

# Библиотеки Python

# Содержание урока

- ★ Что такое и зачем?
- ★ Какие бывают библиотеки
- ★ Стандартная библиотека Python
- ★ Библиотека requests. Воркшоп
- ★ Библиотека pandas. Воркшоп

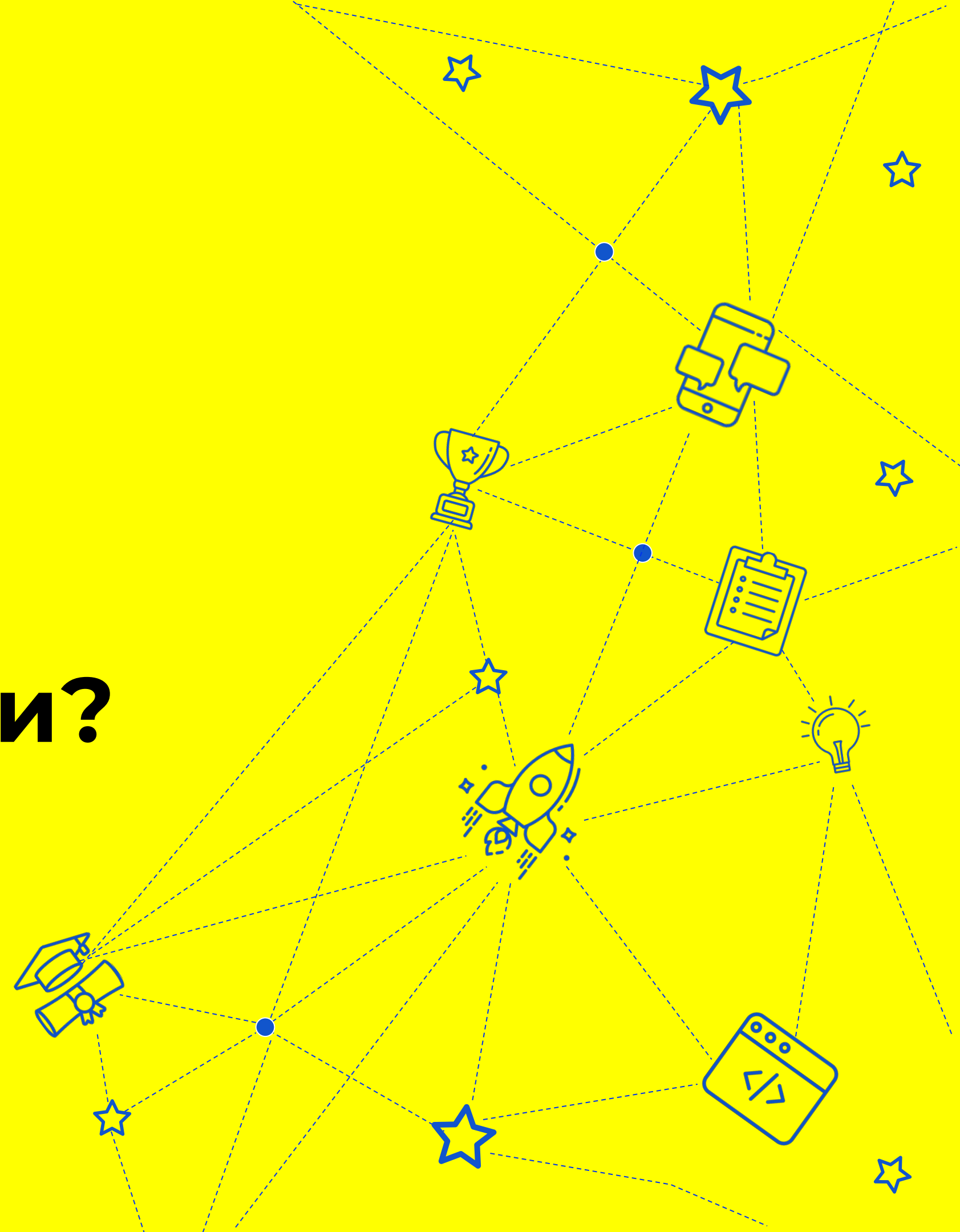


# ЖАМШИД РУЗИЕВ

**Ведущий ML разработчик,  
Яндекс**

- Делаю рекомендательную систему для рекламной сети Яндекса
- Преподаю на факультете компьютерных наук ВШЭ
- Изредка [пишу](#) статьи

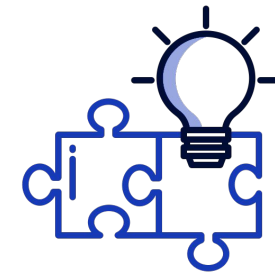
# Что такое библиотека и зачем это придумали?



# Желание упрощать



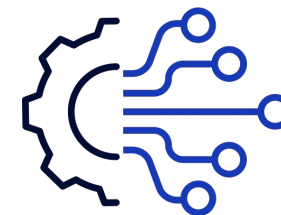
Программисты пишут код



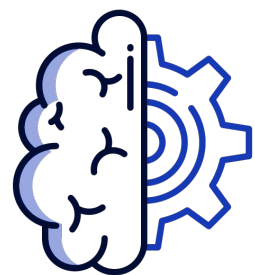
Возникает натуральное желание группировать



В какой-то момент кода становится **слишком** много

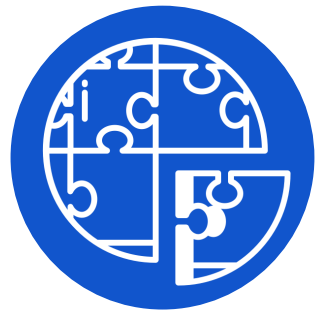


Код пишется и читается не как единое целое, а как набор «логических блоков», секций, **подпрограмм**.



Человеческий мозг ограничен и не может держать в себе большое количество мелких подробностей одновременно

# Разделяй и властвуй



**«Никакое дело не покажется невыполнимым, если разбить его на мелкие части»**

**Генри Форд**

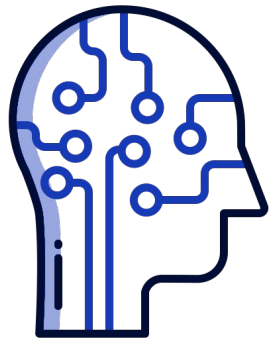
Построение большой программы выглядит:

☆ Не «сели и написали 20 тысяч строк кода за ночь, как крутой хакер из фильмов»

☆ А вот так:



# Пример проектирования



Хотим телеграм-бота, который сидит в чате и переводит все сообщения на иностранных языках на русский.

## 1 Верхнеуровневые логические блоки:

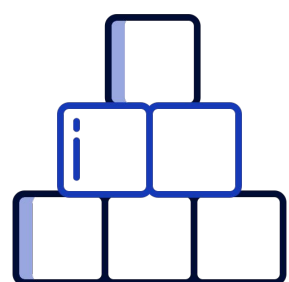
- ★ Получить сообщение из чата
- ★ Нужно ли его переводить?
- ★ Определить язык оригинала
- ★ Перевести с оригинала на русский
- ★ Отправить сообщение обратно в чат

## 2 Идем глубже. Логические блоки секции перевода:

- ★ Допустим, пользуемся сервисом переводов
- ★ Сформировать запрос на перевод текста
- ★ Отправить запрос по нужному адресу
- ★ Дождаться ответа. Если не получилось – попробовать еще раз
- ★ Вернуть переведенный текст (или текст об ошибке)

# Библиотека

Или «библиотека функций», «библиотека объектов», «библиотека прочих полезностей»



Существуют готовые лего-кубики – **компоненты**.

Каждый кубик выполняет свою мини-функцию:

- ★ представляет из себя самостоятельный (или не очень) логический блок
- ★ выглядит как функция / класс / объект / иной кусок кода



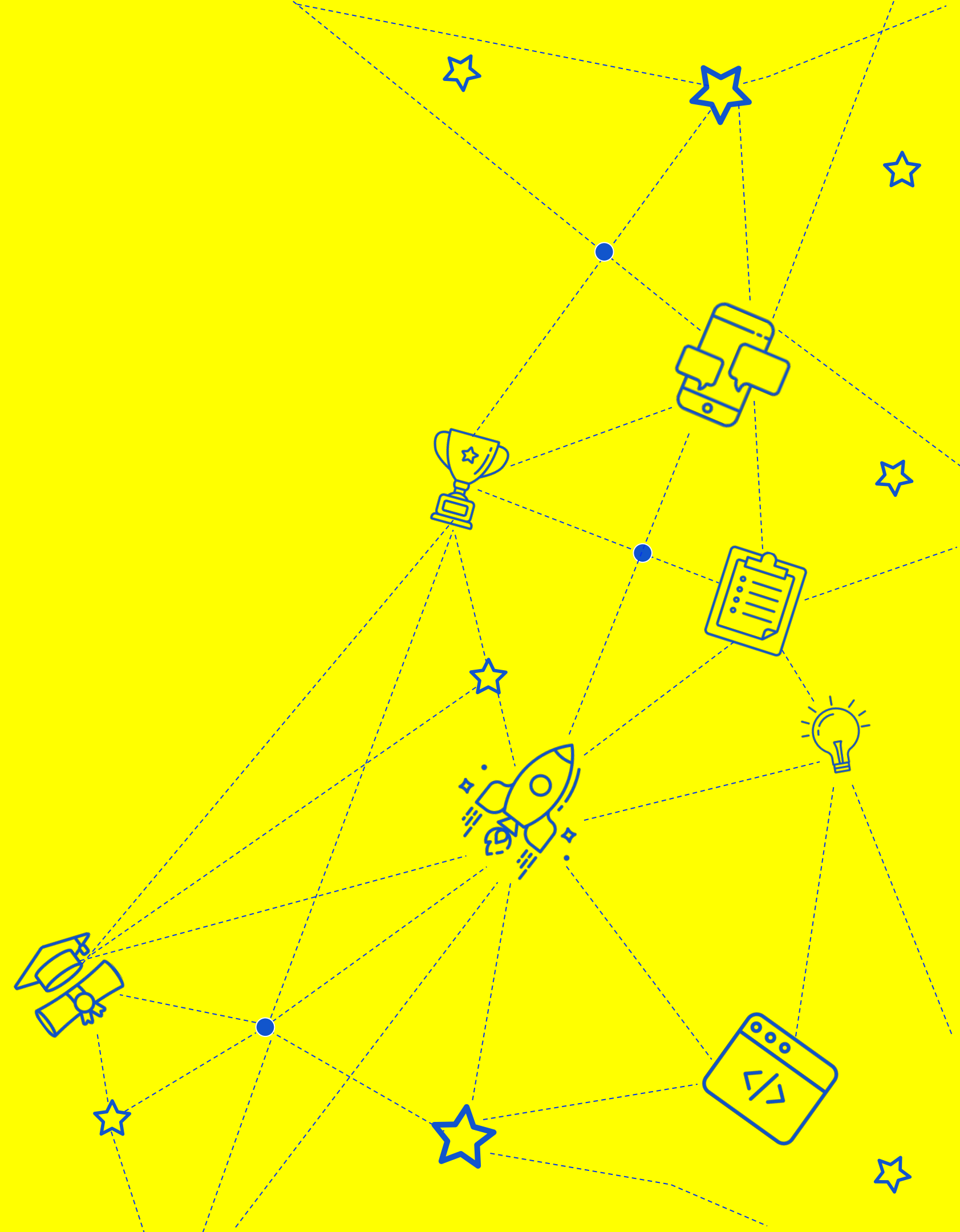
Набор лего-кубиков – **библиотека**.

Обычно библиотека представляет из себя набор компонентов, которые помогают в какой-то конкретной одной задаче:

- ★ работа с изображениями
- ★ отправка sms сообщений
- ★ работа с геолокацией



# Типы библиотек

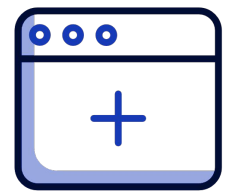


# По функционалу

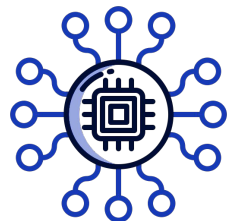
## Сколько задач – столько библиотек



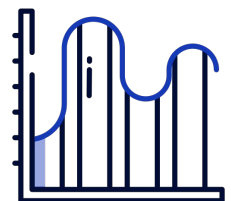
Для работы с видео / аудио / изображениями



Для создания серверных приложений (веб-сайты)



Для вычислений и работы с данными



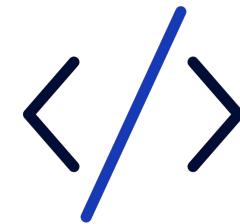
Для визуализации – построения графиков и отчетов



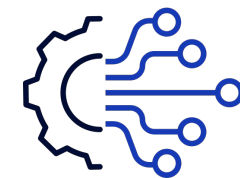
Для работы с файлами



Для работы с сетями



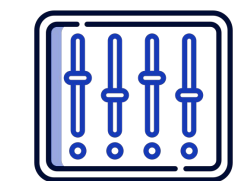
Для тестирования кода



Для логирования хода работы программы



Для анализа кода самой программы



Для расширения возможностей языка

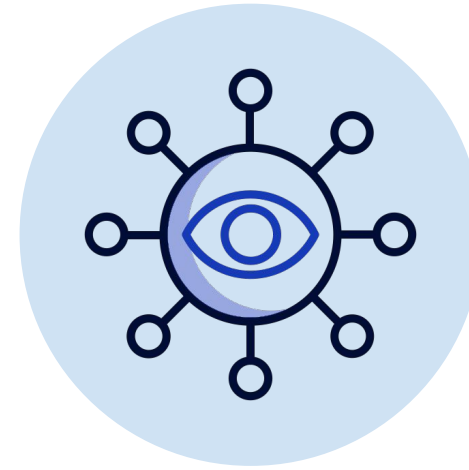
и бесконечно много другого...

# Встроенные и внешние



**Встроенная в язык –**  
стандартная библиотека.

- ★ Доступна из коробки
- ★ Устанавливается вместе с Python
- ★ Достаточно богатый функционал (в отличие от других языков)



**Внешние библиотеки**

- ★ Нужно устанавливать дополнительно
- ★ Для установки внешних библиотек (пакетов, package) – пакетный менеджер **pip**

# Python Standard Library



# Стандартная библиотека

Документация:

<https://docs.python.org/3/library/>

Перевод:

[https://digitology.tech/docs/python\\_3/library/index.html](https://digitology.tech/docs/python_3/library/index.html)



**Встроенные (builtins)** –  
не нужно **импортировать**

- ★ функции – print(), len(), ...
- ★ типы и классы – int, float, str, list, dict, ...



**Импортируемые**

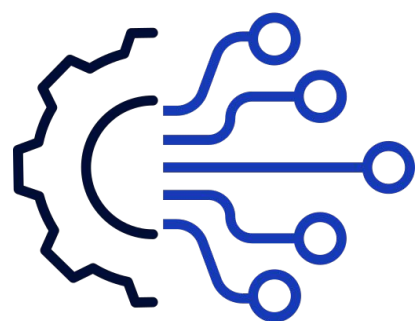
- ★ для работы с операционной системой – os
- ★ для красивой печати переменных и данных – pprint
- ★ для взаимодействия с интернетом – urllib

и многое другое...

# Библиотека **os**

Документация:

<https://docs.python.org/3/library/os.html>



Предоставляет возможность взаимодействия с операционной системой – **кроссплатформенно**, т.е. можно один и тот же код использовать в Windows, macOS и Linux.

## 1 Навигация по операционной системе

- ★ **os.getcwd()** – узнать текущую папку
- ★ **os.listdir(".../path/to/folder")** –  
узнать содержимое папки по пути  
**.../path/to/folder**
- ★ **os.path.isfile(".../path/to/file")** – существует ли  
уже файл по сути **.../path/to/file**

## 2 Объединение путей

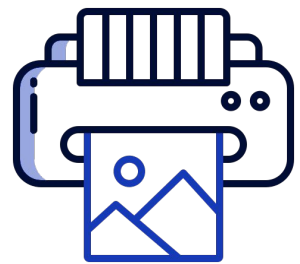
- ★ **os.path.join(".../parent", "child.txt")** –  
выдаст **.../parent/child.txt**

## 3 Создание папок

- ★ **os.mkdir("new\_folder")** – создать папку  
**new\_folder**

# Библиотека **pprint**

Документация: <https://docs.python.org/3/library/pprint.html>



Красивый вывод сложных структур данных в консоль или в файл

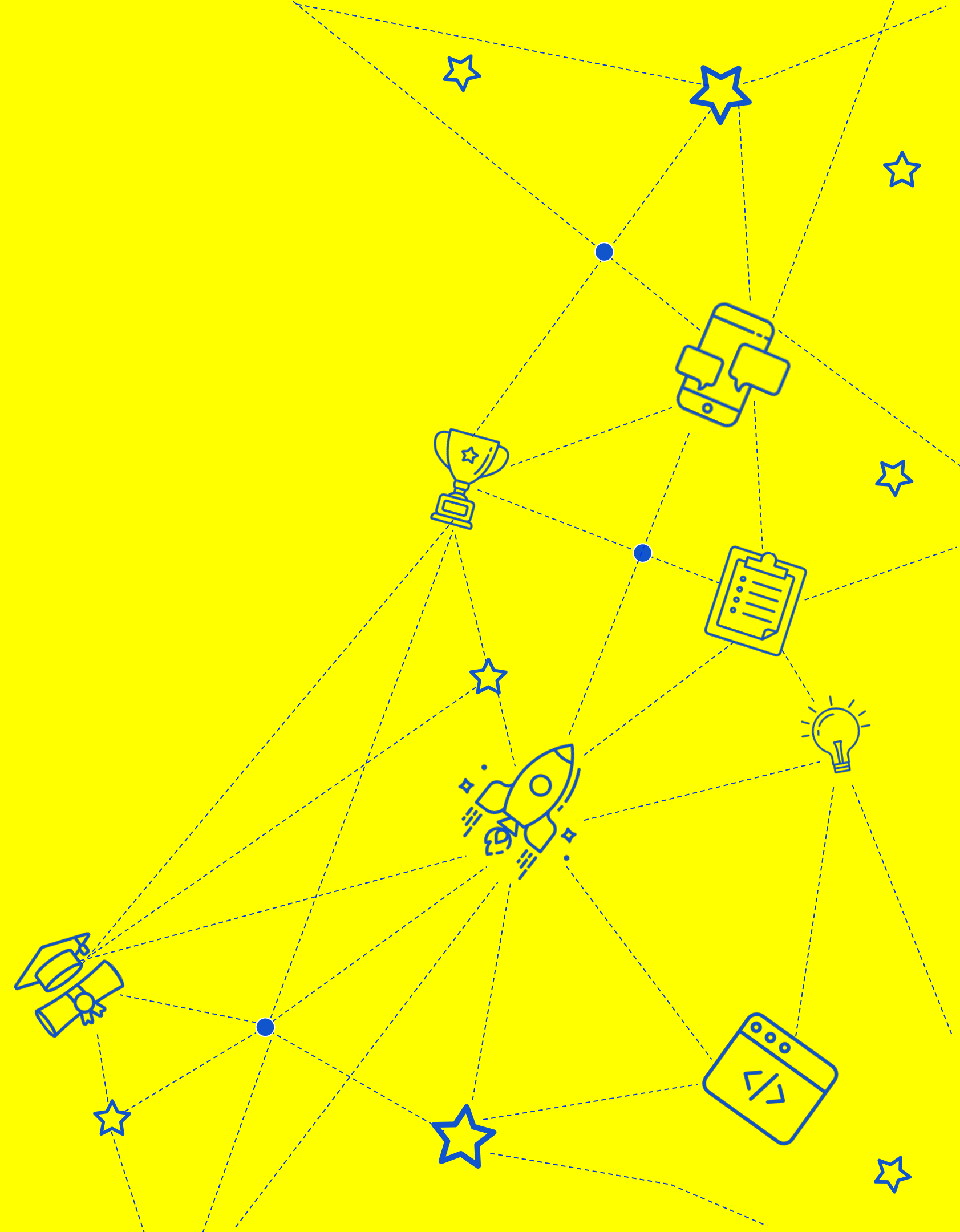
**pprint.pprint(object)** – напечатать object в удобочитаемом формате, пример:

☆ изначальный object = [{"tags": ["green", "suburban", "calm"], "city": "Reading", "population": 34540}, {"tags": ["vibrant", "close", "chill"], "city": "Hampton", "population": 16730}]

☆ **pprint(object):**

```
[{'city': 'Reading',  
  'population': 34540,  
  'tags': ['green', 'suburban', 'calm']},  
 {'city': 'Hampton',  
  'population': 16730,  
  'tags': ['vibrant', 'close', 'chill']}]
```

# Внешние библиотеки

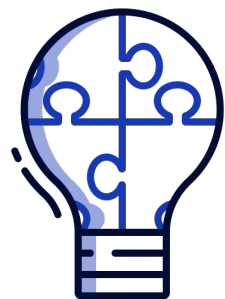




# Пакетный менеджер **pip**

Документация:

<https://pip.pypa.io/en/stable/installation/>



В Anaconda уже установлен

**1** Установка новой библиотеки:

★ **pip install requests**

★ **requests** – имя библиотеки

**2** Обновление существующей библиотеки:

★ **pip install --upgrade requests**

**3** Удаление:

★ **pip uninstall requests**

**4** **pip list** – просмотр установленных пакетов

# Библиотека **requests**

Документация:

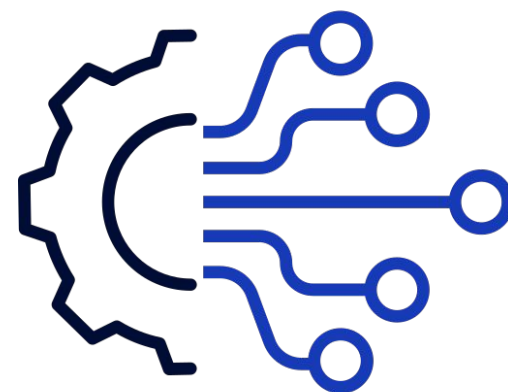
<https://pypi.org/project/requests/>



> **pip install requests**



Проще работать  
чем с **urllib**

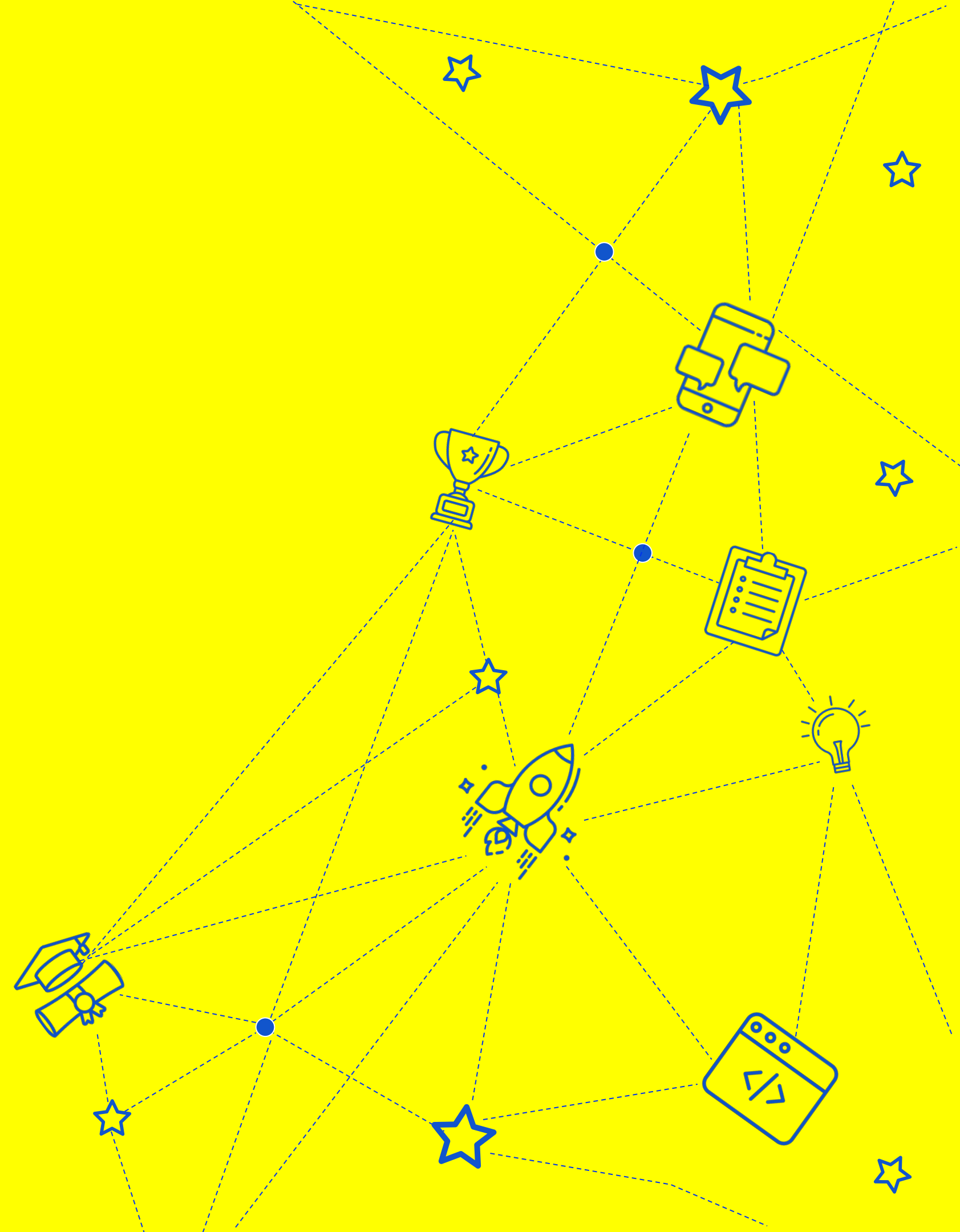


Более  
продвинутый  
и более гибкий

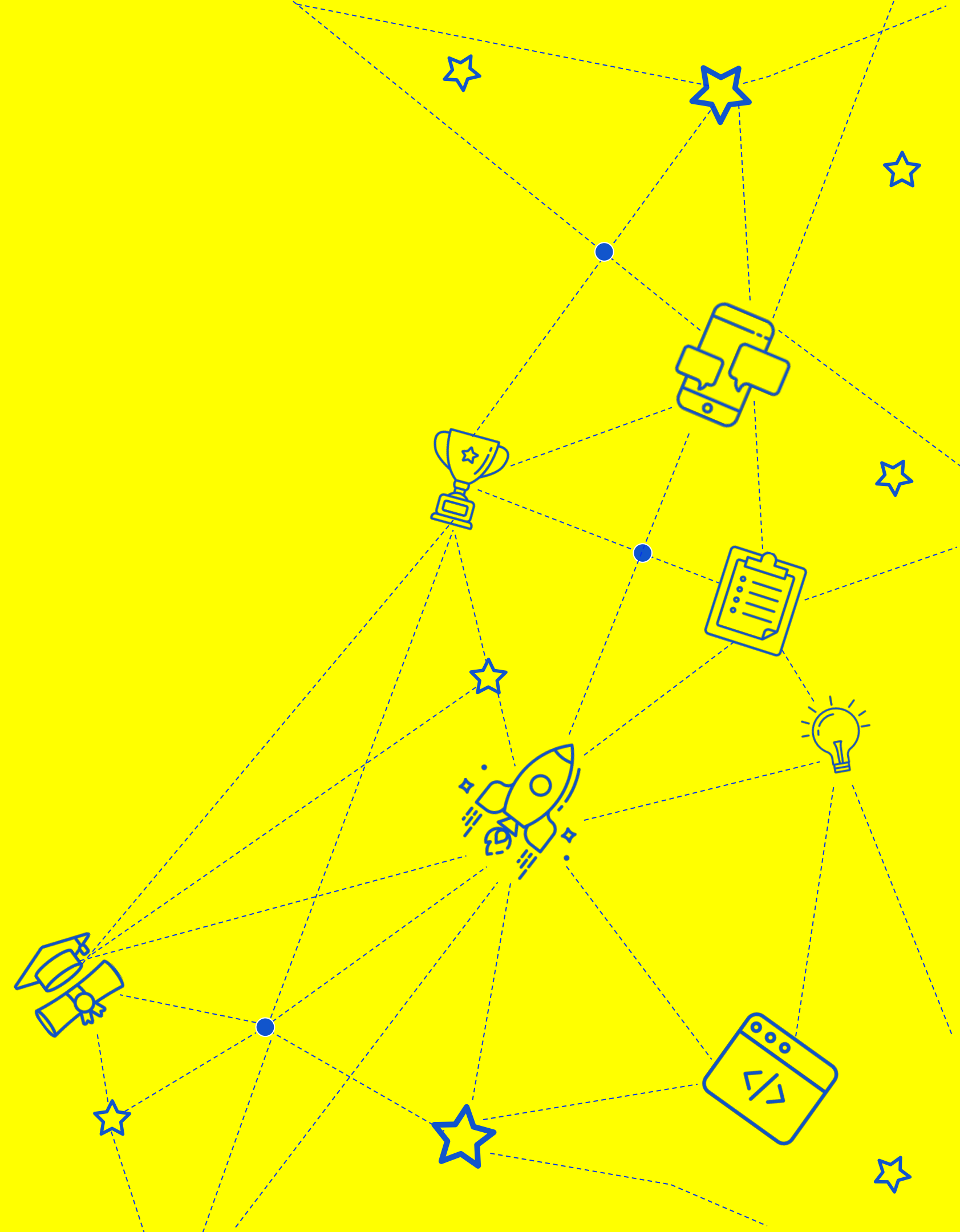


Проще настраивать  
сложные  
HTTP-запросы

# Воркшоп по urllib & requests



# Pandas



# Библиотека **pandas**

Документация: <https://pandasguide.readthedocs.io/en/latest/>

На русском: [http://pandas.geekwriter.ru/getting\\_started/index.html](http://pandas.geekwriter.ru/getting_started/index.html)



## > **pip install pandas**

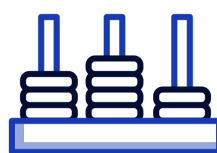
Большая библиотека (фреймворк) для работы с данными.



Чтение и запись данных различных форматов (csv, excel, sql, json, ...)



Фильтрация, поиск и модификация данных



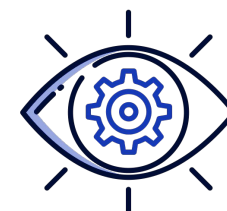
Группировка и подсчет агрегирующих функций



Конкатенация, слияние и джойн



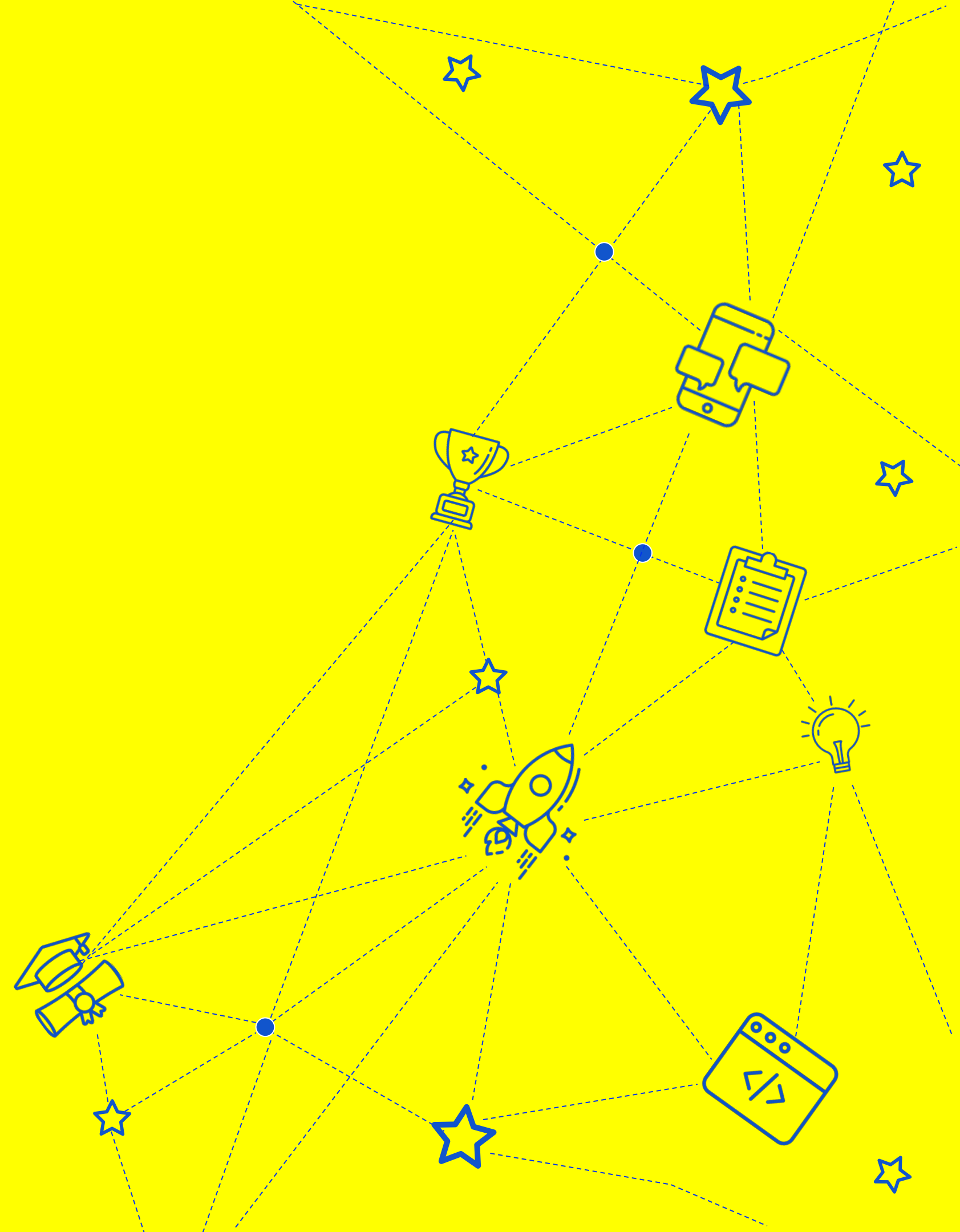
Анализ временных рядов



Визуализация

и многое другое...

# Воркшоп по pandas



# Домашнее задание



1

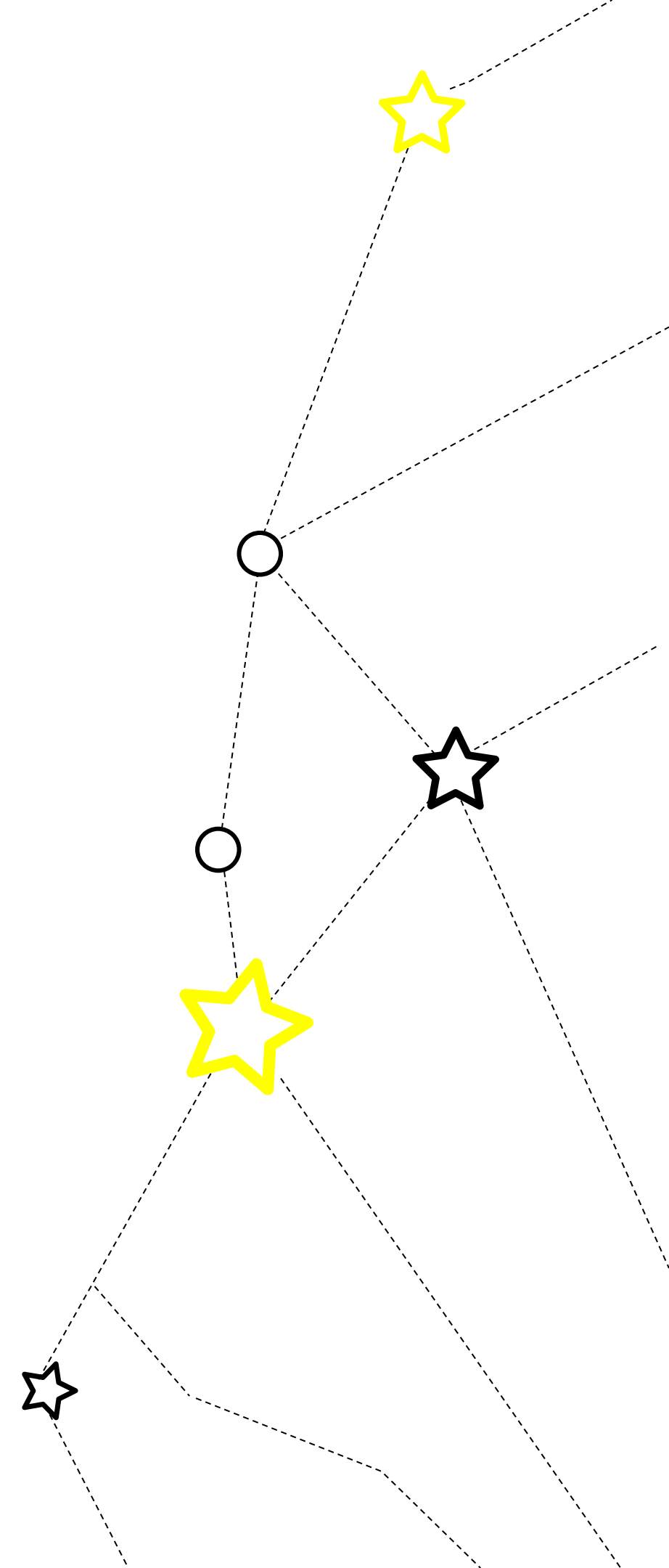
Домашнее задание будет в форме jupyter notebook

2

Нужно будет проанализировать данный датасет в Pandas

3

И ответить на несколько вопросов на основе данных





# СПАСИБО ЗА ВНИМАНИЕ

**Жамшид Рузиев**



<https://www.linkedin.com/in/ruziev/>



@veizur



<https://www.youtube.com/channel/UCq3Ygsydbd4YOMyGCvtpNDA>