

# Переменные и типы данных. Условные операторы

# Содержание урока

- ★ Установка и настройка окружения
- ★ Переменные
- ★ Условные операторы



# ИВАН ГРОМОВ

**Старший разработчик в одной  
из FAANG-компаний**

ex-Yandex, Cian, Lamoda

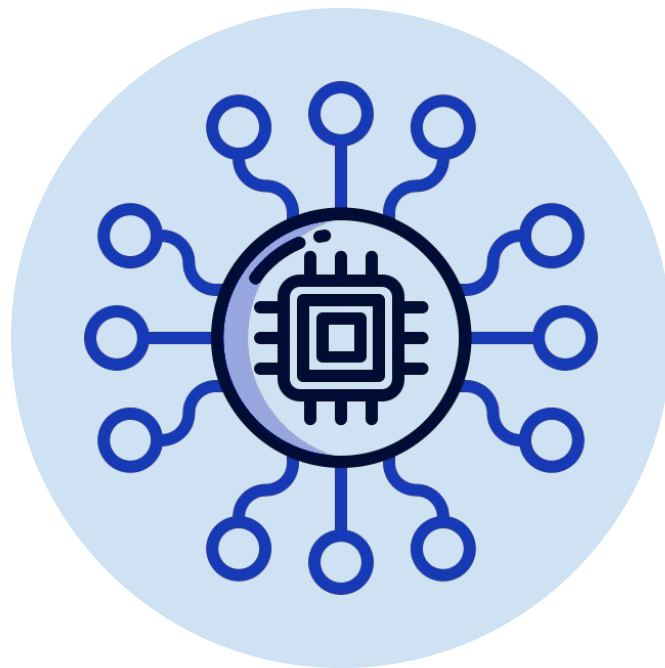
- Развиваю операционную систему для шлемов виртуальной реальности
- Делал Алису и Яндекс.Диалоги
- Занимался аналитикой и ML в Cian

# Установка и настройка окружения

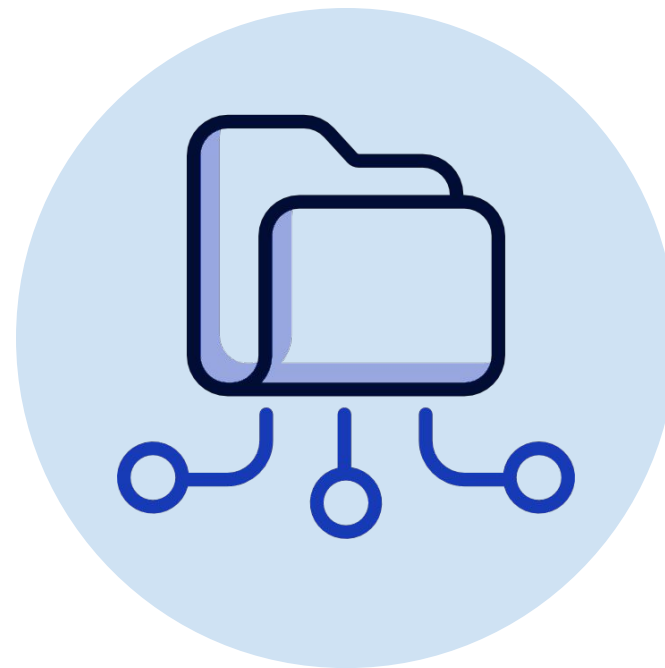


# Установка и настройка окружения

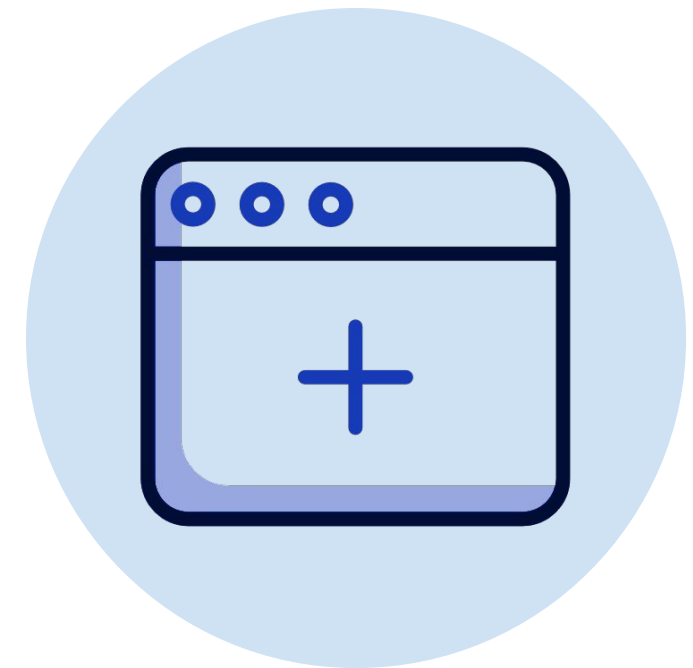
Из чего состоит интерактивное окружение:



**Интерпретатор  
Python**



**Менеджер  
пакетов pip**

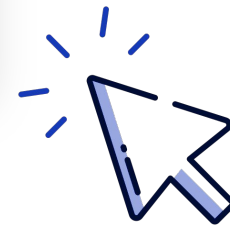


**Интерактивная  
среда Jupyter –**  
позволяет писать код  
в браузере

# Установка Python

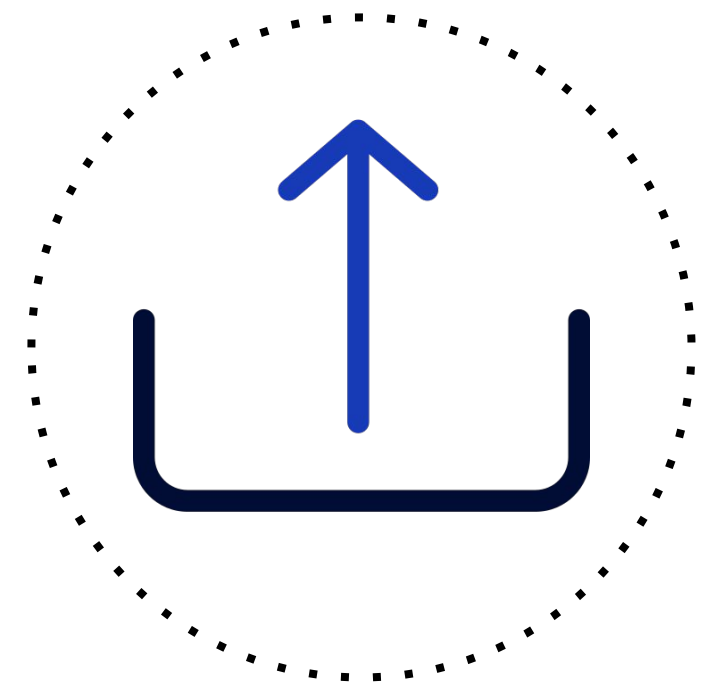
- 1 Скачайте инсталлятор со страницы

[Download Python](#)



- 2 Запустите инсталлятор и дождитесь конца установки

- 3 Для проверки запустите в «Командной строке» команду `python` – в результате должен запуститься интерпретатор в интерактивном режиме



# PIP – менеджер пакетов

**Пакет** – это другое название для дополнительной библиотеки.

Пакеты устанавливаются командой:

```
pip install <название пакета>
```



# Установка Jupyter

**Jupyter** – это интерактивная среда для разработки, работающая в браузере.

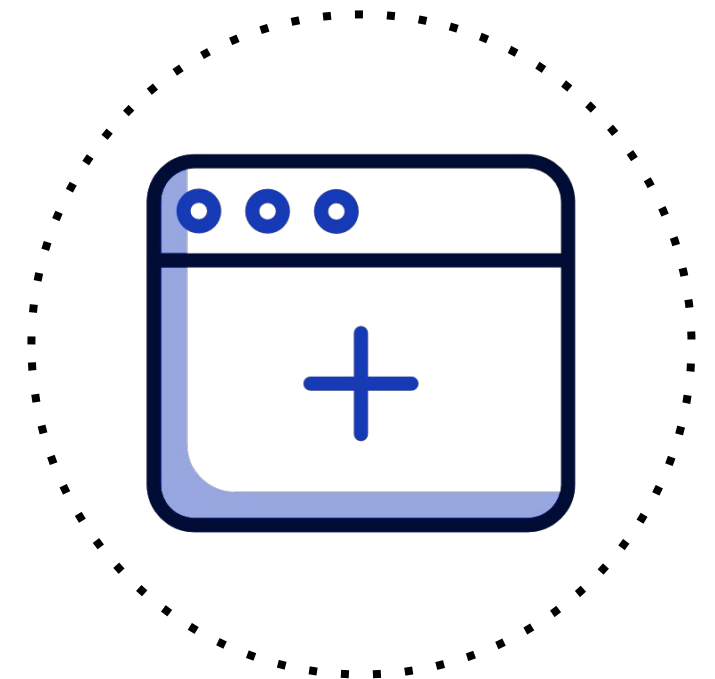
**1** Откройте «Командную строку»

**2** Выполните команду:

```
pip install jupyter
```

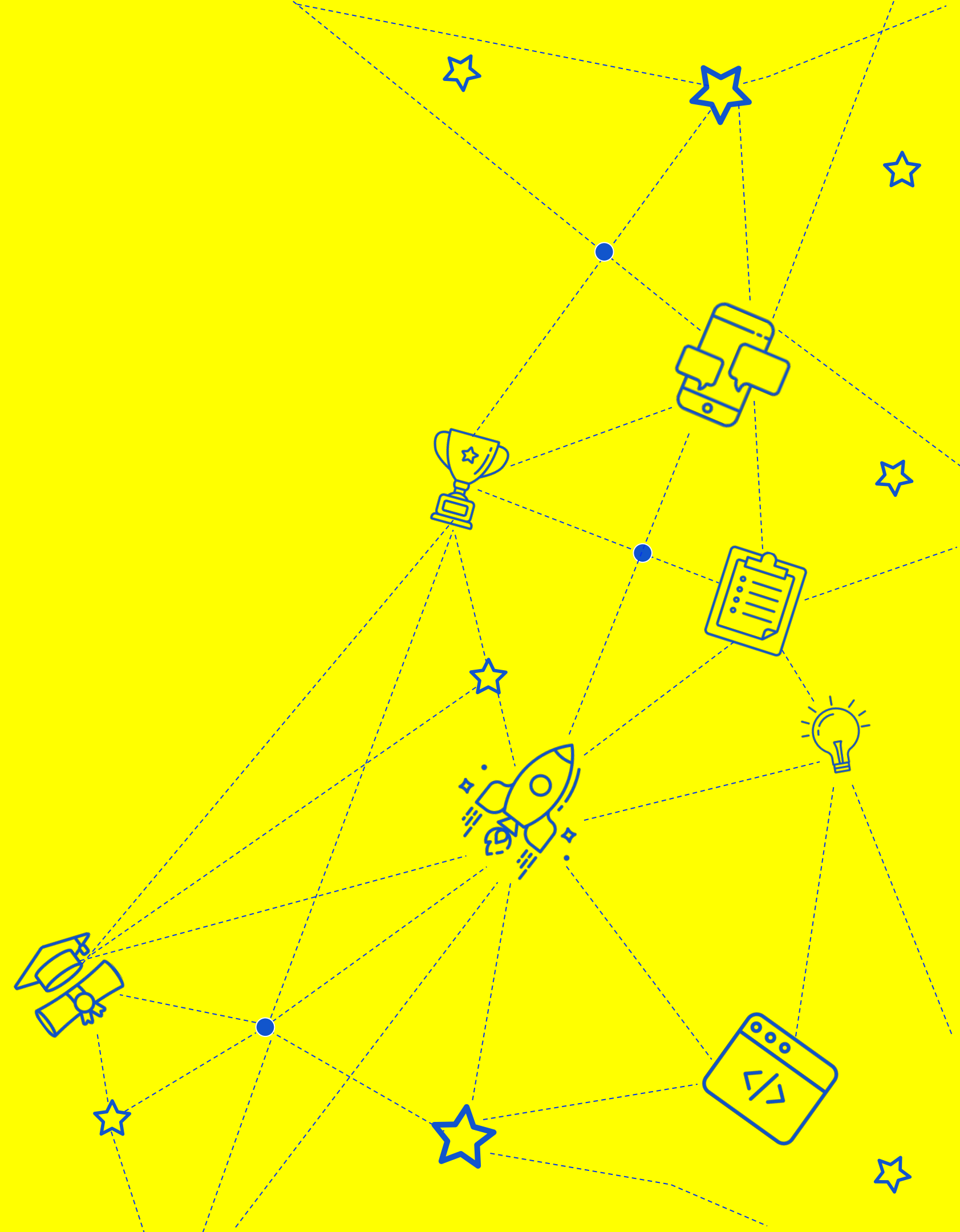
**3** Запустите **jupyter** командой:

```
jupyter notebook
```





# Переменные



# Что такое переменная?

**Переменная** – это «контейнер» в памяти программы, хранящий определенное значение (число, строку или более сложный объект).

Примеры объявления переменных:

- ★ `x = 10`
- ★ `pi = 3.14`
- ★ `hello = 'Привет!'`



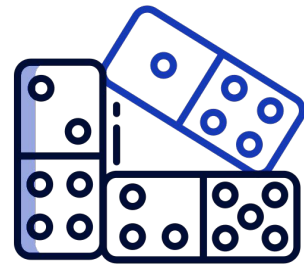
# Типы данных

Переменная может хранить разные объекты – число, строку, набор объектов (коллекцию).

Сегодня мы изучим **базовые типы данных**:



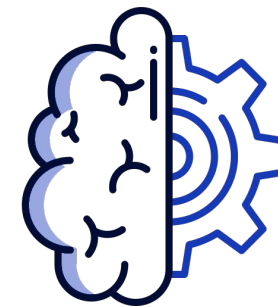
**целые  
числа**



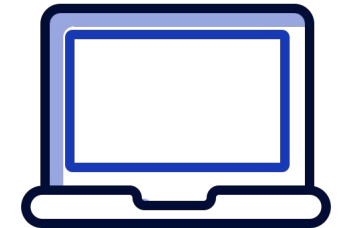
**дробные числа**  
(числа  
с плавающей  
точкой)



**строки**



**логические  
значения**



**None**

# Численные типы

Числа в Python представлены двумя типами:  
**целочисленным (int)** и **дробным (float)**

★  $x = 10$

★  $x = 10.0$  или  $x = 10.$

Над числами можно совершать арифметические операции – складывать, вычитать, умножать и делить.

$$a = b + c * d / e - f$$



# Строки

**Строка** – это произвольная последовательность символов, в том числе пустая.

★ `empty_string = ''`

★ `s = 'Hello world!'`

Строки можно складывать, **но нельзя вычитать:**

```
>>> print('Hello ' + 'world')  
Hello world
```

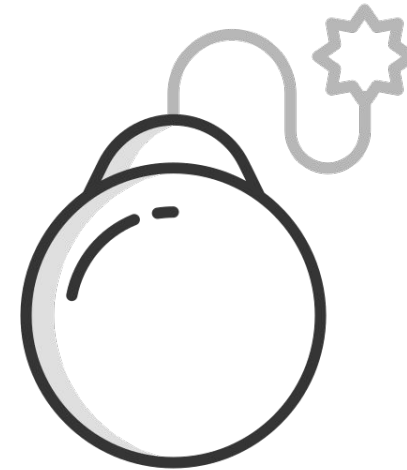
```
>>> print('Hello ' - 'world')  
TypeError: unsupported operand type(s) for -:  
'str' and 'str'
```

# Логический тип данных

Логический тип данных может принимать одно из двух значений – **True (истина)** или **False (ложь)**



**a = True**



**b = False**

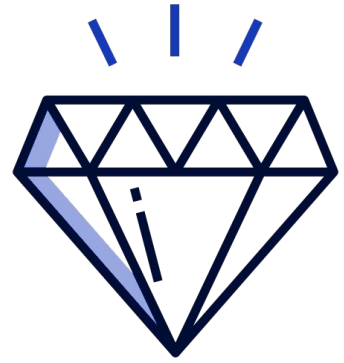
# Логическое **ИЛИ**



Выражение **a or b** принимает значение **True**, если значение хотя бы одного из аргументов ИСТИННО.

<b>a</b>	<b>b</b>	<b>a or b</b>
False	False	False
False	True	True
True	False	True
True	True	True

# Логическое И



Выражение **a and b** принимает значение **True**, если значение всех аргументов истинно.

a	b	a and b
False	False	False
False	True	False
True	False	False
True	True	True



# Логическое отрицание

Выражение **not a** меняет значение аргумента на противоположное.

a	not a
False	True
True	False



# None

**None** – это специальный тип данных, обозначающий «ничего».

**Основное назначение этого типа** – обозначить переменную с необязательным значением.

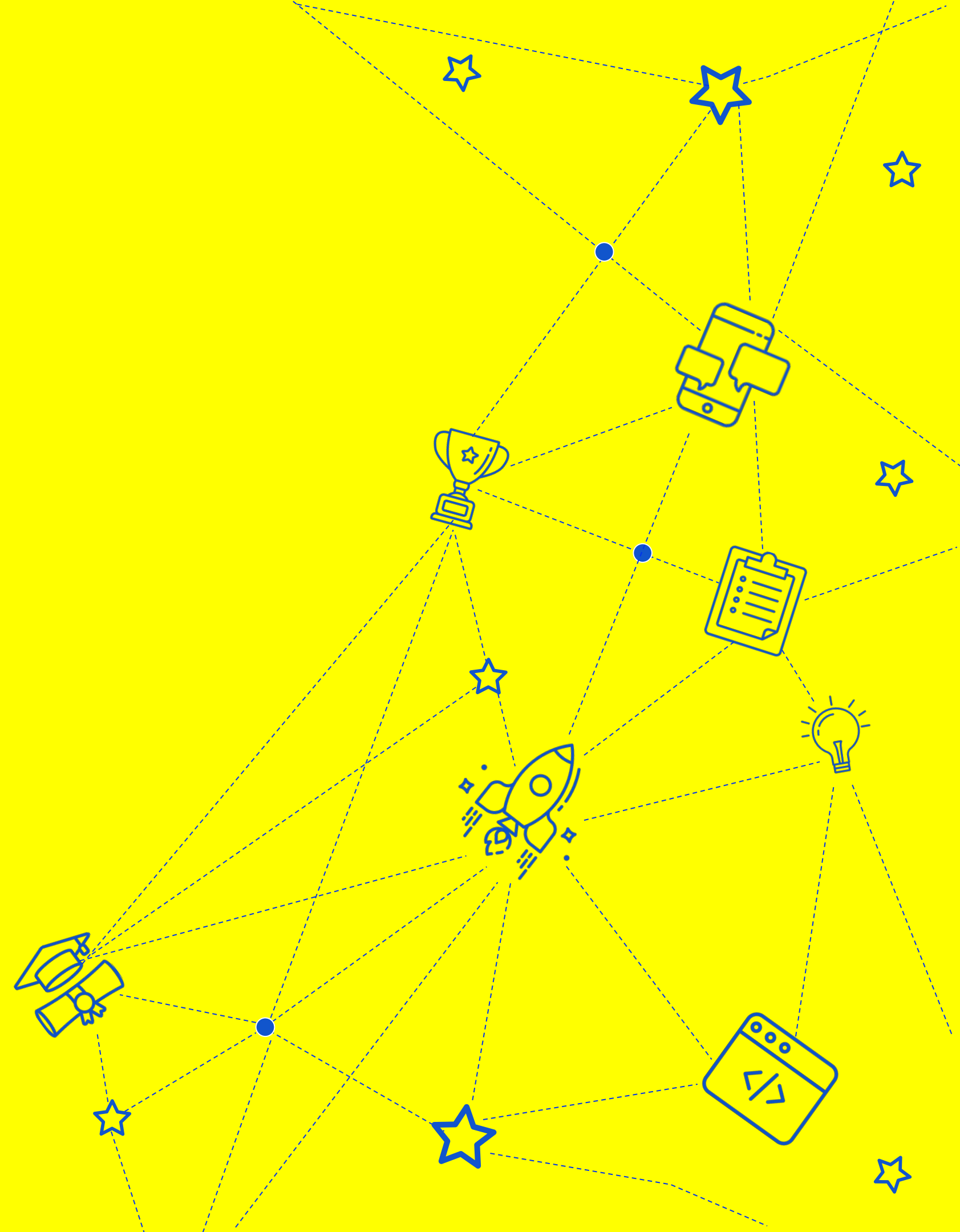
`a = None` # a не имеет никакого значения

...

`a = 10`



# Условные операторы



# If-Else

**Оператор if-else** позволяет изменить логику программы в зависимости от истинности условия.

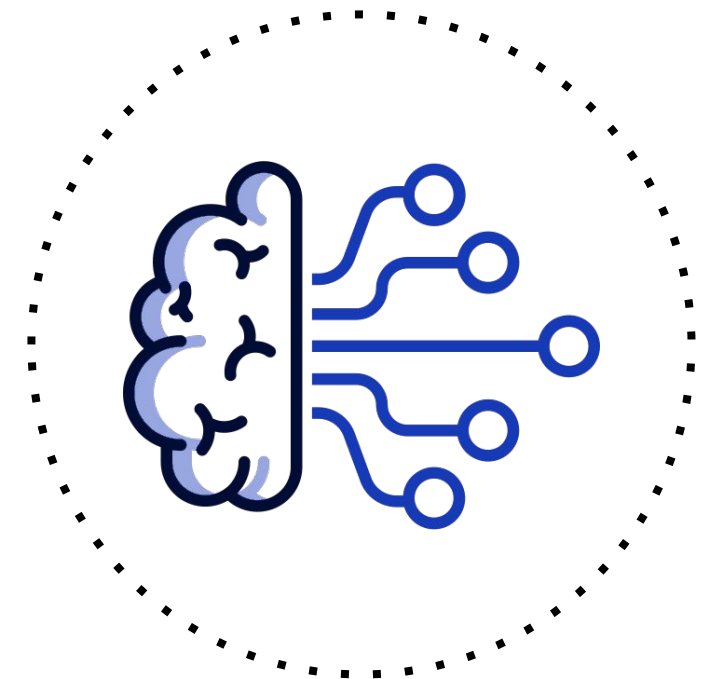
```
if a > 0:  
    print('Positive')  
elif a < 0:  
    print('Negative')  
else:  
    print('Zero')
```



# If-Else

В условии так же можно написать логическое выражение:


```
if a > 0 or a < 0:  
    print('Non-zero')  
else:  
    print('Zero')
```



# If-Else

Любую переменную можно проверить на истинность.

Значение **False** принимают:

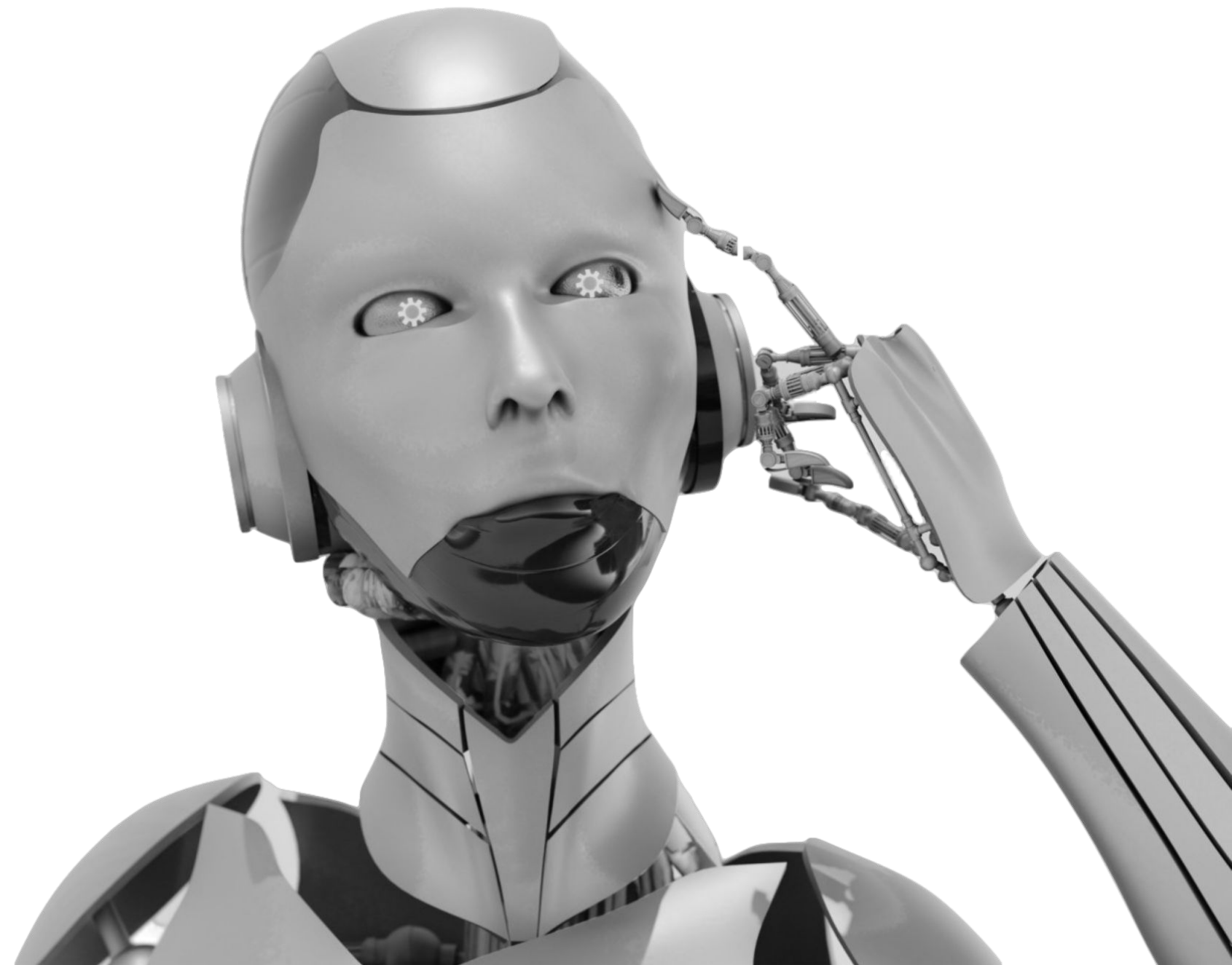
- ☆ Численные переменные, равные 0
- ☆ Логические переменные со значением False
- ☆ Пустые строки
- ☆ None
- ☆ Пустые словари, списки и множества
-  Остальные значения конвертируются в значение **True**

# Объявление переменной блоком **if-else**

Значение переменной можно  
определить в зависимости от условия:

```
a = 100
```

```
x = 1 if a > 10 else 2
```



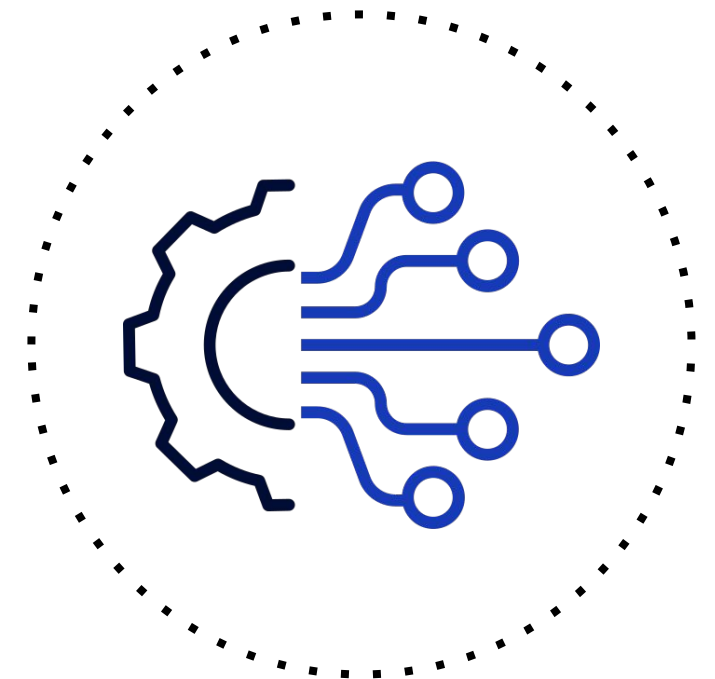


# Пример: обработка команд

Каждая команда задается **массивом**, в котором **первый элемент** – это название команды, **а все остальные элементы** – это аргументы.

## Список команд:

- ★ **['show', file1, file2, ...]** – показать все файлы из списка
- ★ **['remove', file1, file2, ...]** – удалить все файлы из списка
- ★ **['delete', file1, file2, ...]** – то же, что remove
- ★ **['remove', '--ask', file1, file2, ...]** – удалить файлы, но запросить подтверждение перед удалением





# Обработка команд при помощи **if-else**

```
if command[0] == 'show':  
    print('Список файлов: ')  
elif (command[0] == 'remove' or command[0] == 'delete') and '--ask' in command:  
    print('Пожалуйста, подтвердите удаление файлов')  
elif command[0] == 'remove' or command[0] == 'delete':  
    print('Удаляю файлы')  
else:  
    print('Неизвестная команда')
```

# Match-case

```
match <variable>:  
  case <condition 1>:  
    ...  
  case <condition 2>:  
    ...  
  case <condition 3>:  
    ...
```



# Обработка команд при помощи **match-case**

match command:

```
case ['show', files]:
```

```
    print('Список файлов: ')
```

```
case ['remove' | 'delete', *files] if '--ask' in files:
```

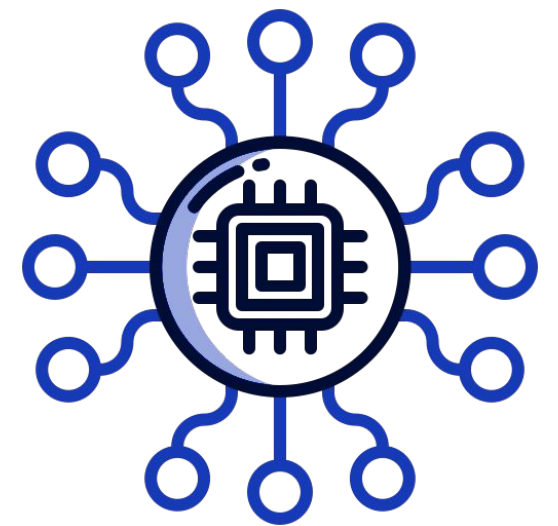
```
    print('Пожалуйста, подтвердите удаление файлов')
```

```
case ['remove' | 'delete', *files]:
```

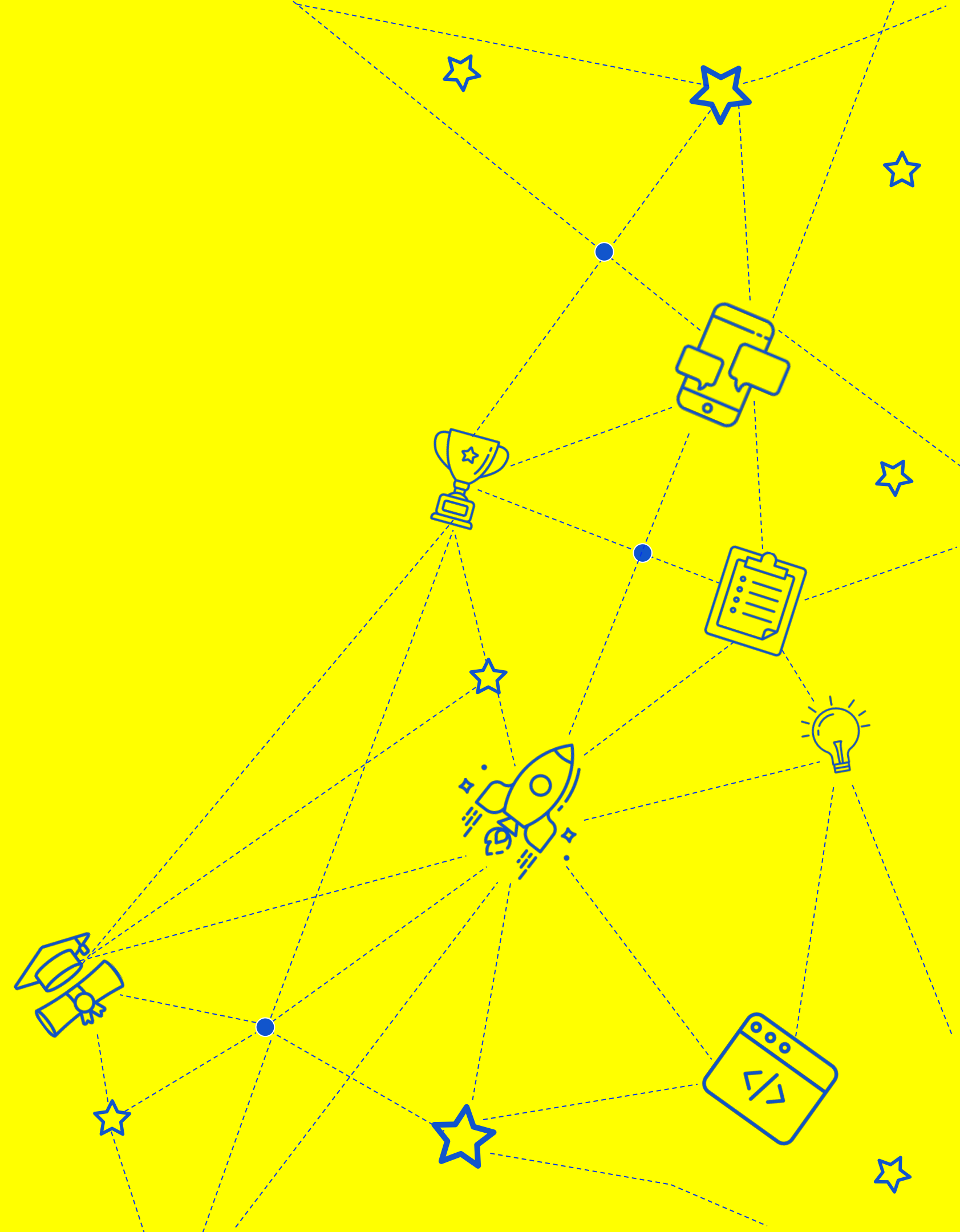
```
    print('Удаляю файлы')
```

```
case _:
```

```
    print('Неизвестная команда')
```



# Домашнее задание



# Домашнее задание

Напишите ячейку в **Jupyter Notebook**, которая решает квадратное уравнение с коэффициентами **a**, **b** и **c**.

$$a \cdot x^2 + b \cdot x + c = 0$$

$$D = b^2 - 4 \cdot a \cdot c$$

$$D > 0 \Rightarrow x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$$

$$D = 0 \Rightarrow x = \frac{-b}{2a}$$

$$D < 0 \Rightarrow \text{решений нет}$$



# Домашнее задание

**a = 1**

**b = -5**

**c = 6**

```
import math
```

```
D = ...
```

```
...
```

```
x1 = (-b - math.sqrt(D)) / (2*a)
x2 = (-b + math.sqrt(D)) / (2*a)
print(f'x1 =', x1)
print(f'x2 =', x2)
```

```
...
```

```
x = ...
print(f'x =', x)
```

```
...
```

```
print('Решений нет')
```

