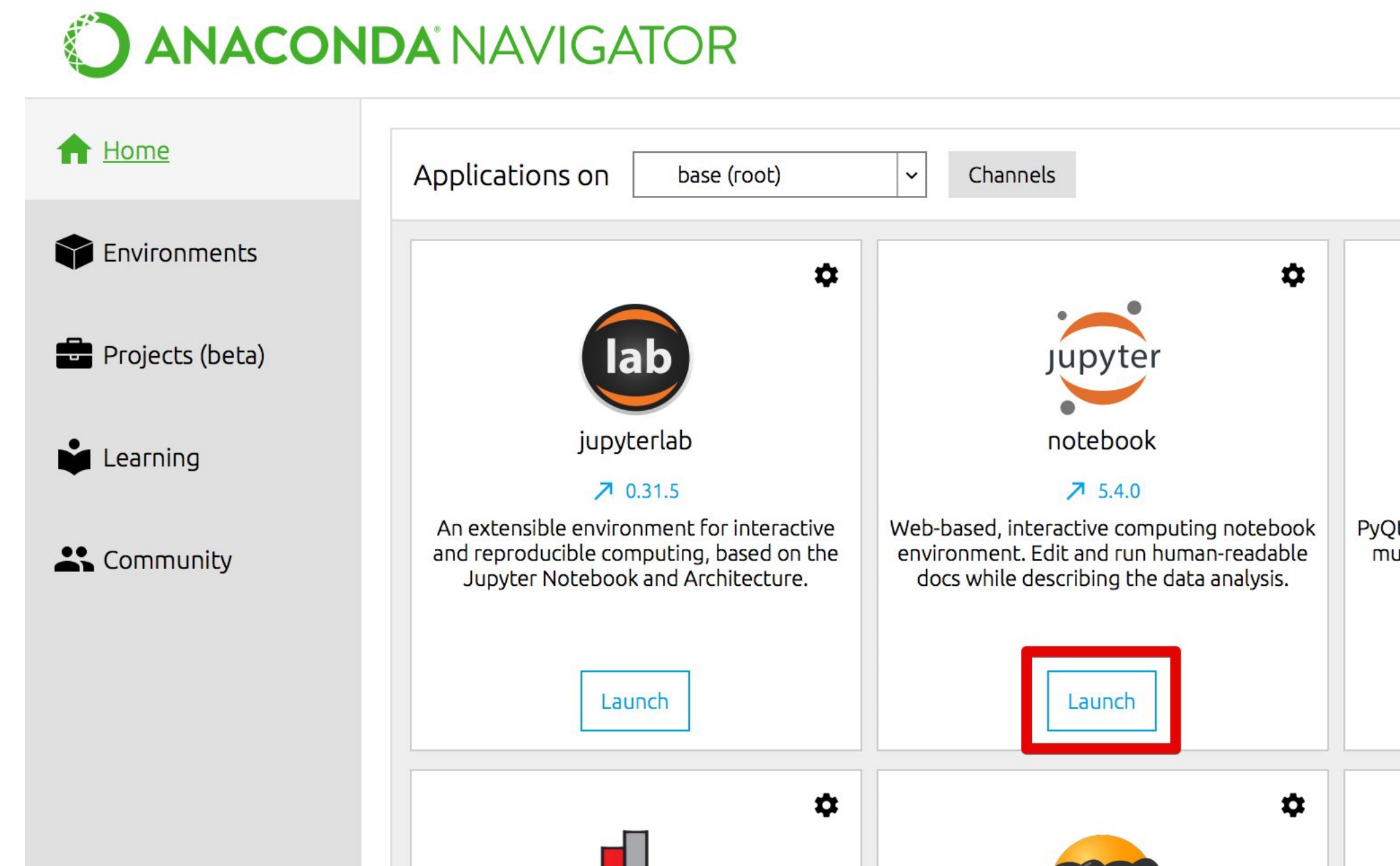
Interactive widgets



Big data integration

КАК ЗАПУСТИТЬ JUPYTER NOTEBOOK

- В командной строке;
- Anaconda prompt;
- Через Anaconda Navigator.



Выполнить команду и перейти на следующую строку

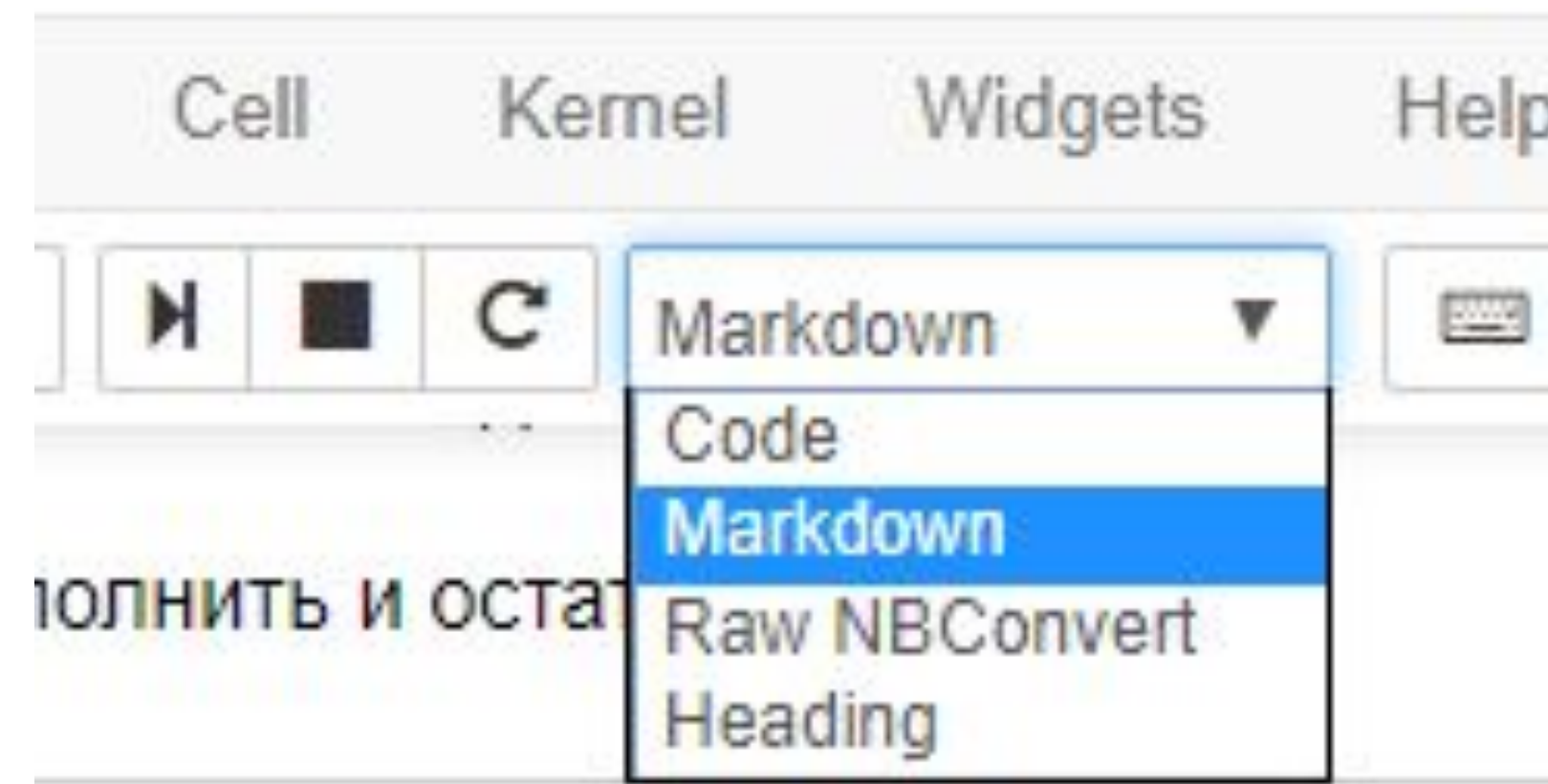
Shift + Enter

Выполнить команду и остаться на строке

Ctrl + Enter

Комментарии и формулы

Markdown



ПОЧЕМУ PYTHON?

PYTHON

1. Простой синтаксис

```
print("Hello world")
```

PYTHON

1. Простой синтаксис

```
print("Hello world")
```

JAVA

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

1. Простой синтаксис

```
print("Hello world")
```

C

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    printf("Hello, world\n");
    return EXIT_SUCCESS;
}
```


1. Простой синтаксис

Brainfuck

```
>+++++++[[<++++++>-]<.>+++++++[[<++++>-]  
<+.+++++..+++. [-]>+++++++[[<++++>-]  
<.>+++++++[[<++++>-]  
<. >+++++++[[<+++>-]<.+ + + . - - - - - . - - - - - . [-]>  
+++++++[[<++++>-]<+. [-]+++++++.>  
+++++++[[<++++++>-]<.>+++++++[[<++++>-]  
<+.+++++..+++. [-]>+++++++[[<++++>-]  
<.#>+++++++[[<++++>-]<. >+++++++  
[[<+++>-]<.+ + + . - - - - - . - - - - - . [-]>+++++++[[<++++>-]  
<+. [-]+++++++.
```

2. Динамическая типизация

```
a = 1
print( 'Тип переменной a ' + str( type(a) ) )

a = '1'
print( 'Тип переменной a ' + str( type(a) ) )
```

Тип переменной a <class 'int'>

Тип переменной a <class 'str'>

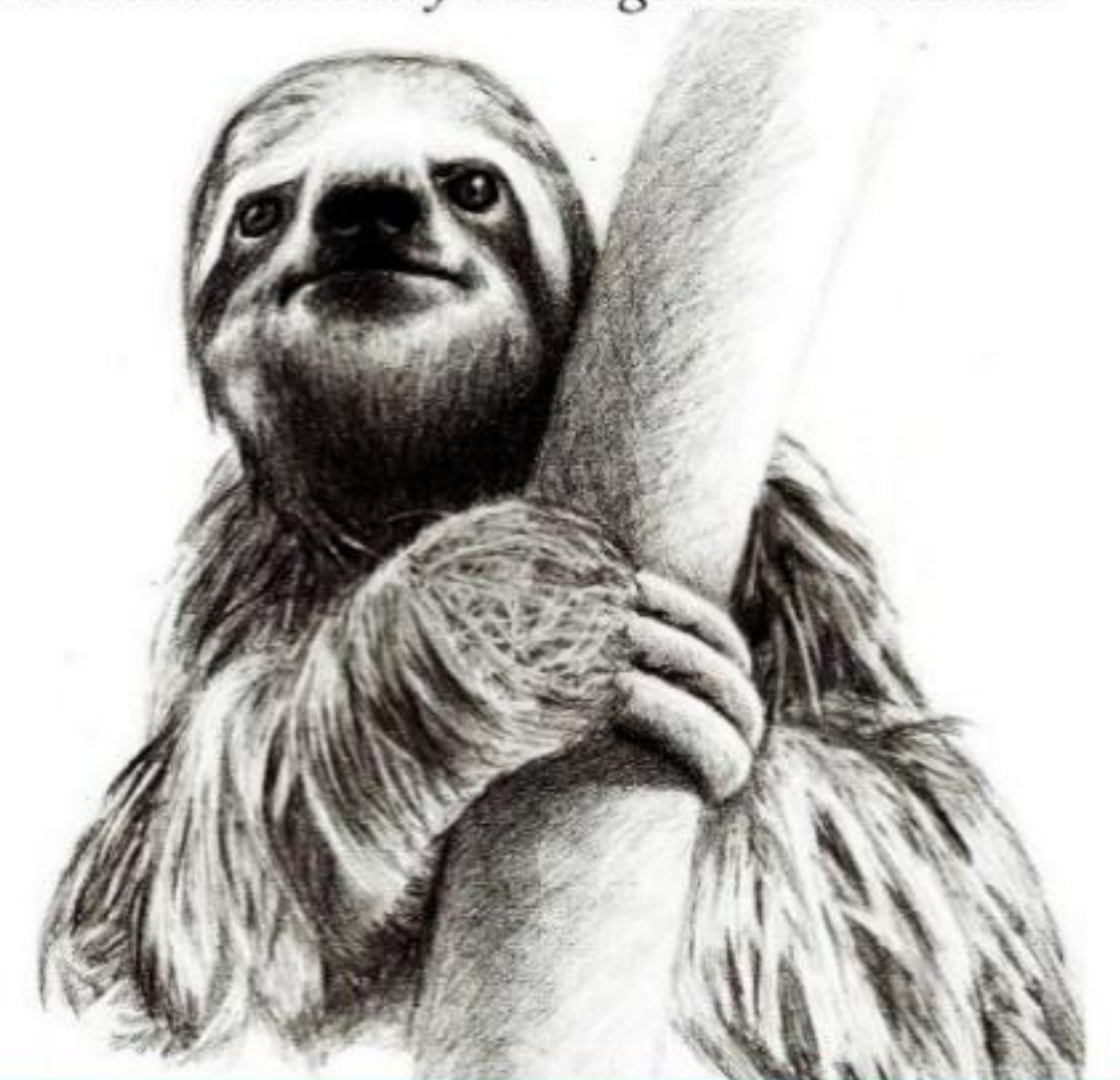
3. Много готовых библиотек для работы с данными.

- все системы пишут API и коннекторы;
- базы данных (pymysql, pymongo), инструменты работы с большими данными (pyspark, Logs API «Яндекс.Метрики»).

PYTHON

4. Большое сообщество

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

*The Practical Developer
@ThePracticalDev*

PYTHON



5. По умолчанию стоит в Linux и Mac

ОБЩИЕ РЕКОМЕНДАЦИИ*

- давайте переменным интуитивно понятные названия;
- пишите комментарии хотя бы иногда;
- разбивайте код на отдельные функции или классы;
- каждая функция должна содержать описание и (желательно) пример ее вызова;

```
def status( u, s ):
    return s[ u ]
```

ТОТ ЖЕ ПРИМЕР С ОПИСАНИЕМ

```
def findUserStatusInDict( userID, statusDict ):
    """
    Поиск статуса пользователя в словаре statusDict по ID пользователя userID.
    Входные данные:
        - userID - целое число
        - statusDict - словарь вида:
            statusDict = {
                1: 'Админ',
                2: 'Блокирован',
                3: 'Активен'
            }

    Пример:
    findUserStatusInDict( 2, statusDict )

    'Блокирован'

    Если ID пользователя нет в словаре statusDict, то возвращается 'Неизвестный ID пользователя'
    """

    if userID in statusDict:
        return statusDict[ userID ]

    else:
        return 'Неизвестный ID пользователя'
```

ГЛАВНАЯ РЕКОМЕНДАЦИЯ

«Пишите код так, как будто
сопровождать его будет
склонный к насилию психопат,
который знает, где вы живете»



КОГДА ОТКРЫЛ СВОЙ СТАРЫЙ ПРОЕКТ



GIT

ПОЛЕЗНЫЕ КОМАНДЫ

Первоначальная настройка

```
git config --global user.name "Ivan Ivanov"
```

```
git config --global user.email iivanov@mywork.com
```

Когда репозиторий уже есть в папке (.git)

```
git add .
```

```
git commit -m "что было сделано"
```

```
git push origin master
```

ПОЛЕЗНЫЕ МАТЕРИАЛЫ

Git за 20 минут <https://bit.ly/2Nauj4D>

(домашние задания лучше выкладывать на Github)

Что за PEP8 <https://bit.ly/2y65xOC>

Параметры, логины и пароли лучше хранить в YAML-
файлах <https://bit.ly/2BBUGPp>

Подпишитесь на Хабр 😊



НЕТОЛОГИЯ
групп

Спасибо за внимание!

КОНСТАНТИН БАШЕВОЙ

Habr: @kpi_maker