Знакомство с основными парадигмами программирования. Обзор современных языков программирования. Знакомство с языком Python. **Сферы применения** Python.

Что сегодня будет?

- Проведем краткий обзор ЯП
- Подробней рассмотрим основные парадигмы ЯП
- Что такое Python и где он применяется (попробуем написать программу для анализа данных)
- Решим простую задачу

Современные ЯП

- JavaScript

Прототипно-ориентированный сценарный язык программирования. JavaScript изначально создавался для того, чтобы сделать web-странички «живыми». В браузере они подключаются напрямую к HTML и, как только загружается страничка – тут же выполняются.

Когда создавался язык JavaScript, у него изначально было другое название: «LiveScript». Но тогда был очень популярен язык Java, и маркетологи решили, что схожее название сделает новый язык более популярным.

Планировалось, что JavaScript будет эдаким «младшим братом» Java. Однако, история распорядилась по-своему, JavaScript сильно вырос, и сейчас это совершенно независимый язык, со своей спецификацией и к Java не имеет никакого отношения.

JavaScript™

Java

Строго типизированный объектно-ориентированный язык программирования. Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной.

Изначально язык назывался Oak («Дуб») разрабатывался Джеймсом Гослингом для программирования бытовых электронных устройств. Впоследствии он был переименован в Java и стал использоваться для написания клиентских приложений и серверного программного обеспечения.

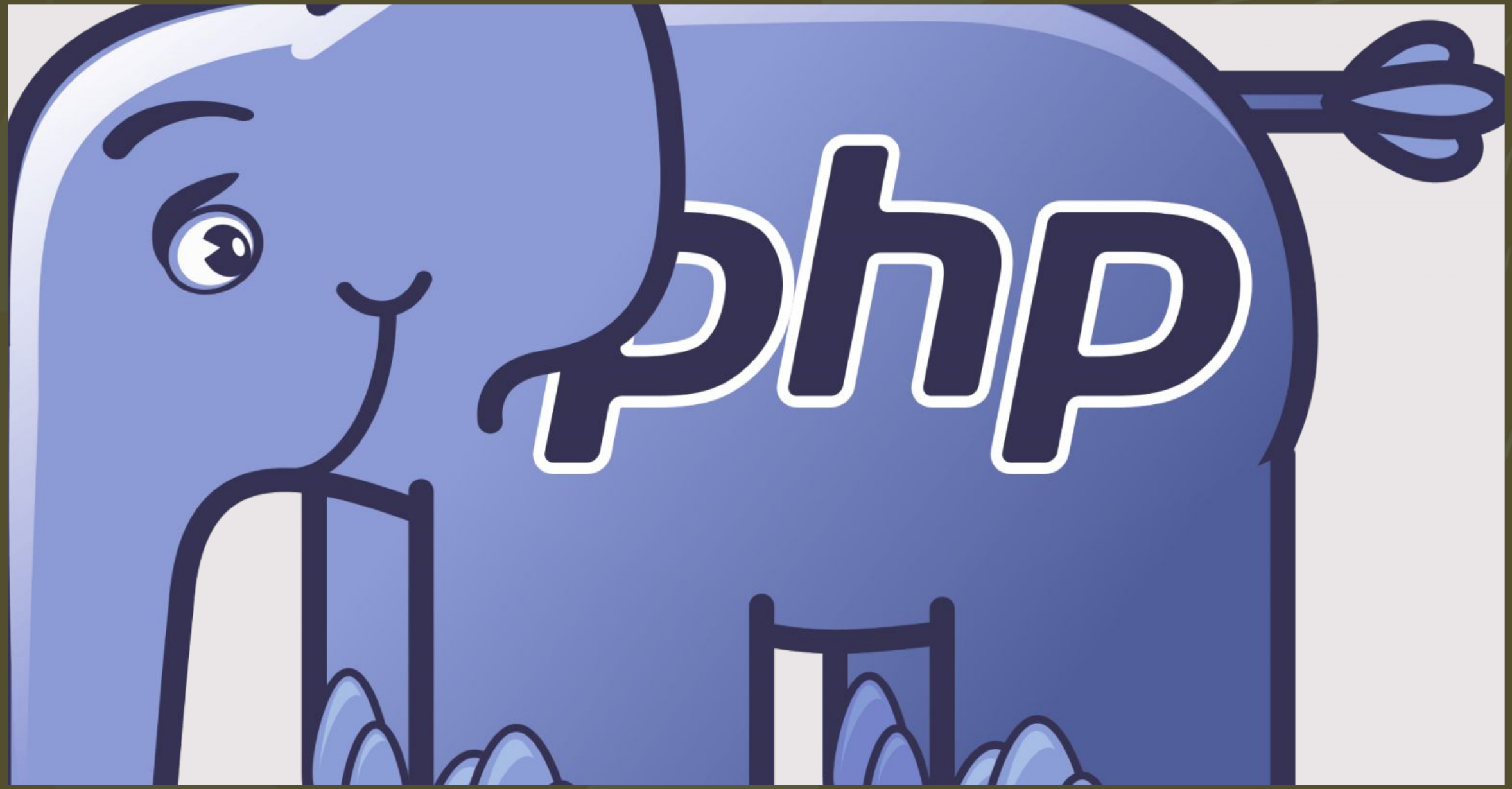
Java™

ORACLE

PHP

Является распространенным интерпретируемым языком общего назначения с открытым исходным кодом (скриптовый язык). PHP создавался специально для ведения web-разработок и код на нем может внедряться непосредственно в HTML-код. Синтаксис языка берет начало из C, Java и Perl, и является легким для изучения.

Основной целью PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц, однако область применения PHP не ограничивается только этим.


php

Objective-C

Компилируемый объектно-ориентированный язык программирования, используемый корпорацией Apple, построенный на основе языка Си и парадигм Smalltalk. Язык Objective-C является надмножеством языка Си, поэтому Си-код полностью понятен компилятору Objective-C. Язык используется в первую очередь для Mac OS X (Cocoa) и GNUstep — реализаций объектно-ориентированного интерфейса OpenStep. Также язык используется для iOS (Cocoa Touch).


OBJECTIVE-C

Go

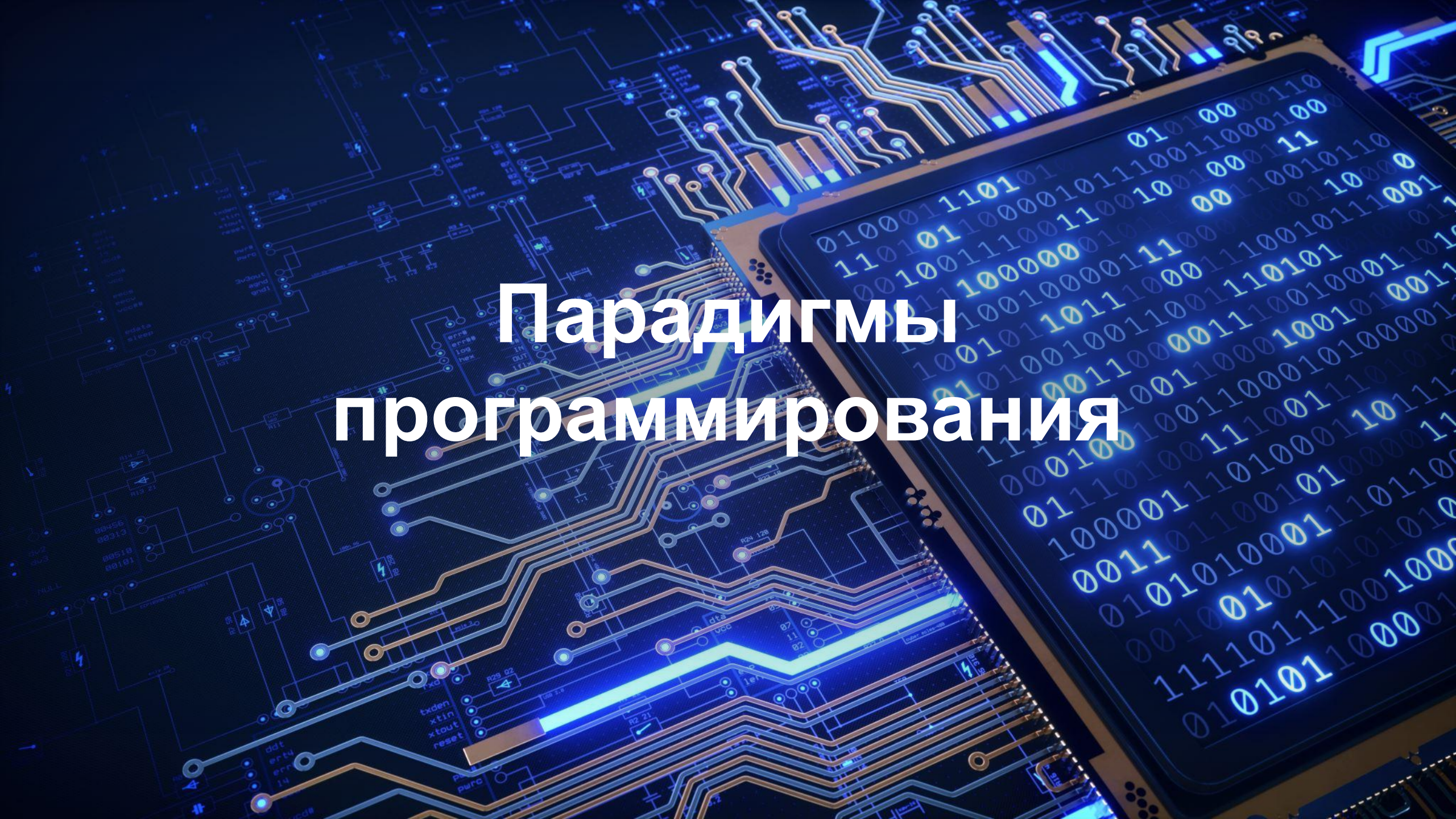
Компилируемый многопоточный язык программирования, разработанный компанией Google. Язык Go разрабатывался как язык системного программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языку Си. При разработке уделялось особое внимание обеспечению высокоэффективной компиляции. Программы на Go компилируются в объектный код и не требуют для исполнения виртуальной машины.


Go

Swift

Открытый мультипарадигмальный компилируемый язык программирования общего назначения. Создан компанией Apple в первую очередь для разработчиков iOS и OS X. Swift работает с фреймворками Cocoa и Cocoa Touch и совместим с основной кодовой базой Apple, написанной на Objective-C. Swift задумывался как более легкий для чтения и устойчивый к ошибкам программиста язык, нежели предшествовавший ему Objective-C. Swift заимствовал довольно многое из Objective-C, однако он определяется не указателями, а типами переменных, которые обрабатывает компилятор. По аналогичному принципу работают многие скриптовые языки.


Swift


Парадигмы
программирования

Существует три основных парадигмы:

- Структурное
- объектно-ориентированное
- функциональное.

Структурное программирование

- Структурное программирование было открыто Дейкстрой в 1968 году. Он понял, что goto – это зло, и программы должны строиться из трёх базовых структур: последовательности, ветвления и цикла.

- Дейкстра понял, что программирование – это сложно. Большие программы имеют слишком большую сложность, которую человеческий мозг не способен контролировать.
- Чтобы решить эту проблему, Дейсктра решил сделать написание программ подобно математическим доказательствам, которые также организованы в иерархии. Он понял, что если в программах использовать только if, do, while, то тогда такие программы можно легко рекурсивно разделять на более мелкие единицы, которые в свою очередь уже легко доказуемы.
- С тех пор оператора goto не стало практически ни в одном языке программирования.
- Таким образом, структурное программирование позволяет делать функциональную декомпозицию.
- Однако на практике мало кто реально применял аналогию с теоремами для доказательства корректности программ, потому что это слишком накладно. В реальном программировании стал популярным более «лёгкий» вариант: тесты. Тесты не могут доказать корректности программ, но могут доказать их некорректность. Однако на практике, если использовать достаточно большое количество тестов, этого может быть вполне достаточно.

Объектно-ориентированное программирование

- **ООП** – это парадигма, которая характеризуется наличием инкапсуляции, наследования и полиморфизма.
- **Инкапсуляция** позволяет открыть только ту часть функций и данных, которая нужна для внешних пользователей, а остальное спрятать внутри класса.

- **Наследование** позволяет делать производные структуры на основе базовых, тем самым давая возможность осуществлять повторное использование этих структур. Наследование было реально сделать в языках до ООП, но в объектно-ориентированных языках оно стало значительно удобнее.
- **Полиморфизм** – это ключевое свойство ООП для построения грамотной архитектуры. Он позволяет сделать модуль независимым от конкретной реализации (реализаций) интерфейса. Этот принцип называется инверсией зависимостей, на котором основаны все плагинные системы.

Функциональное программирование

- В основе функционального программирования лежит запрет на изменение переменных. Если переменная однажды проинициализирована, её значение так и остаётся неизменным.
- Применяя функциональный подход, мы разделяем компоненты на изменяемые и неизменяемые. Причём как можно больше функциональности нужно положить именно в неизменяемые компоненты и как можно меньше в изменяемые. В изменяемых же компонентах приходится работать с изменяемыми данными, которые можно защитить с помощью транзакционной памяти.
- Интересным подходом для уменьшения изменяемых данных является Event Sourcing. В нём мы храним не сами данные, а историю событий, которые привели к изменениям этих данных. Так как в лог событий можно только дописывать, это означает, что все старые события уже нельзя изменить. Чтобы получить текущее состояние данных, нужно просто воспроизвести весь лог.

Что такое Python **и где он применяется**

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. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами.

3 **самых важных сферы применения** Python: **возможности языка**

- **Веб-разработка**
- Относительно недавно в веб-разработке стали очень популярны Python-фреймворки, такие как Django и Flask. Они облегчают процесс написания на языке Python кода серверной части приложений. Это тот код, который запускается на сервере, а не на устройствах и браузерах пользователей (frontend-код).
- **Зачем нужен веб-фреймворк**?

Фреймворки позволяют легко и быстро создать базовую логику бэкенда. Она включает в себя сопоставление разных URL-адресов с частями Python-кода, работу с базами данных, создание HTML-представлений для отображения на устройствах пользователя.

Data Science: **машинное обучение**

- Машинное обучение (ML) — это использование математических моделей данных, которые помогают компьютеру обучаться без непосредственных инструкций. Оно считается одной из форм искусственного интеллекта (ИИ). При машинном обучении с помощью алгоритмов выявляются закономерности в данных.

Существуют разные библиотеки и фреймворки для машинного обучения на Python. Две самые популярные – это scikit-learn и TensorFlow.

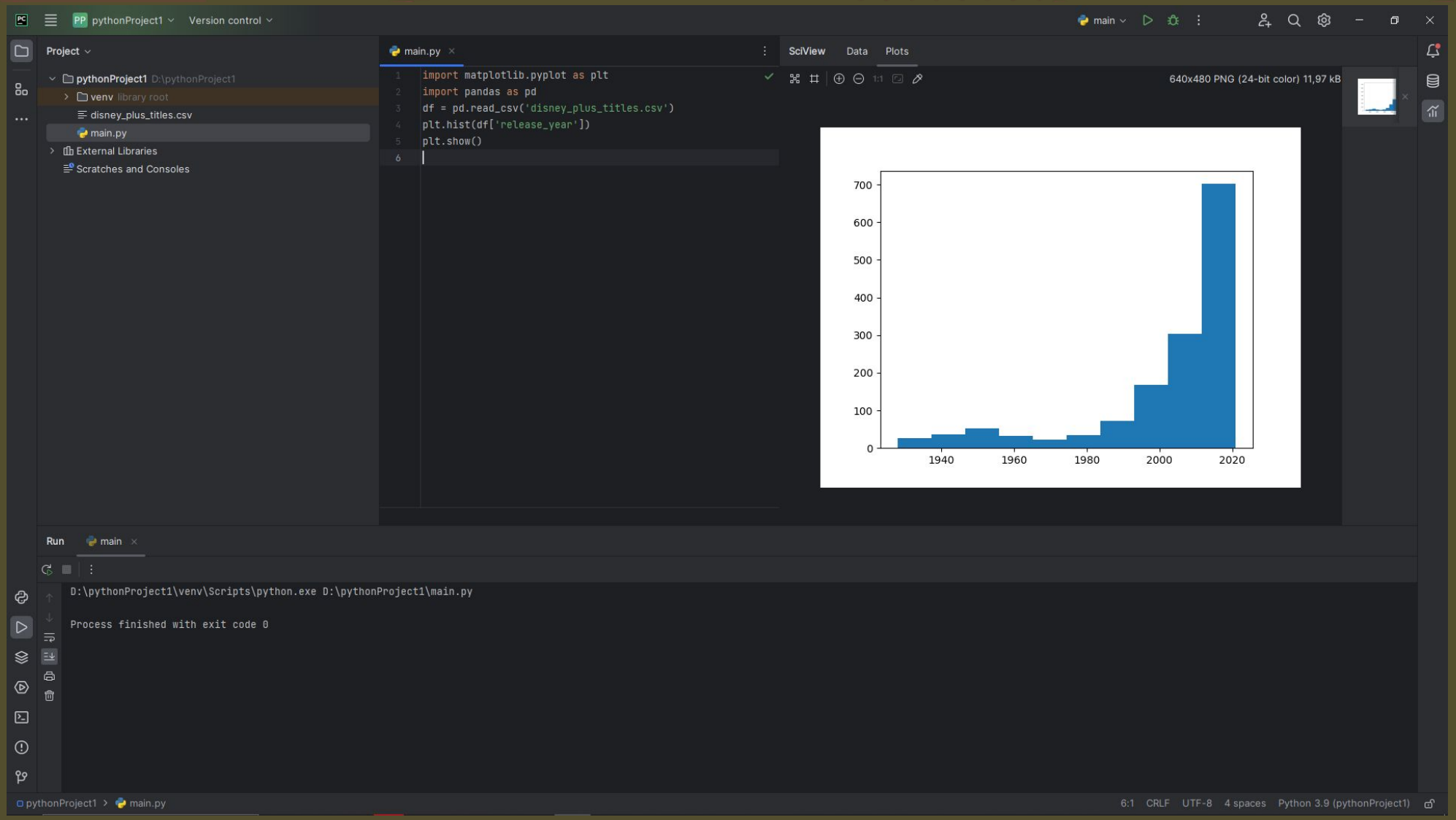
- scikit-learn из коробки имеет несколько встроенных популярных алгоритмов обучения;
- TensorFlow – это более низкоуровневая библиотека. Она позволяет создавать пользовательские алгоритмы.

Новичкам в машинном обучении лучше начать со scikit-learn. Более опытным разработчикам, которые столкнулись с проблемами эффективности, стоит присмотреться к TensorFlow.

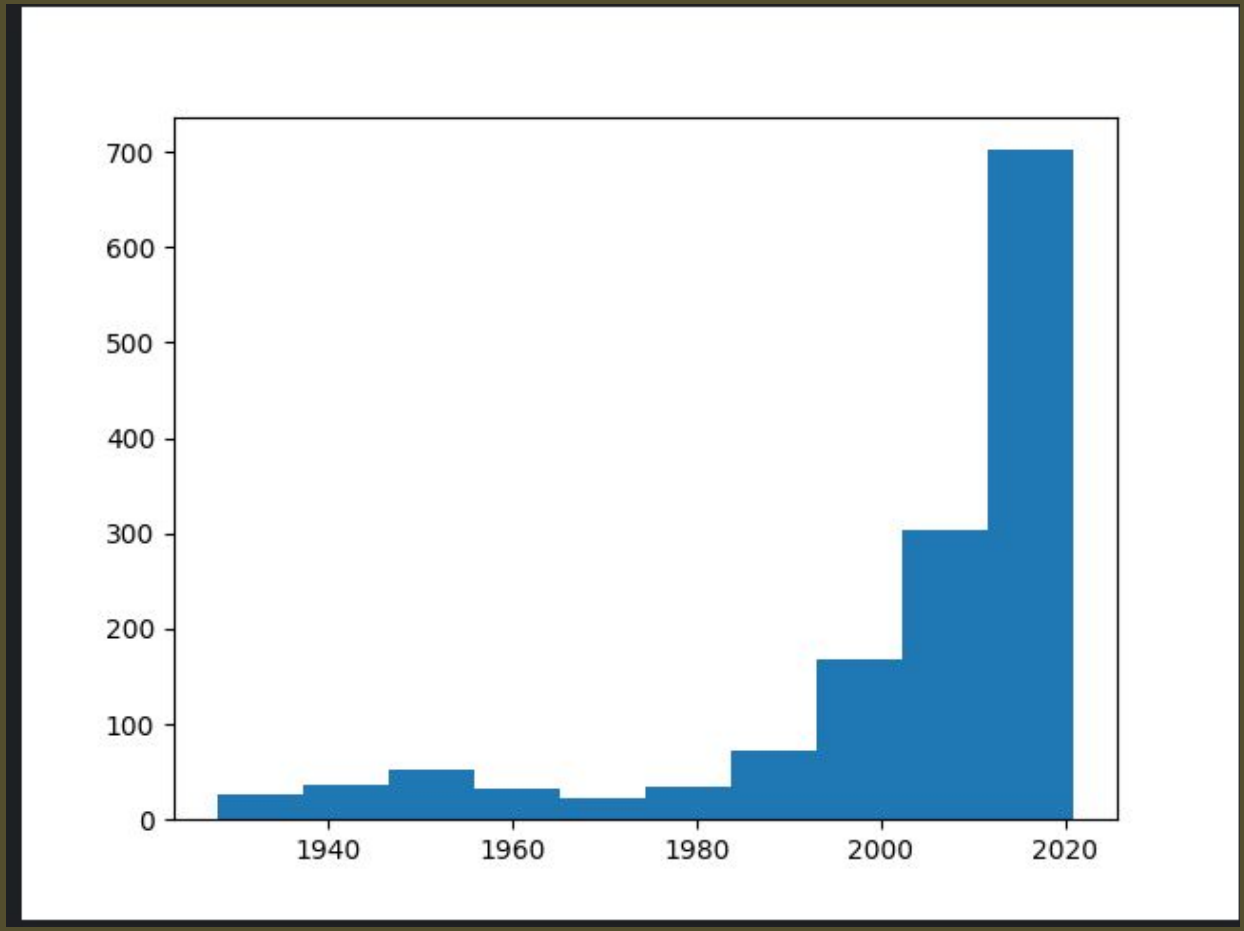
Анализ и визуализация данных

- В Python существует множество библиотек для визуализации данных, однако базовой и одной из самых распространённых является библиотека matplotlib.
- Создание визуализаций можно представить как последовательное добавление различных элементов итогового графика:
- пустого графика с осями;
- элементов, отображающих переменные;
- подписей;
- цвета и стиля.

Пример

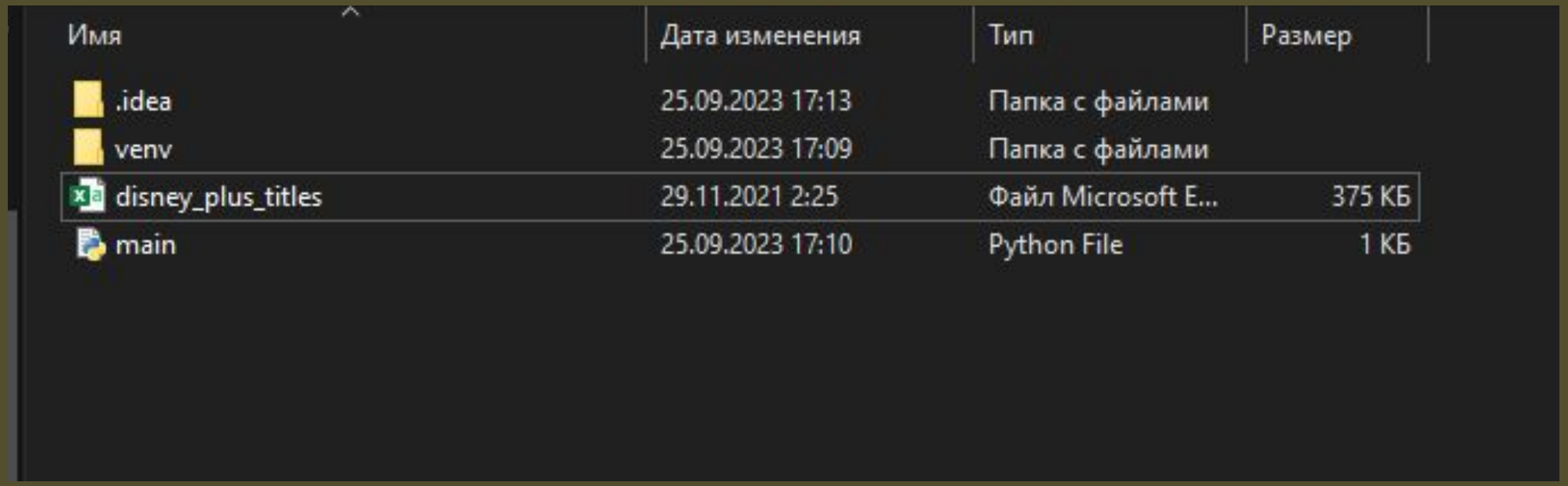

pythonProject1
Version control
main
Project
pythonProject1 D:\pythonProject1
venv library root
disney_plus_titles.csv
main.py
External Libraries
Scratches and Consoles
main.py
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('disney_plus_titles.csv')
plt.hist(df['release_year'])
plt.show()
SciView
Data
Plots
640x480 PNG (24-bit color) 11,97 kB
700
600
500
400
300
200
100
0
1940
1960
1980
2000
2020
Run
main
D:\pythonProject1\venv\Scripts\python.exe D:\pythonProject1\main.py
Process finished with exit code 0
pythonProject1 > main.py
6:1
CRLF
UTF-8
4 spaces
Python 3.9 (pythonProject1)

- По оси X указан год производства, а по оси Y — количество фильмов, произведённых в этот год. Как видно из этого графика, производство фильмов начало расти после восьмидесятых, причём после начала нулевых производство фил



Шаг 1

- Создаем проект и в его папку закидываем документ «disney_plus_titles »



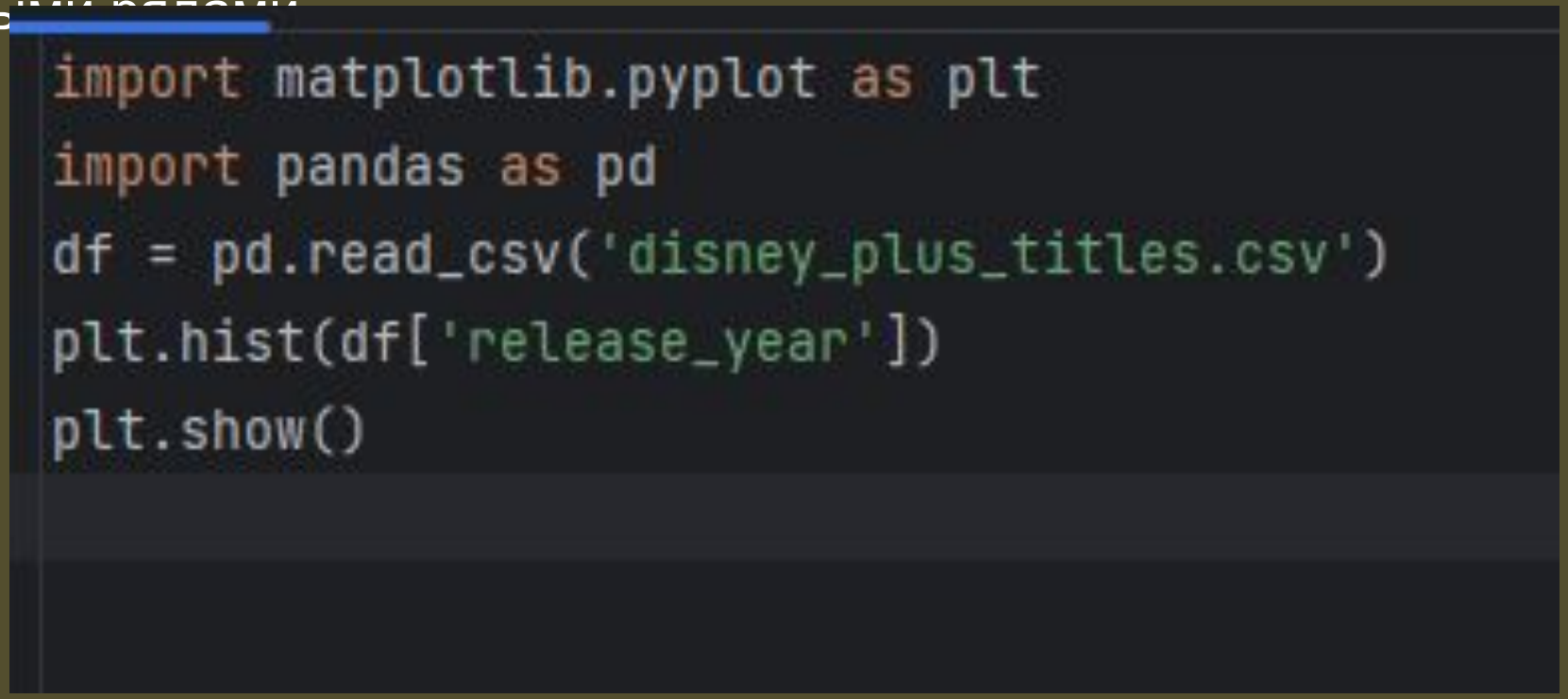
Шаг 2

- Импортируем библиотеки.
- Изначально они будут подчеркнуты красным. Нужно навести на них мышкой и нажать «install»

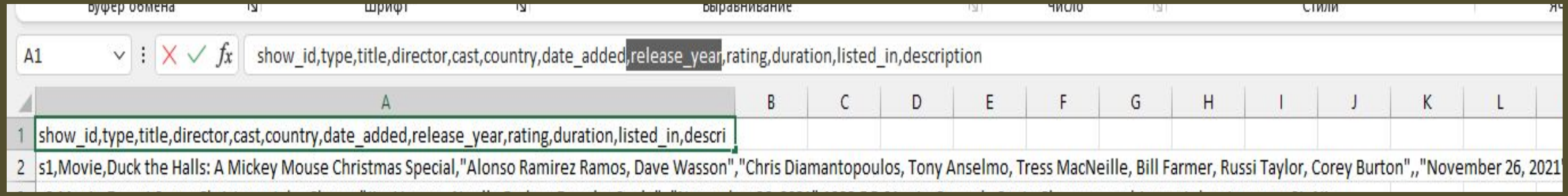
```
import matplotlib.pyplot as plt
import pandas as pd
```

Шаг 3

- Вводим код
- 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.



```
plt.hist(df['release_year'])
```



Автоматизация процессов

- Одна из самых популярных сфер применения Python – это написание небольших скриптов для автоматизации различных рабочих операций и процессов.
- В качестве примера можно привести систему обработки электронной почты. Для сбора статистики и анализа данных требуется подсчитывать количество входящих писем, содержащих определенные ключевые слова. Это можно делать вручную, или же написать простой скрипт, который все посчитает сам.
- Есть несколько причин применения Python для задач автоматизации:
- простой синтаксис, позволяющий быстро писать сценарии;
- легкость отладки, связанная с тем, что код не компилируется перед запуском.

Вспомним прошлое

- Красный, синий и желтый называются основными цветами, потому что их нельзя получить путем смешения других цветов. При смешивании двух основных цветов получается вторичный цвет:
- если смешать красный и синий, то получится фиолетовый;
- если смешать красный и желтый, то получится оранжевый;
- если смешать синий и желтый, то получится зеленый.
- Напишите программу, которая считывает названия двух основных цветов для смешивания. Если пользователь вводит что-нибудь помимо названий «красный», «синий» или «желтый», то программа должна вывести сообщение об ошибке. В противном случае программа должна вывести название вторичного цвета, который получится в результате.
- **Для решения задачи используйте условия or и and**

pythonProject1 Version control

main

Project

pythonProject1 D:\pythonProject1

venv library root

disney_plus_titles.csv

main.py

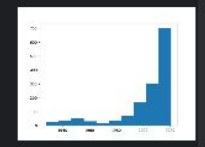
External Libraries

Scratches and Consoles

main.py

```python
a, b = input(), input()
if (a == 'красный' and b == 'синий') or (a == 'синий' and b == 'красный'):
    print('фиолетовый')
elif (a == 'красный' and b == 'желтый') or (a == 'желтый' and b == 'красный'):
    print('оранжевый')
elif (a == 'синий' and b == 'желтый') or (a == 'желтый' and b == 'синий'):
    print('зеленый')
else:
    print('ошибка цвета')
```

SciView Data Plots



else

Run main

```
D:\pythonProject1\venv\Scripts\python.exe D:\pythonProject1\main.py
красный
желтый
оранжевый

Process finished with exit code 0
```

pythonProject1 > main.py

9:26 CRLF UTF-8 4 spaces Python 3.9 (pythonProject1)

ДЗ

- 1. Установите PyCharm
- Используя уроки попробуйте провести собственный анализ данных(постройте гистограмму)
- Используя предоставленную статью постройте точечный график с указанными данными о фильмах Дисней (если получится, то еще с данными о кошках и собаках или людях)