

ОСНОВЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Содержание урока

- ☆ Языки программирования
- ☆ Типы данных и переменные
- ☆ Условные операторы
- ☆ Функции



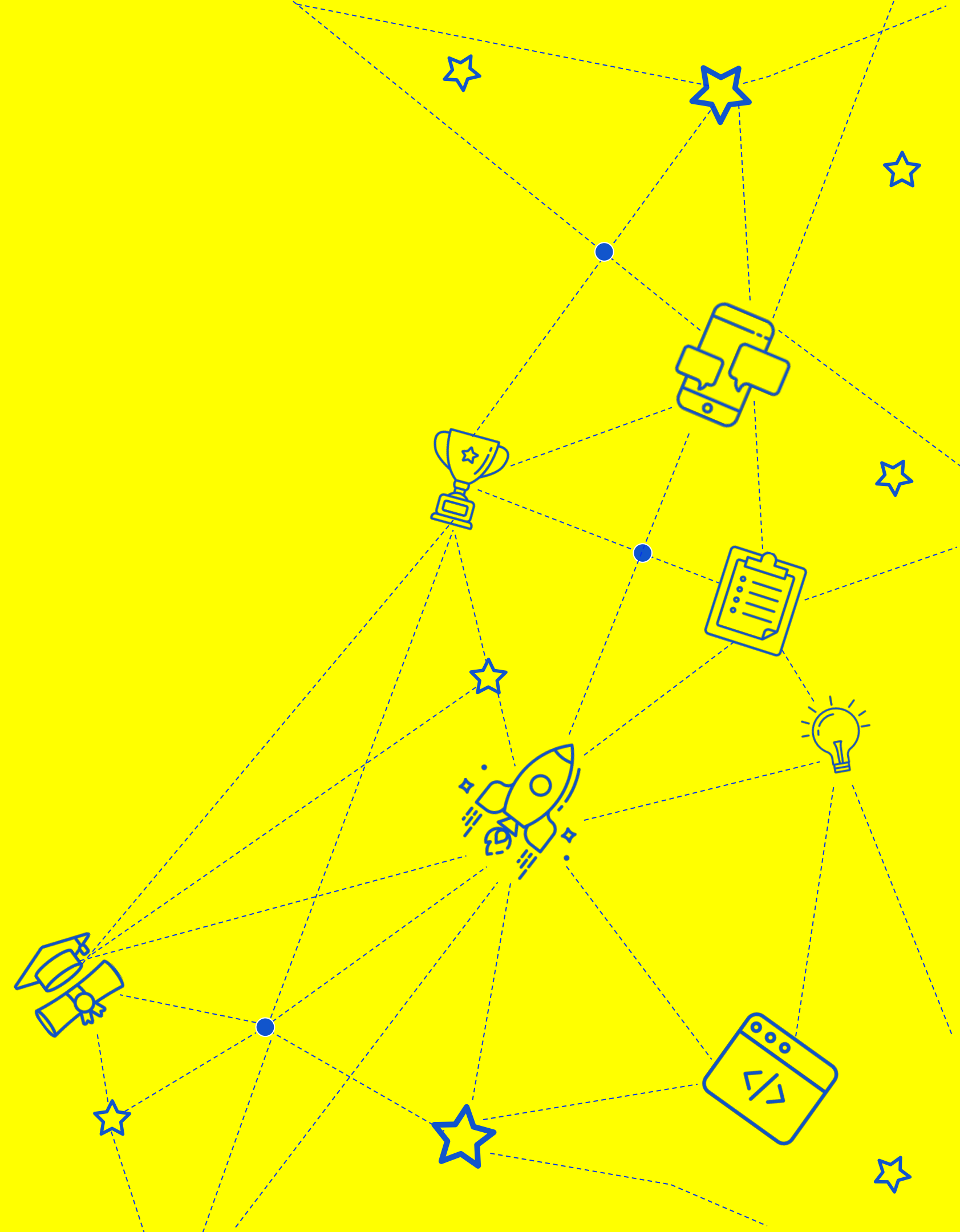
АРТЕМ МАКСИМОВ

Software Engineer, Meta

ex-Software Engineer, Yandex

- Работаю в сфере виртуальной и дополненной реальности в Meta
- Разрабатывал Яндекс Метрику
- Работаю с разными языками программирования

Языки программирования



Языки программирования



Популярные языки
программирования

★ JavaScript

★ Python

★ C++

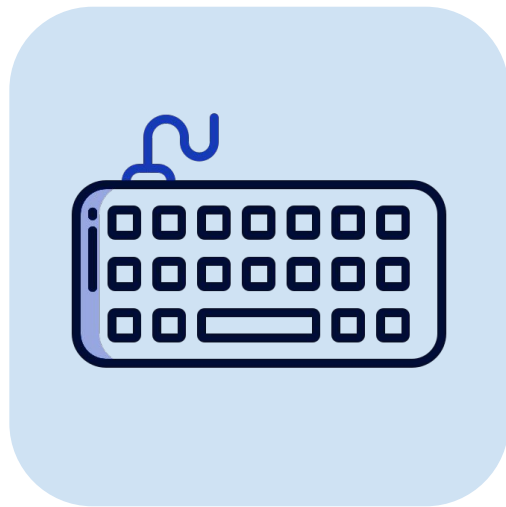
★ Java

★ C#

...

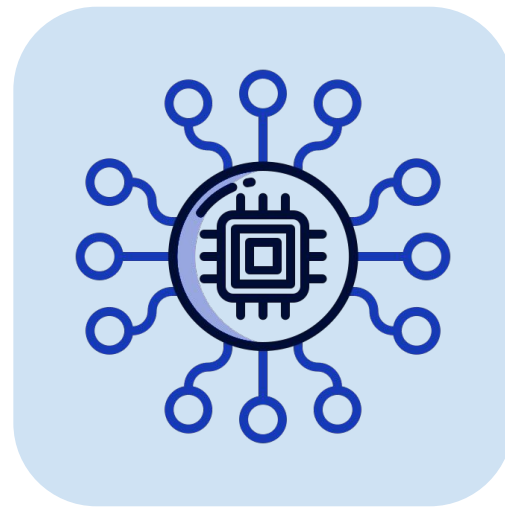
Input / Output

Как работает программа:



Ввод данных

input



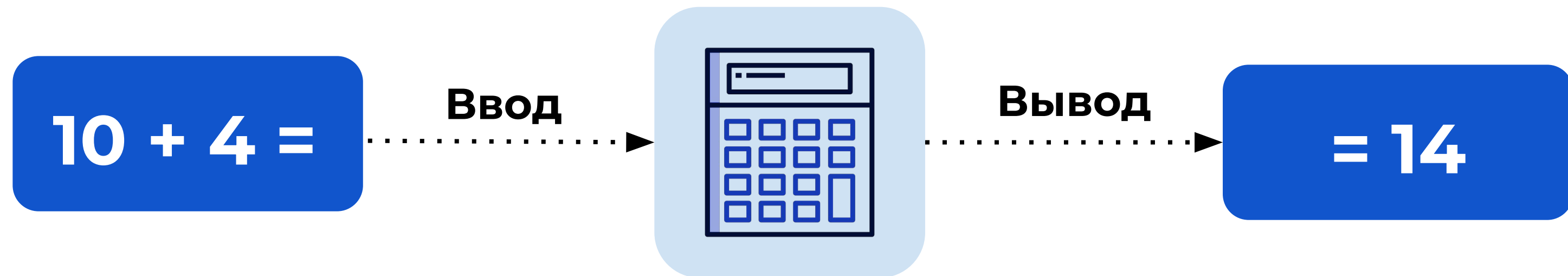
**Вычисления
на основе
данных**



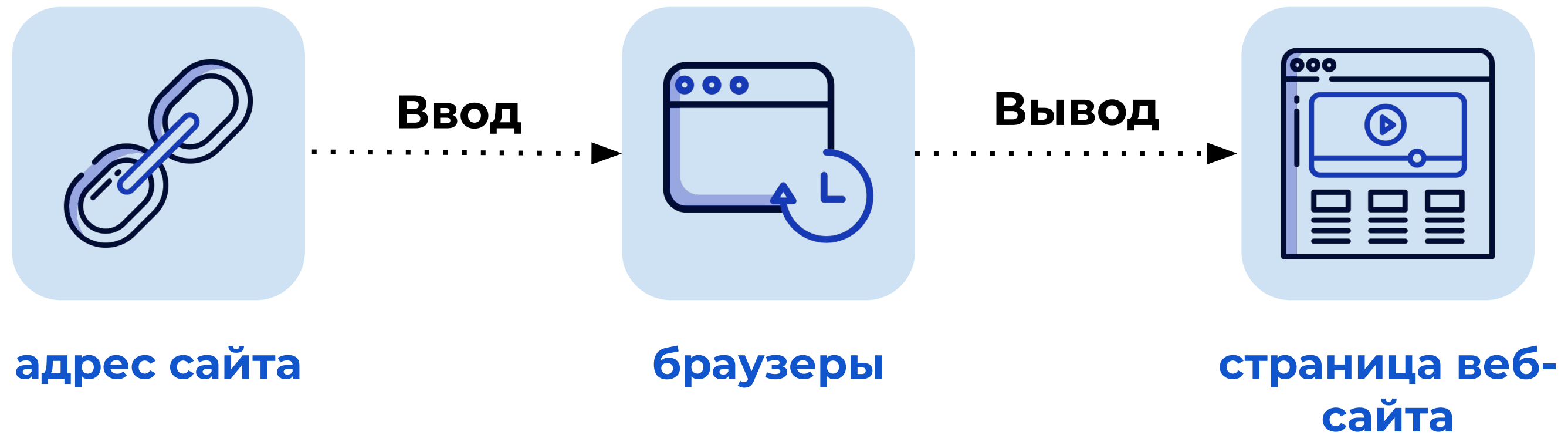
Вывод

output

Input / Output



Input / Output



Как выглядит код

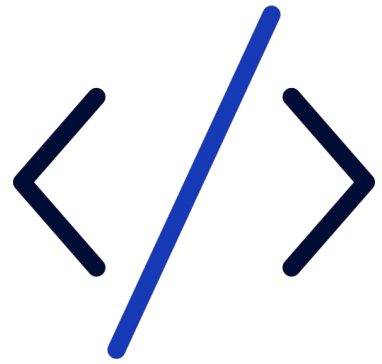
```
const factorialOfNumber = number =>  
  number < 0  
    ? (() => {  
      throw new TypeError('No negative numbers please');  
    })()  
    : number <= 1  
      ? 1  
      : number * factorialOfNumber(number - 1);
```

// EXAMPLES

```
factorialOfNumber(4); // 24
```

```
factorialOfNumber(8); // 40320
```

Синтаксис



Синтаксис – это набор правил и ключевых слов, с помощью которых вы можете писать код на определенном языке программирования.



Проблема

**Как программисты,
пишущие на разных
языках, понимают
друг друга?**



Псевдокод



Человеческий язык:

Введите возраст

Если возраст меньше 18,
то напиши «Доступ запрещен»

Если возраст больше или равен 18,
то напиши «Доступ разрешен»

Псевдокод

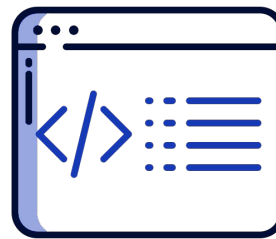


Человеческий язык:

Введите возраст

Если возраст меньше 18,
то напиши «Доступ запрещен»

Если возраст больше или равен 18,
то напиши «Доступ разрешен»



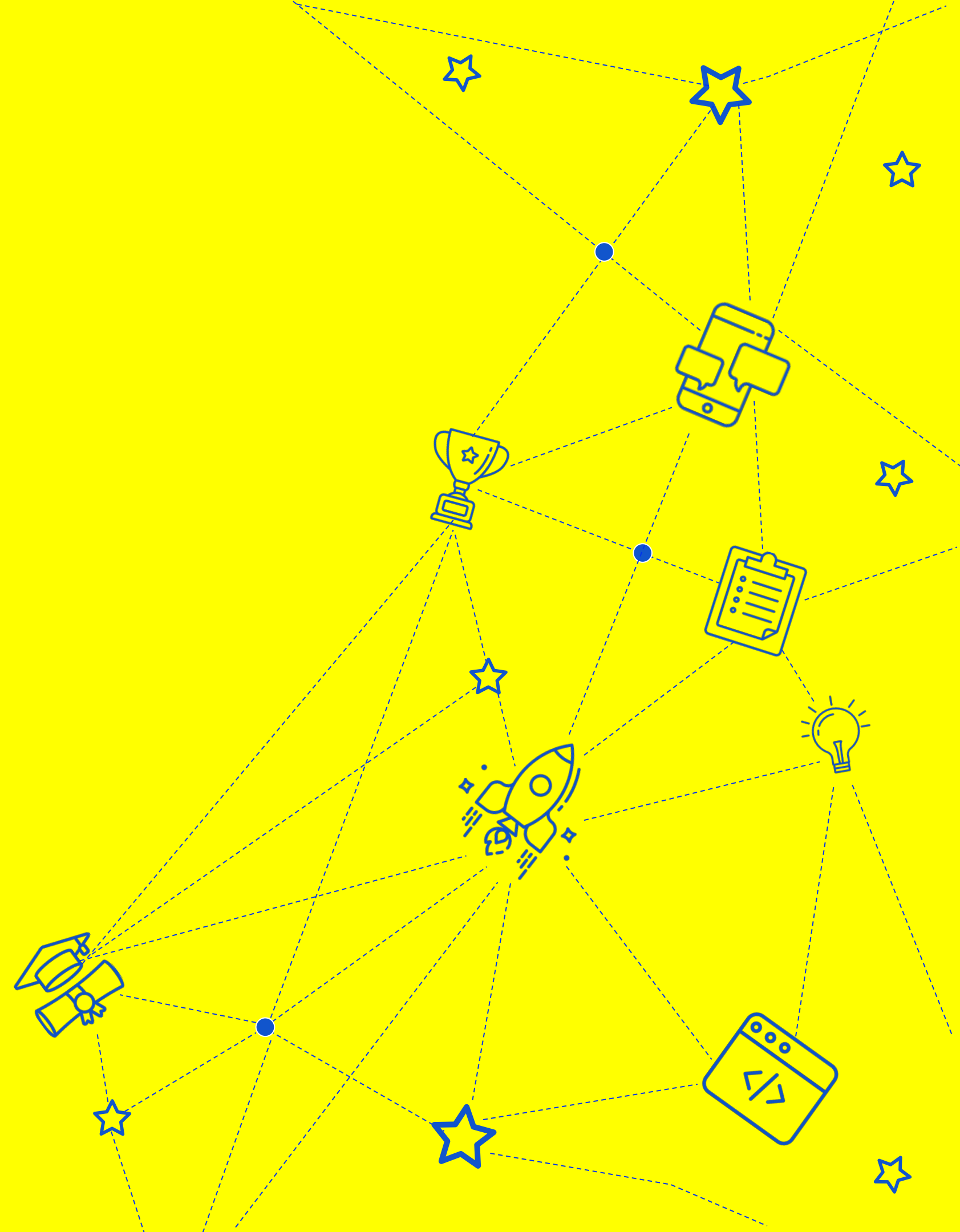
Похожий на язык программирования:

```
age = input();
```

```
if (age < 18) {  
    print ("Доступ запрещен");  
}
```

```
if (age >= 18) {  
    print ("Доступ разрешен")  
}
```

Типы данных и переменные



Данные

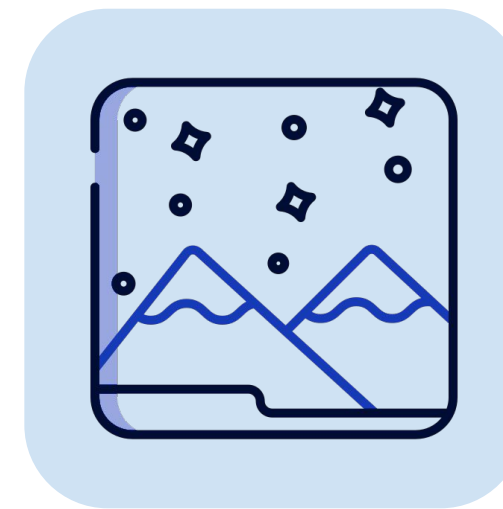
Языки программирования оперируют данными, которые хранятся в памяти компьютера. Например:



**Имя
пользователя**

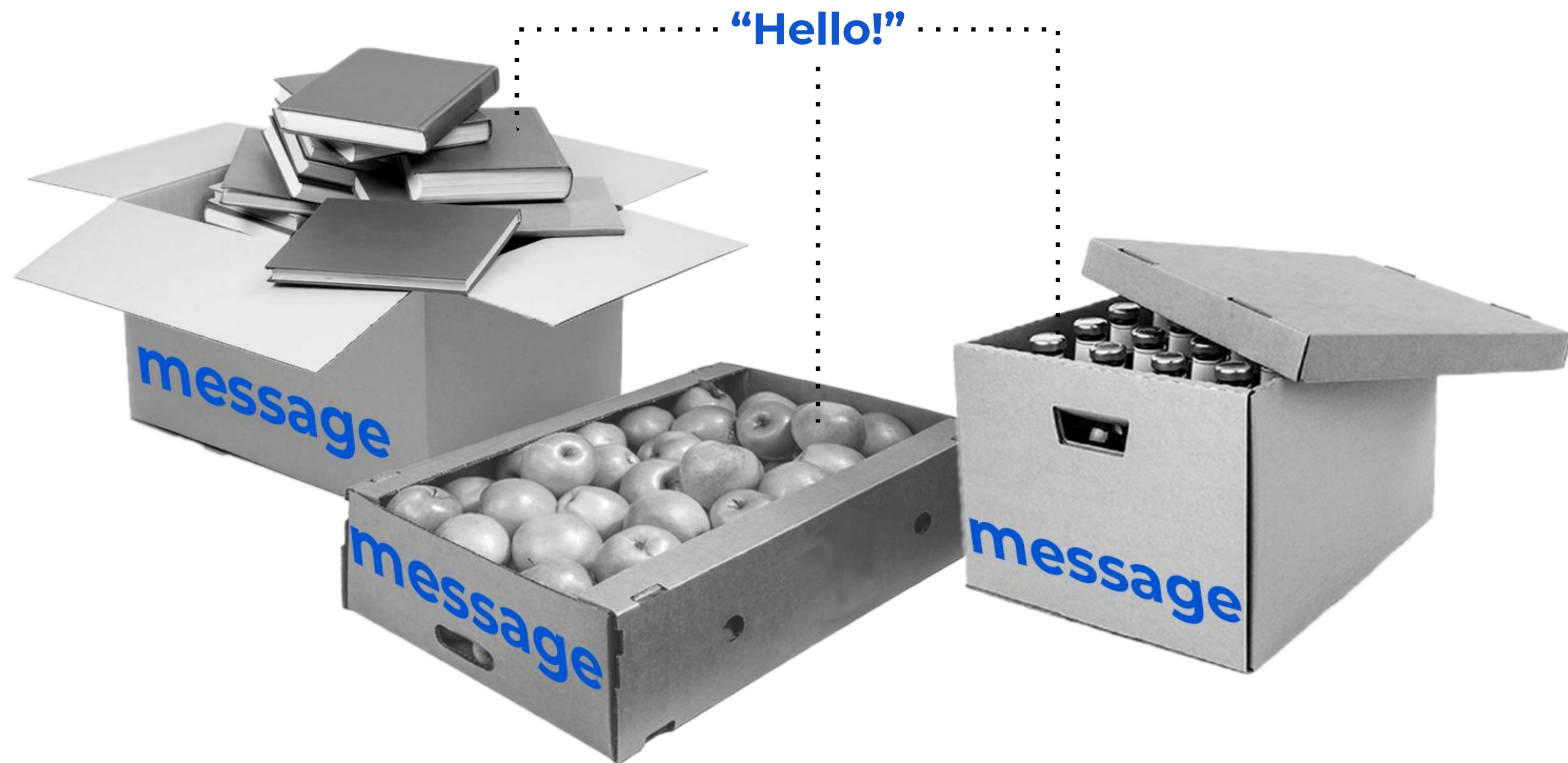


**Баланс
банковского
счета**

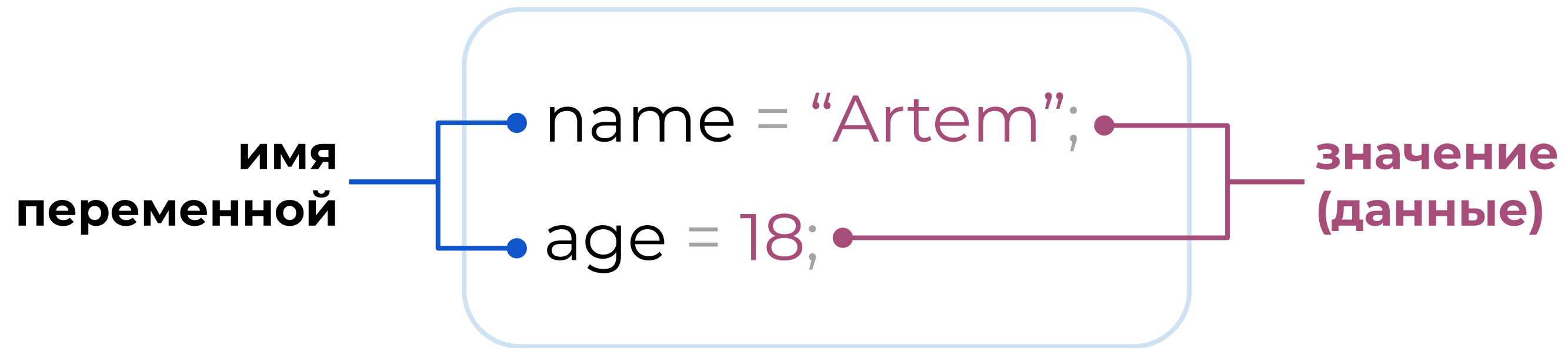


Фотография

Переменные



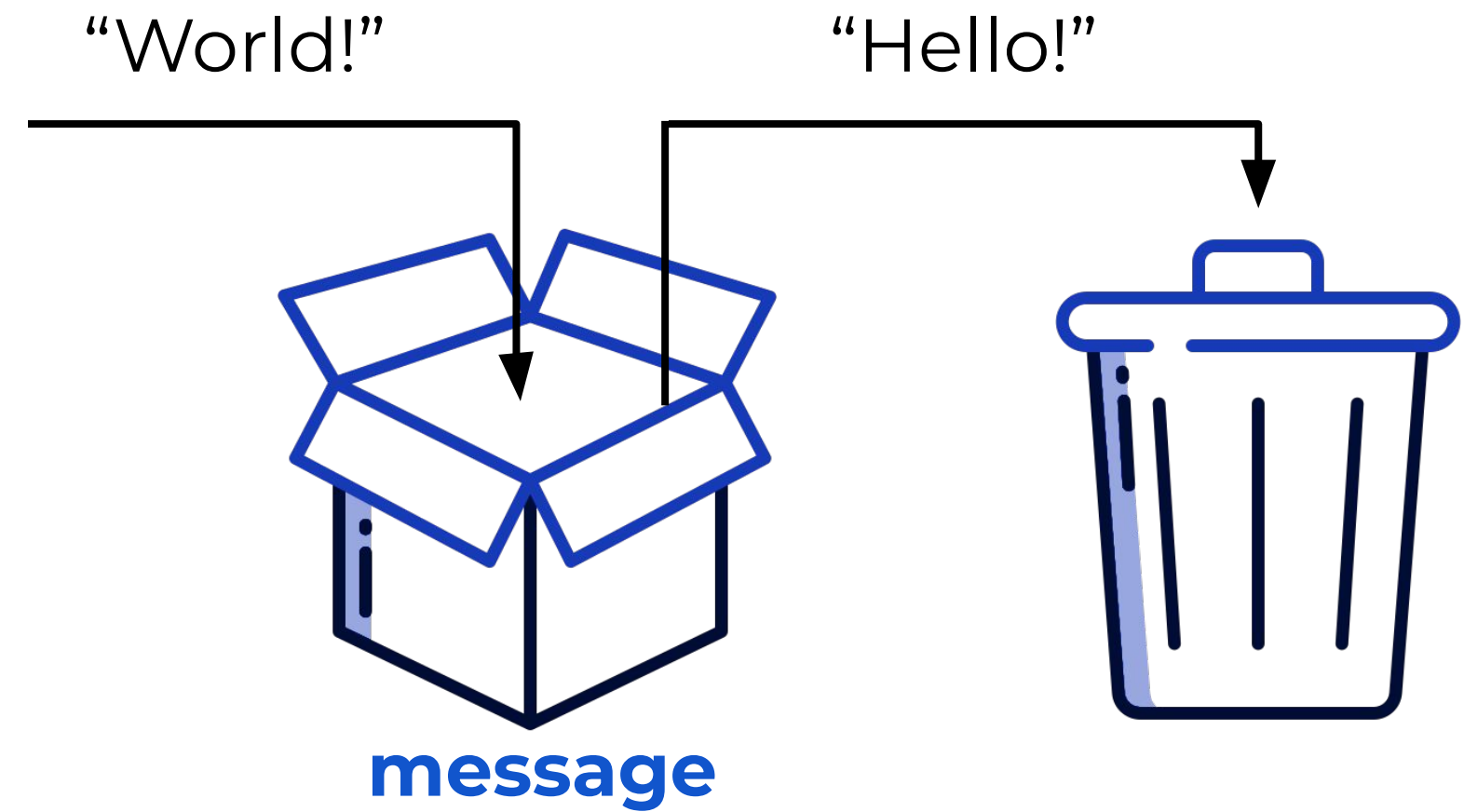
Синтаксис переменной



Оператор присвоения =

```
animal = "cat";  
animal = "dog";
```

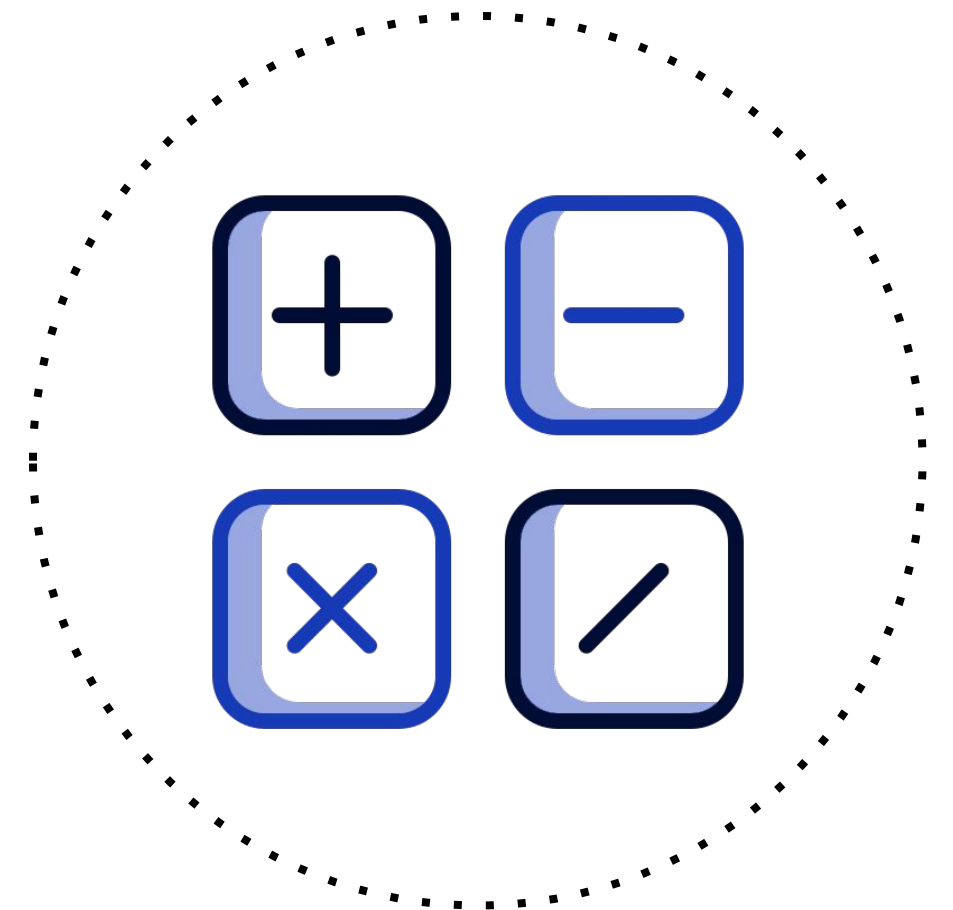
Чему равна переменная **animal**?



Математические операции

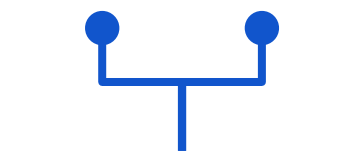
```
a = 10;  
b = 4;  
c = a + b;
```

Чему равно **c**?



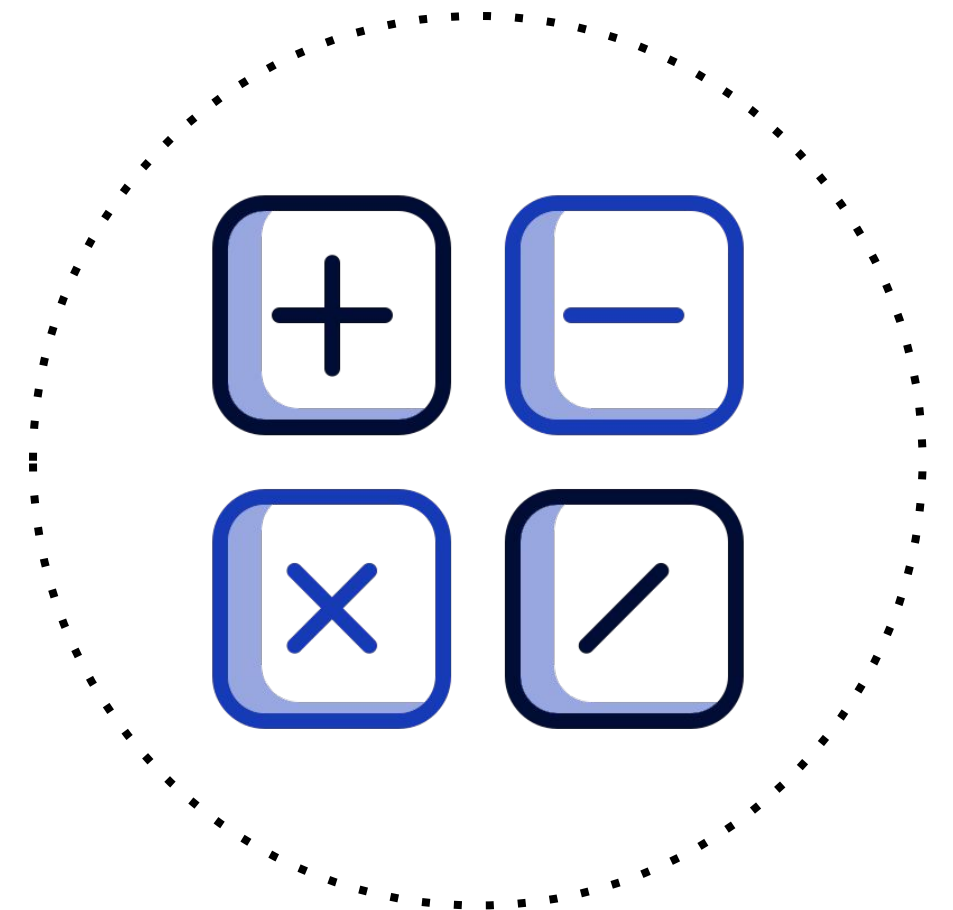
Математические операции

```
name = "Ivan";  
surname = "Petrov";  
fullname = name + surname;  
fullname = name + " " + surname;
```



пробел

Чему равно **fullname**?

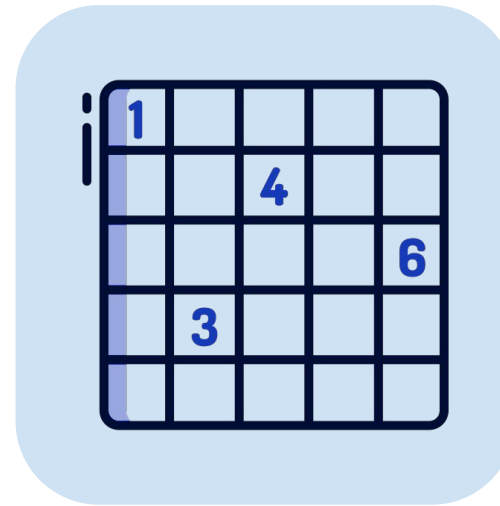


Типы переменных



string

строка



number

число



boolean

**логический
(булевый)
тип**

Тип переменной



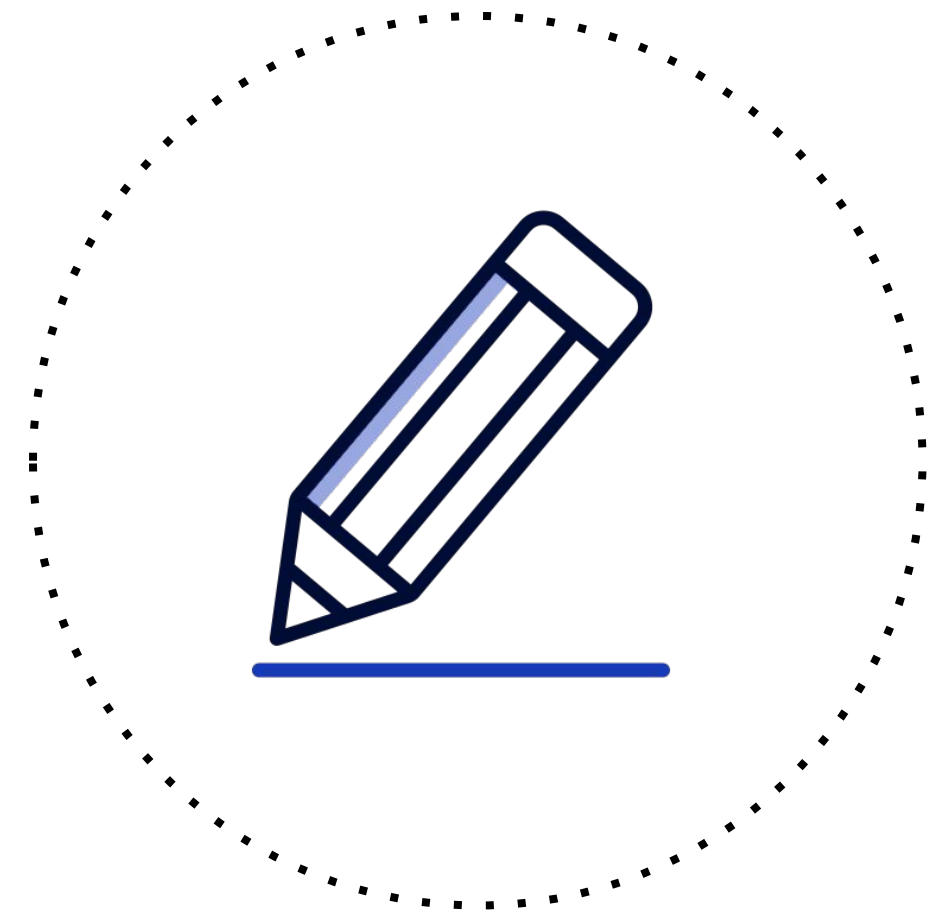
Тип String

```
string name = "Иван";
```

```
string animal = "Кот";
```

```
string book = "Война и мир";
```

```
string about = "Меня зовут " + name + ", у меня есть "  
+ animal + ". Моя любимая книга — " + book;
```

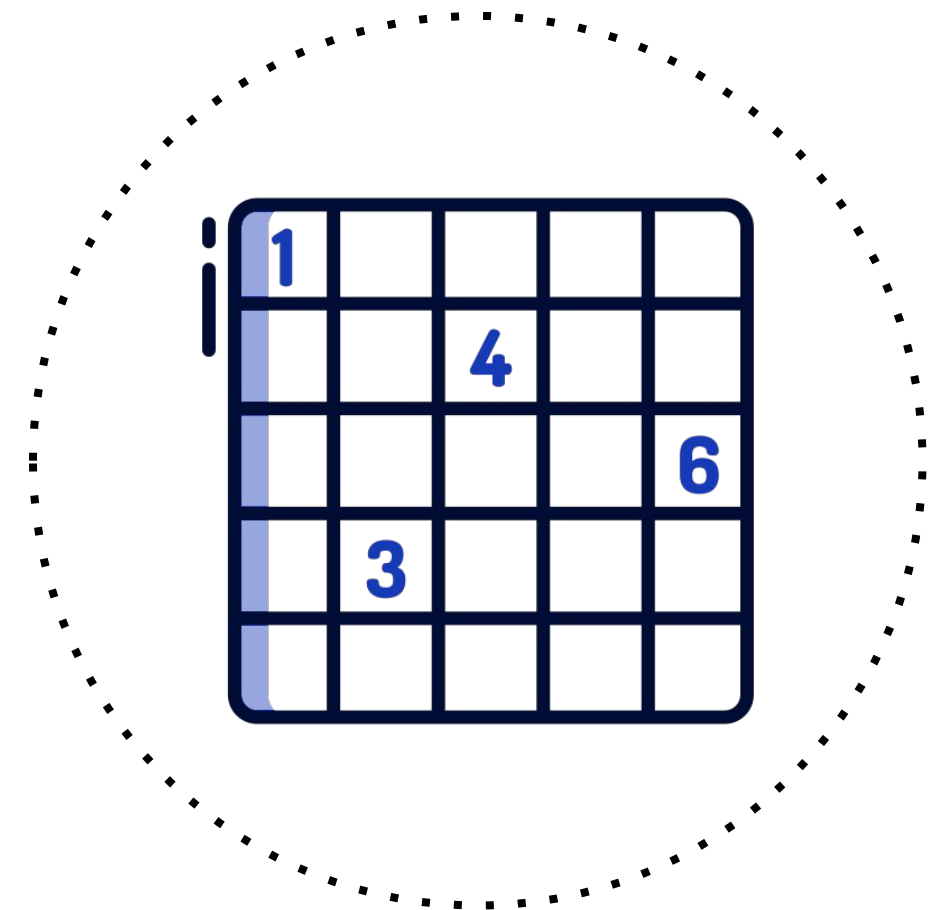


Тип Number

number age = 20;

number year = 2010;

number pi = 3.14;



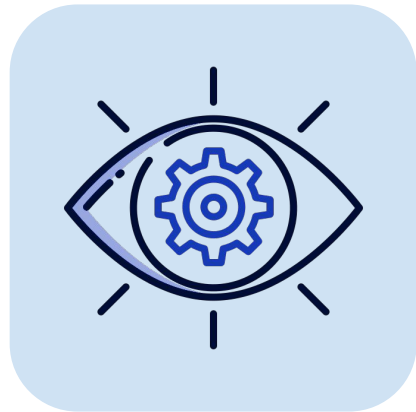
Тип boolean

```
boolean sun = true;
```

```
boolean rain = false;
```



Типизация



Явная

Предполагает, что программист должен указывать типы всех переменных и функций, которые объявляет.

- ★ C++
- ★ Java



Неявная

Язык программирования сам расставит типы на основе данных.

- ★ JavaScript
- ★ Python

Пример

JavaScript

```
var name = "Artem";  
var age = "18";
```



Работает

Пример

JavaScript

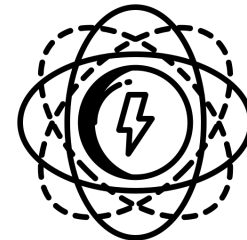
```
var name = "Artem";  
var age = "18";
```



Работает

Java

```
String name = "Artem";  
Int balance = "18";
```



Ошибка

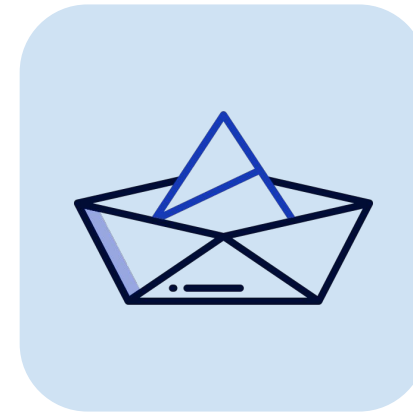
Типизация



Сильная

Не позволяет выполнять выражения с различными типами, например сложить число и строку.

- ★ Java
- ★ Python



Слабая

Выполняет множество неявных преобразований автоматически, например преобразует число в строку при сложении с другой строкой.

- ★ JavaScript

Пример

JavaScript

```
var name = "Artem";  
var age = 18;  
var info = name + age;
```



Работает

Пример

JavaScript

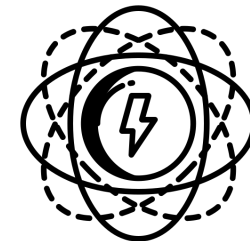
```
var name = "Artem";  
var age = 18;  
var info = name + age;
```



Работает

Python

```
name = "Artem"  
age = 18  
info = name + age
```



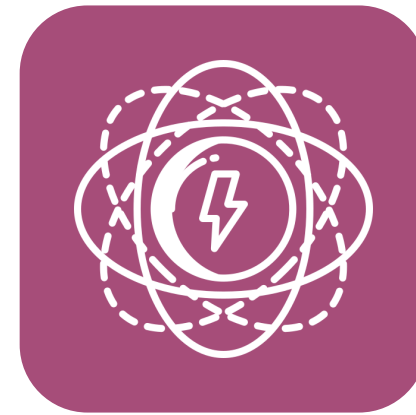
Ошибка

Наименование переменных



Хорошо:

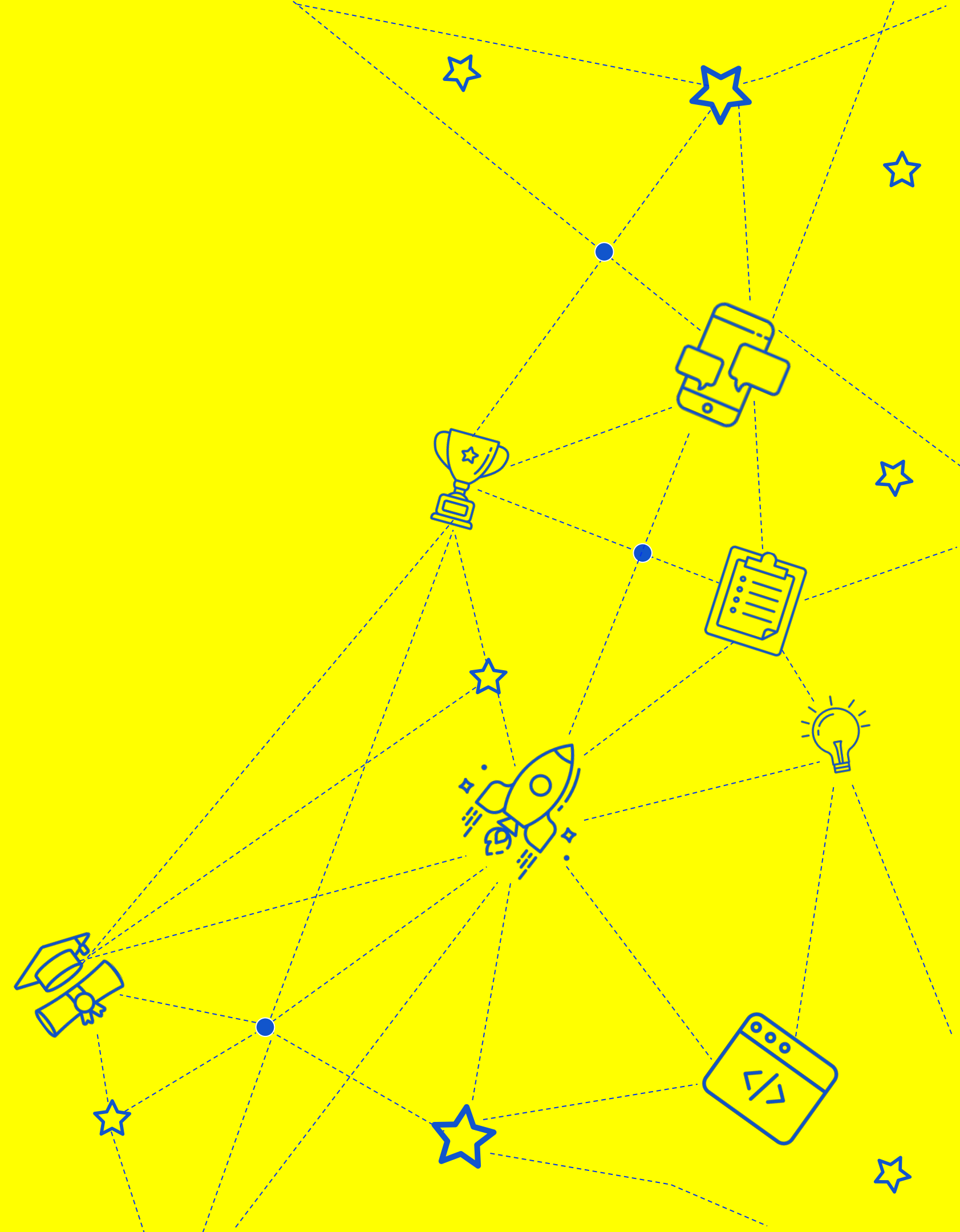
- ★ name
- ★ surname
- ★ message
- ★ myBestFriend
- ★ my_best_friend



Плохо:

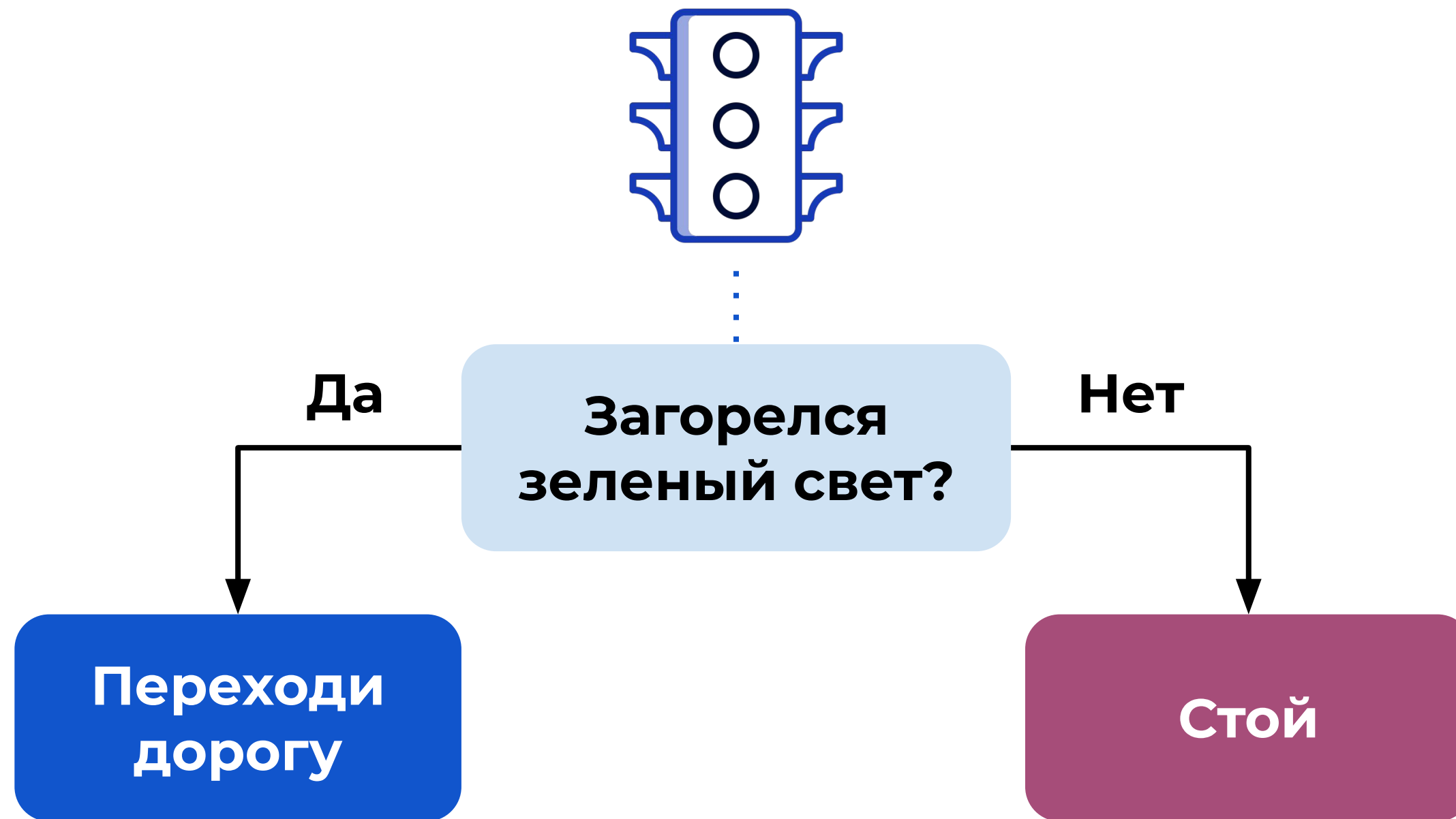
- ★ my best friend
- ★ adgregeswffsd
- ★ 10
- ★ \$&&#

Условные операторы



Блок схема

Блок-схема – это графическое описание алгоритма.

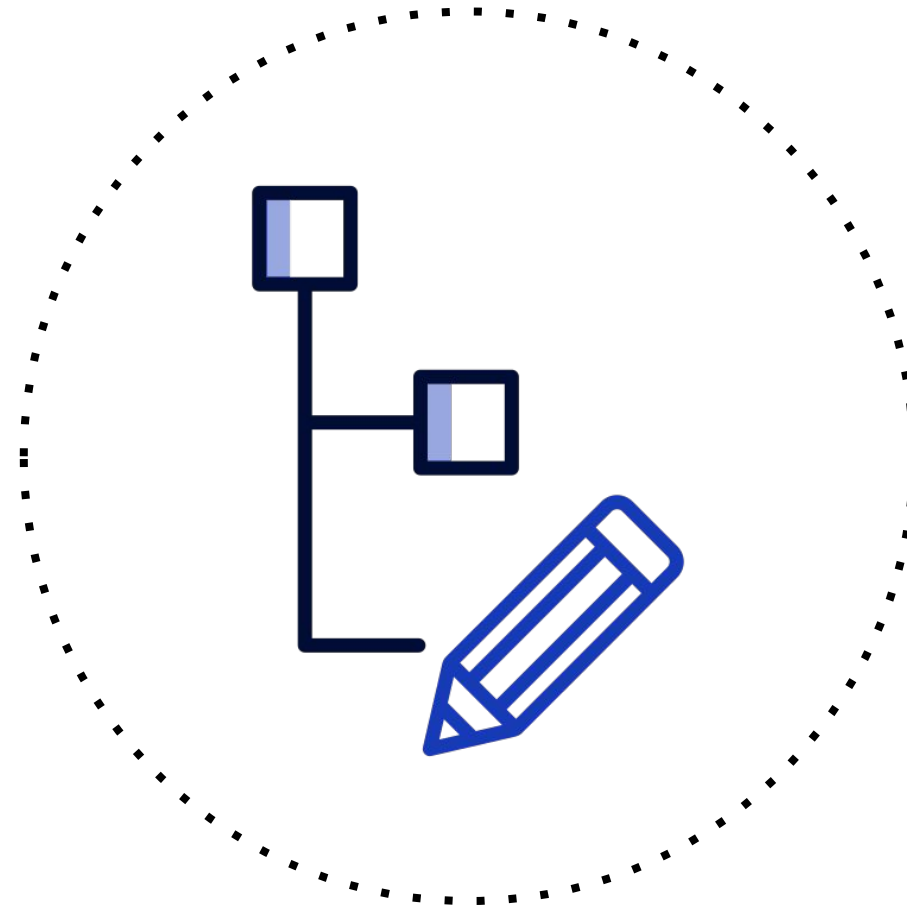


Ключевые слова



if

если



else

иначе

Как писать условия

```
if (условие) {  
    // код если условие выполняется  
} else {  
    // код если условие НЕ выполняется  
}
```



используется для комментариев в коде

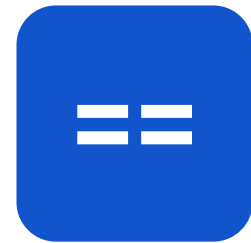


границы для записи условия

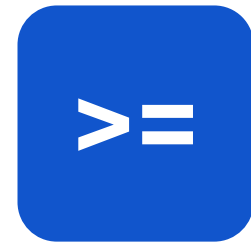


границы выполнения блока кода

Условные операторы



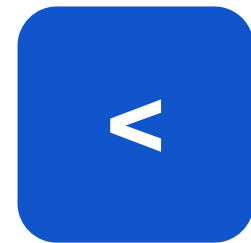
равно



больше или равно



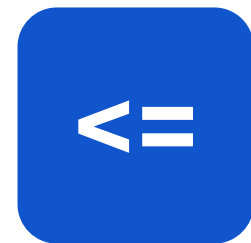
не равно



меньше



больше



меньше или равно

УСЛОВИЯ

Результат условия – логический тип **boolean**
(булевый тип).

Пример:

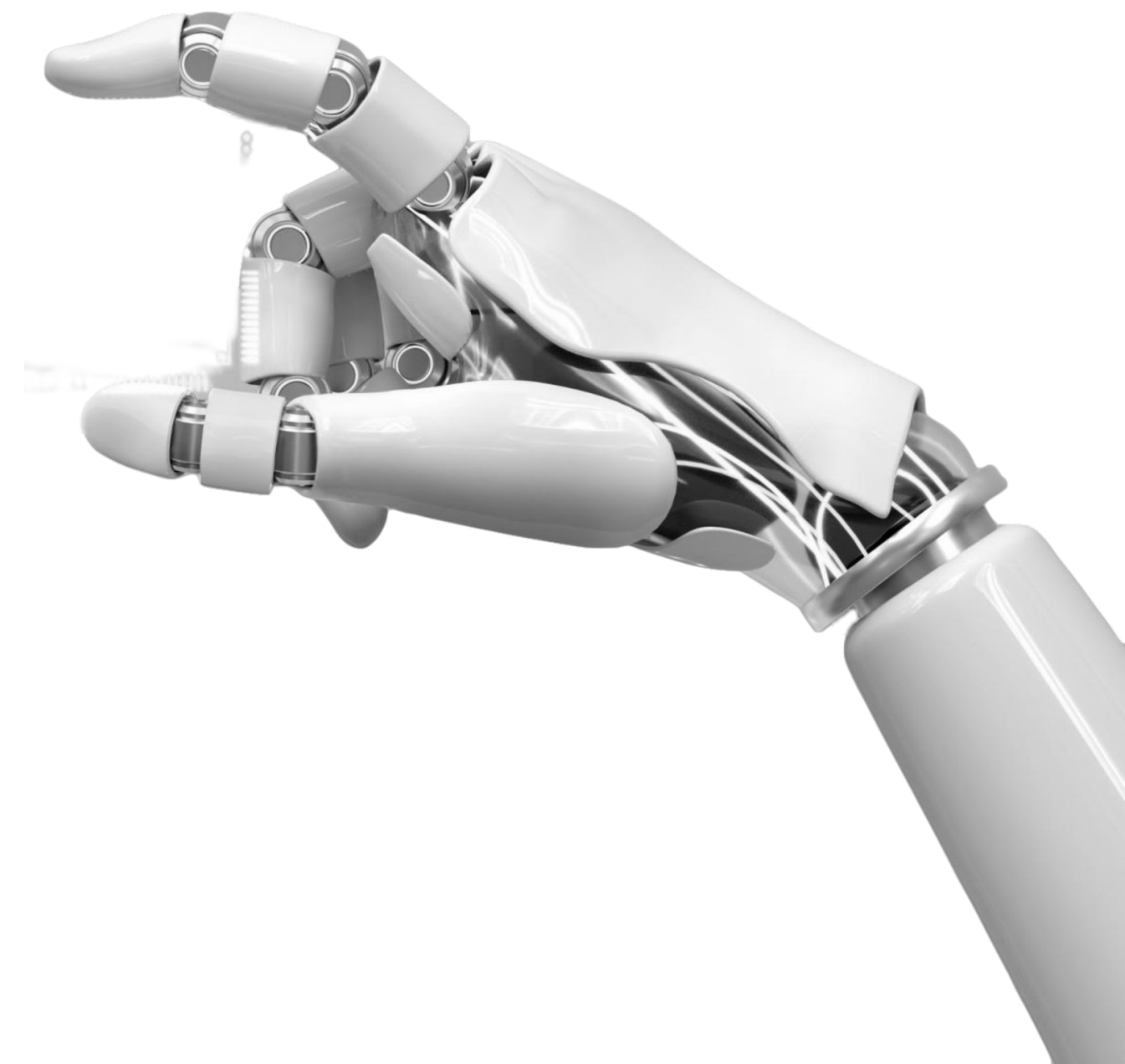
```
number a = 10;
```

```
number b = 20;
```

```
if (a == b) false
```

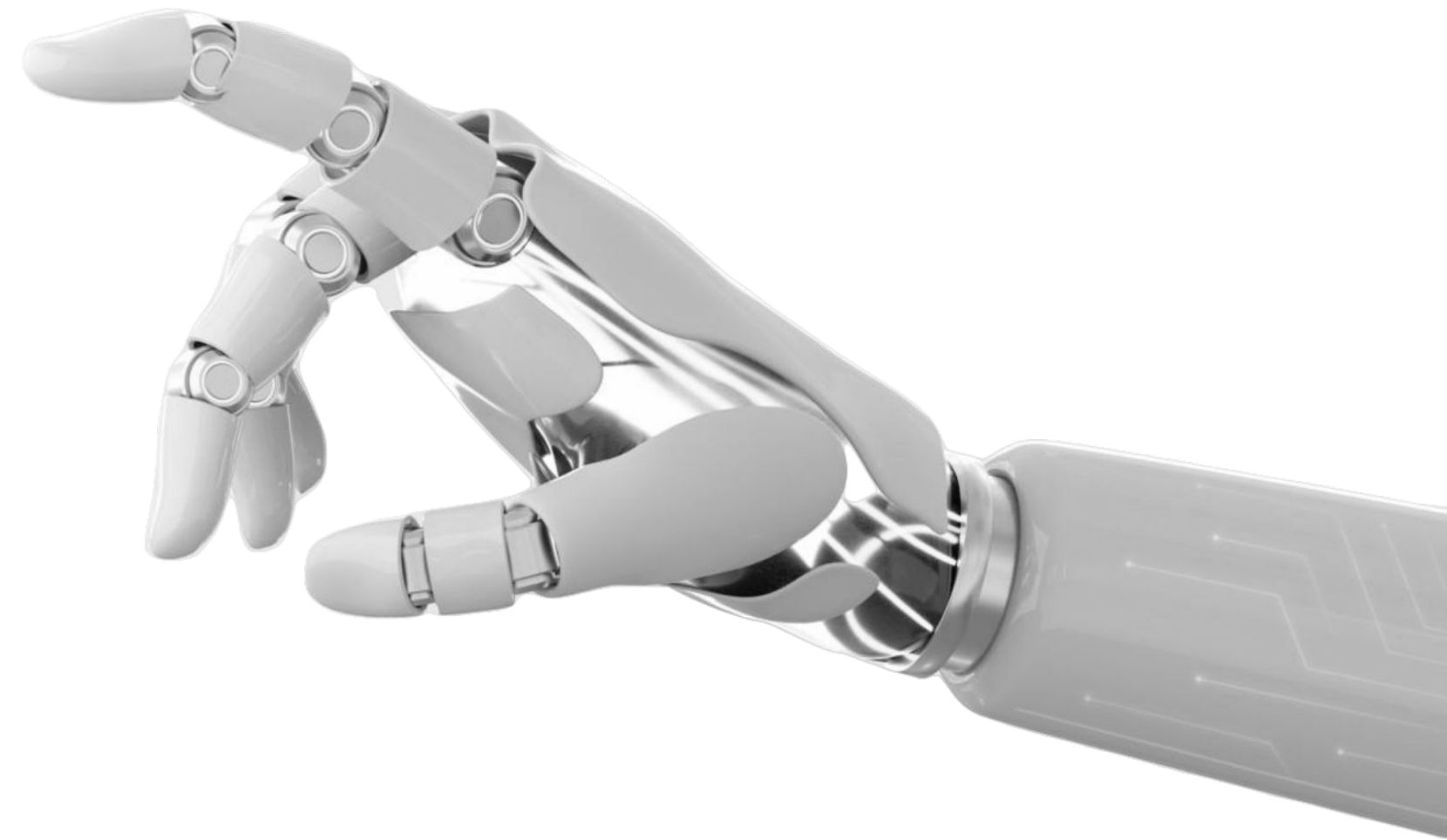
```
if (a > b) false
```

```
if (a < b) true
```



Код с условием

```
if (age >= 18) {  
    drink = "beer";  
}
```



Условие с **else**

```
if (age >= 18) {  
    drink = "beer";  
} else {  
    drink = "water";  
}
```



Логические операторы

Логические операторы – это специальные символы, которые комбинируют логические условия.



оператор **и**



оператор **или**

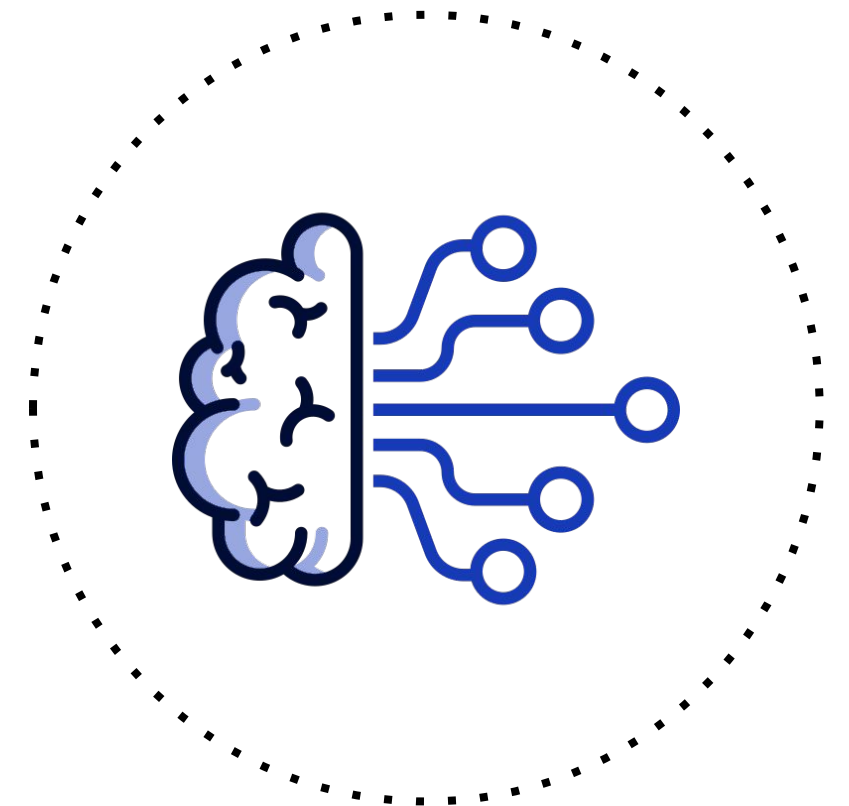
Пример:

```
if (name == "Арте́м"&& surname == "Ива́нов")
```

```
if (temperature > 20 || sun == true)
```

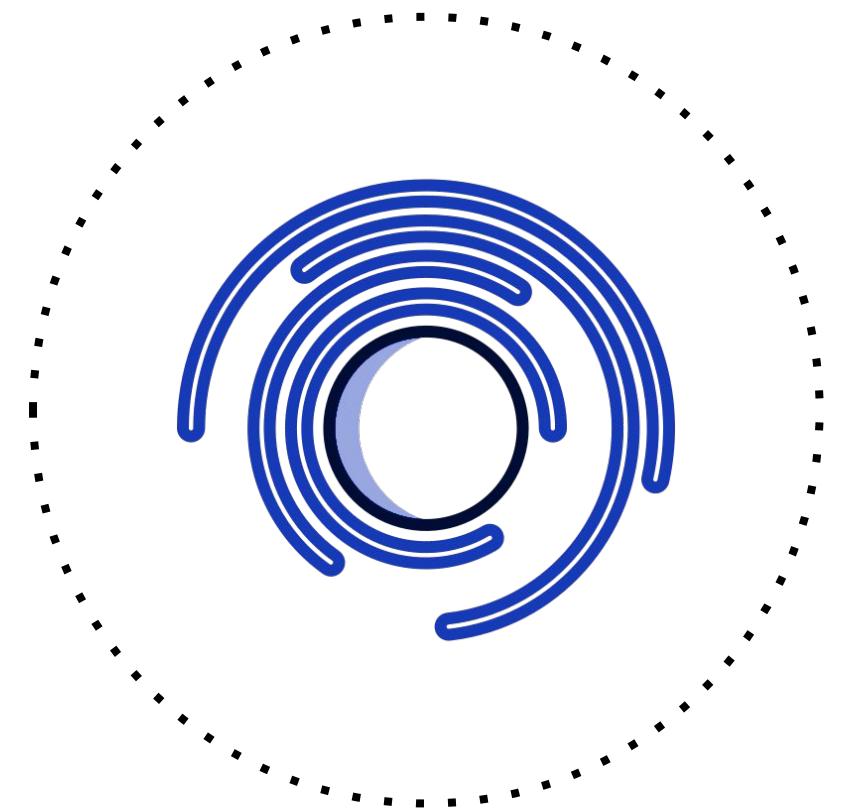
Логические операторы

```
if (age >= 18 && barHasBeer == true) {  
    drink = "beer";  
} else {  
    drink = "water";  
}
```



Boolean переменная

```
boolean weatherGood = sun == true && rain == false;
```



Boolean переменная

```
boolean isWeatherGood = sun == true && rain == false;
```

```
if (isWeatherGood) {
```

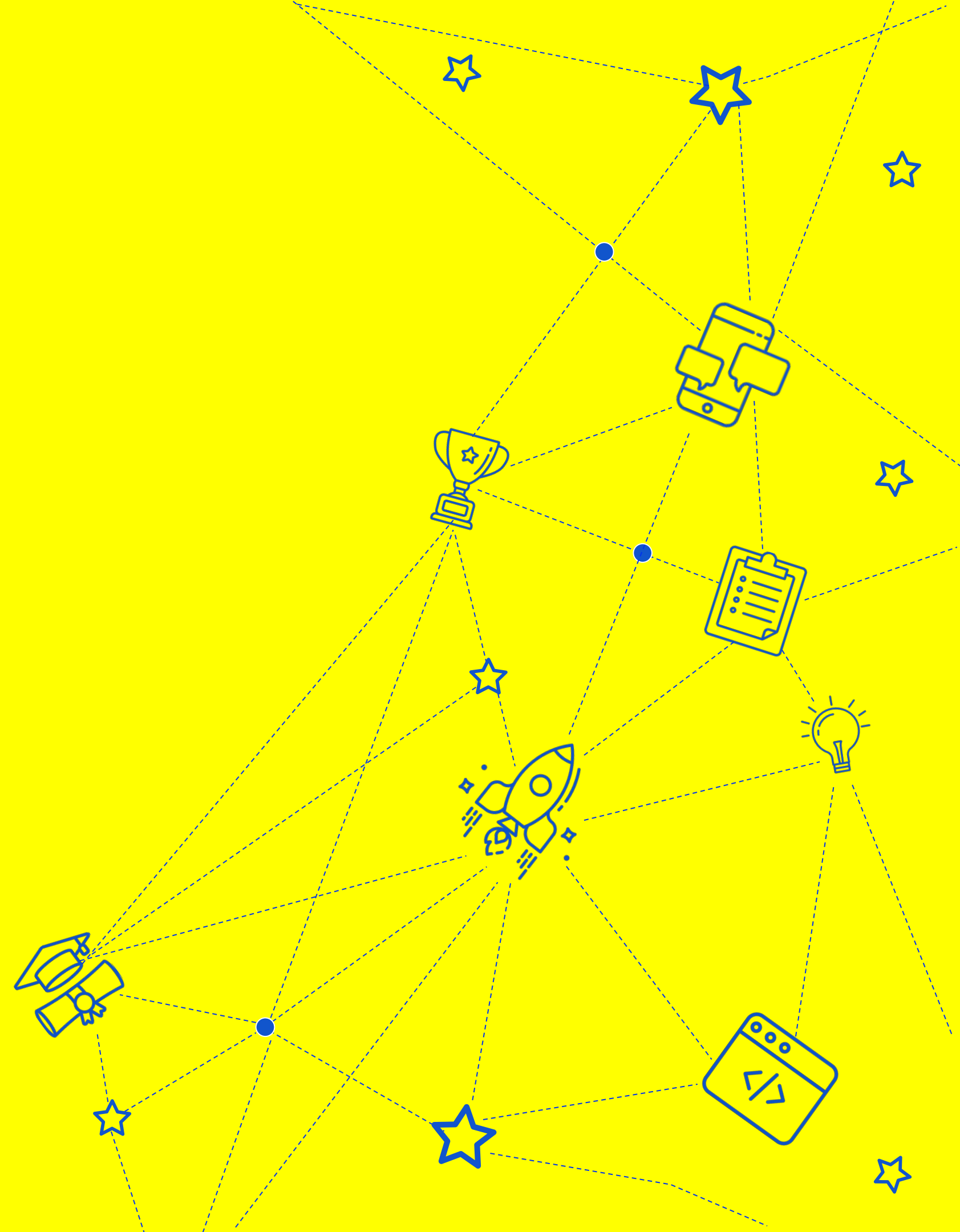
```
    walk = true;
```

```
} else {
```

```
    walk = false;
```

```
}
```

Функции



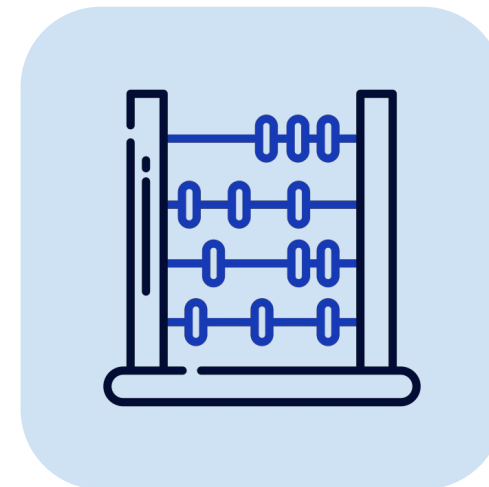
Функции

Функция – это мини-программа внутри вашей основной программы, которая совершает набор определенных действий.

Примеры:



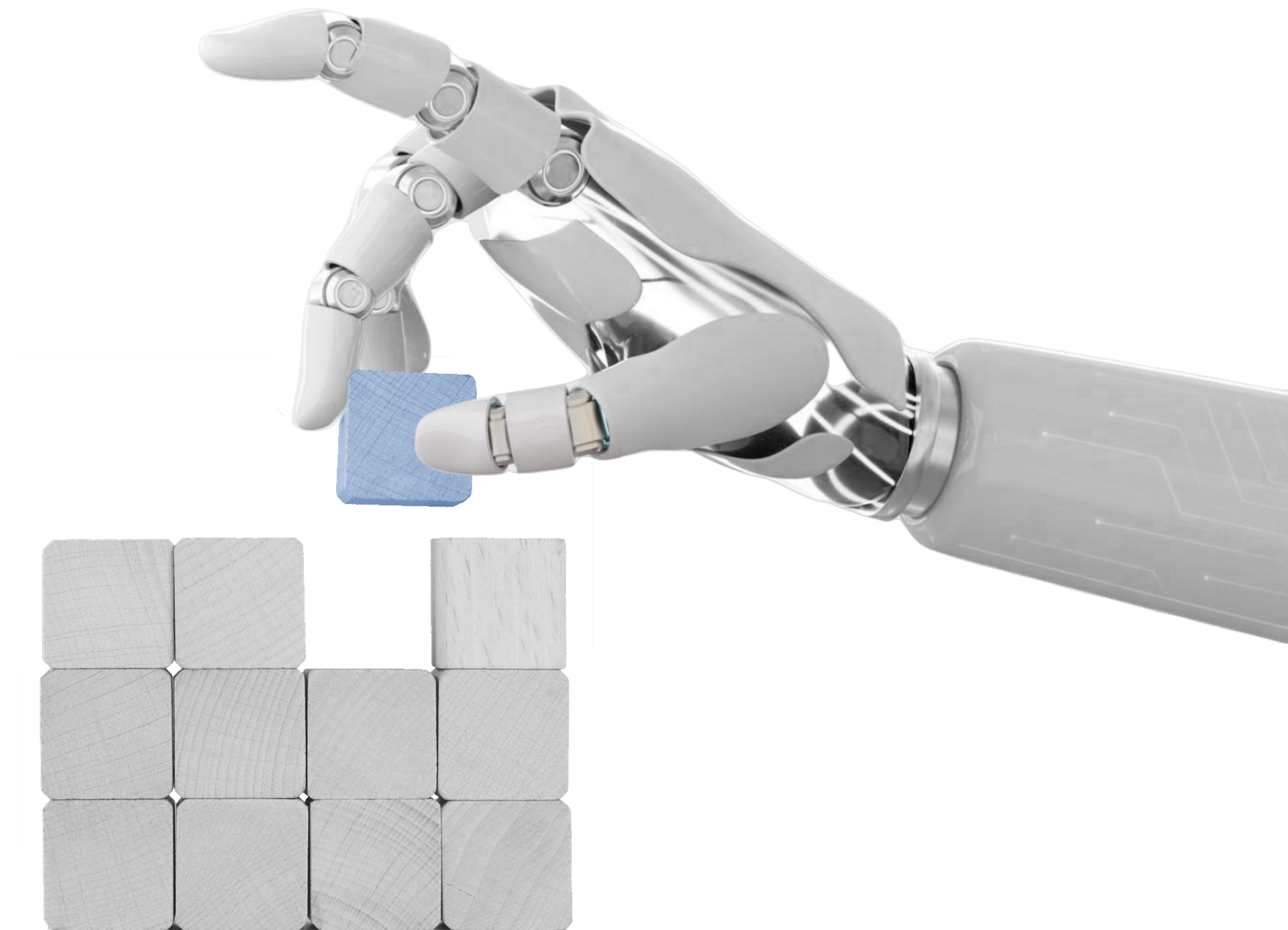
Функция
**«доехать до
университета»**



Функция
**«посчитать площадь
прямоугольника»**

Назначение

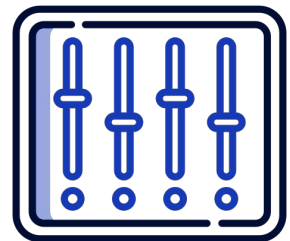
Функции помогают декомпозировать программу на отдельные компоненты и реализовывать их по отдельности.



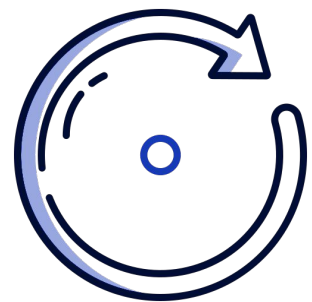
Из чего состоит функция



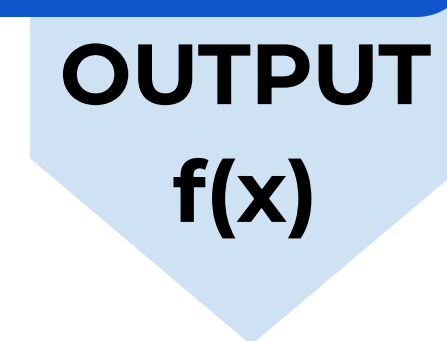
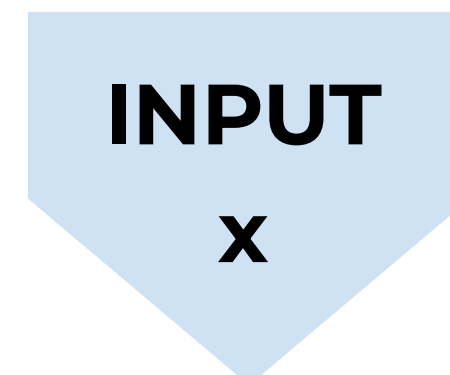
Имя



Параметры



Возвращаемое
значение



Имя функции

Имя функции должно быть понятным и отражать смысл.

Например:

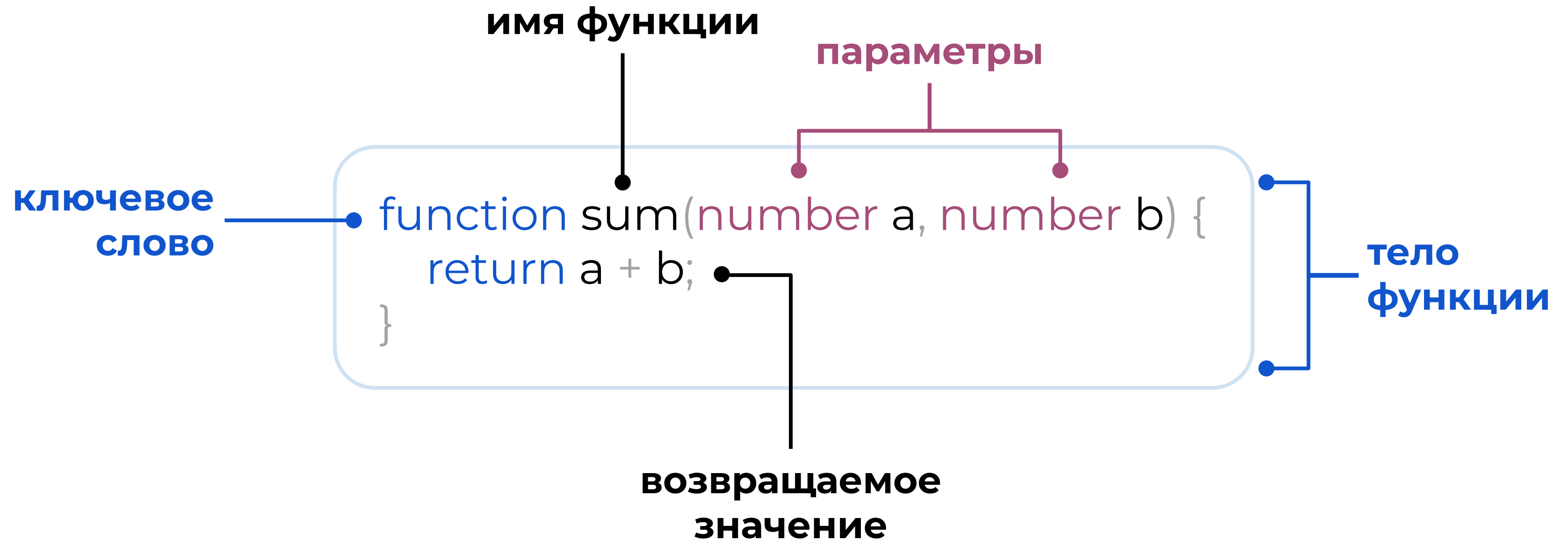
- ★ `sendMoney`
- ★ `getMonth`
- ★ `setTime`
- ★ `calculateDistance`



Синтаксис функции

```
function sum(number a, number b) {  
    return a + b;  
}
```

Синтаксис функции



Вызов функции

```
number result = sum(5, 10);
```



Пример функции

```
string name = "Artem";  
number money = getBalance(name);  
sendMoney("Alexey", money);
```



Локальные переменные

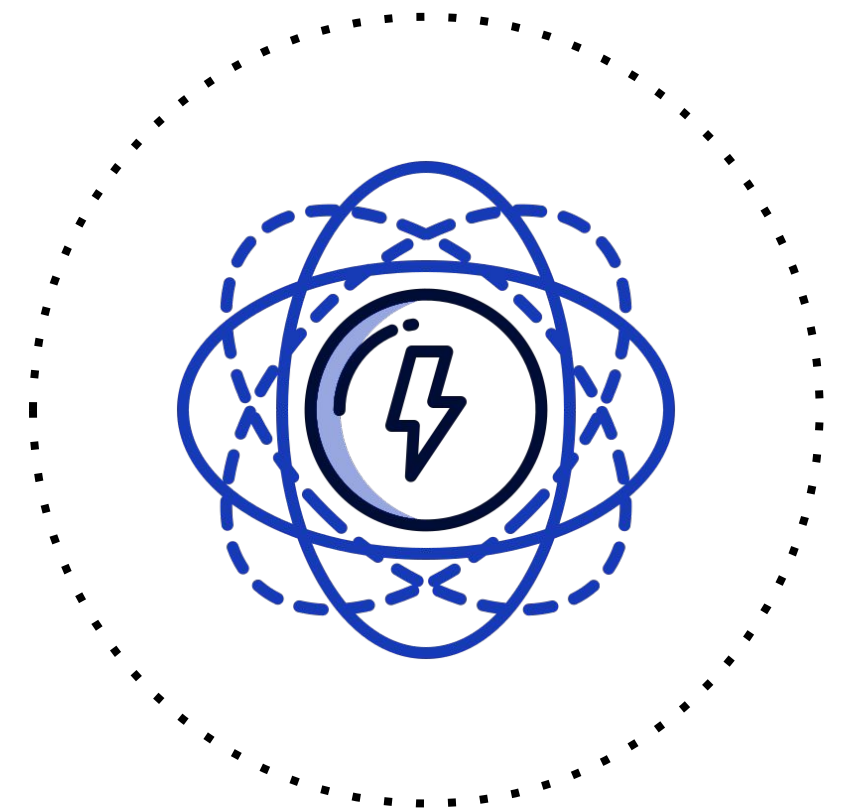
```
string myName = "Artem";
```

```
function sayHello(string name) {  
    string message = "Hello, ";  
    print(message + name);  
}
```

```
sayHello(myName); // "Hello, Artem"
```

```
print(myName); // "Artem"
```

```
print(message); // Ошибка, переменная недоступна
```



Переиспользование

Чтобы переиспользовать функцию
нужно знать:

- 1 Название и назначение
- 2 Параметры
- 3 Возвращаемое значение

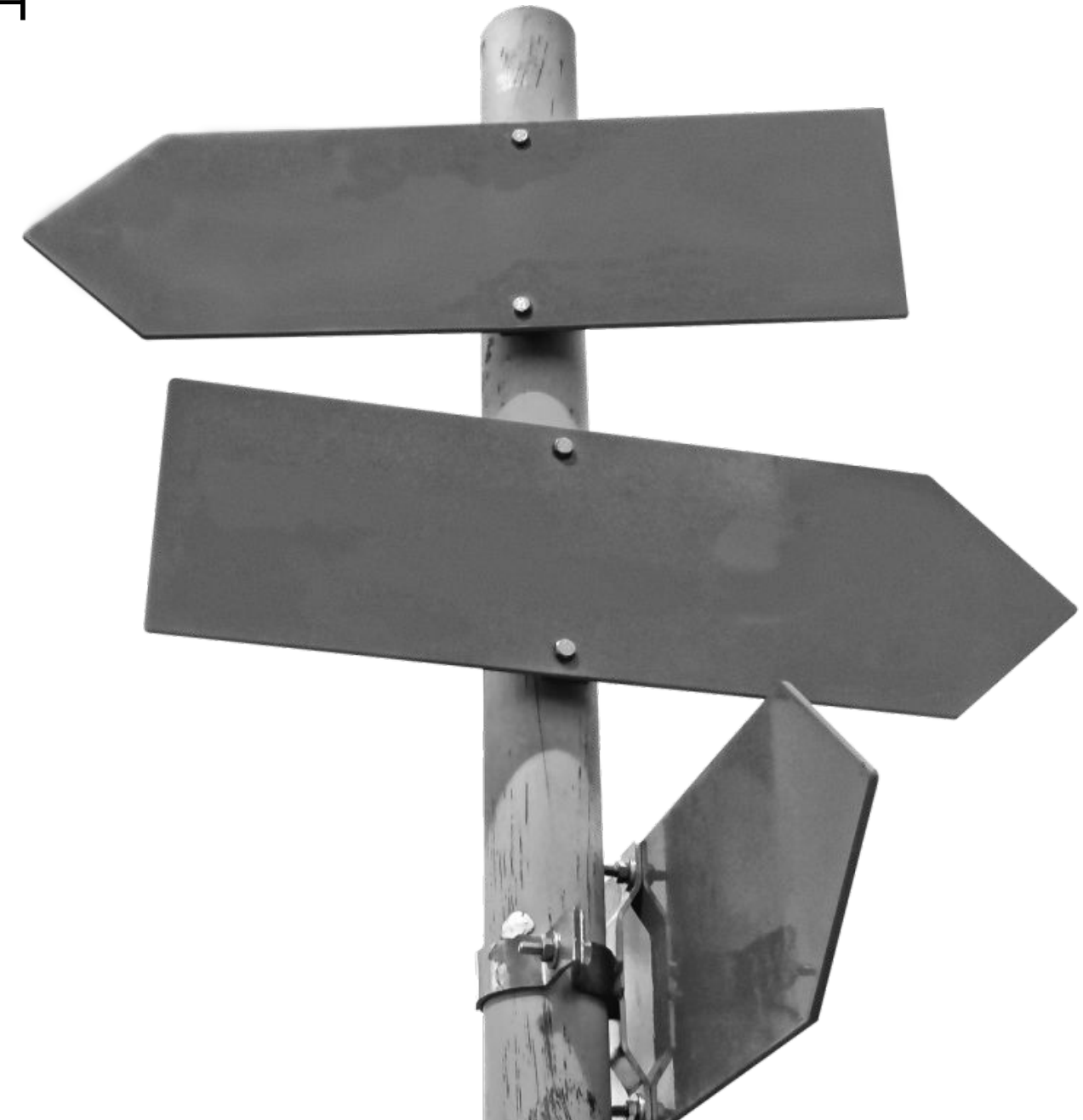


Вы можете загружать чужие функции из библиотек.

```
number result = sum(5, 10);
```

Указатели

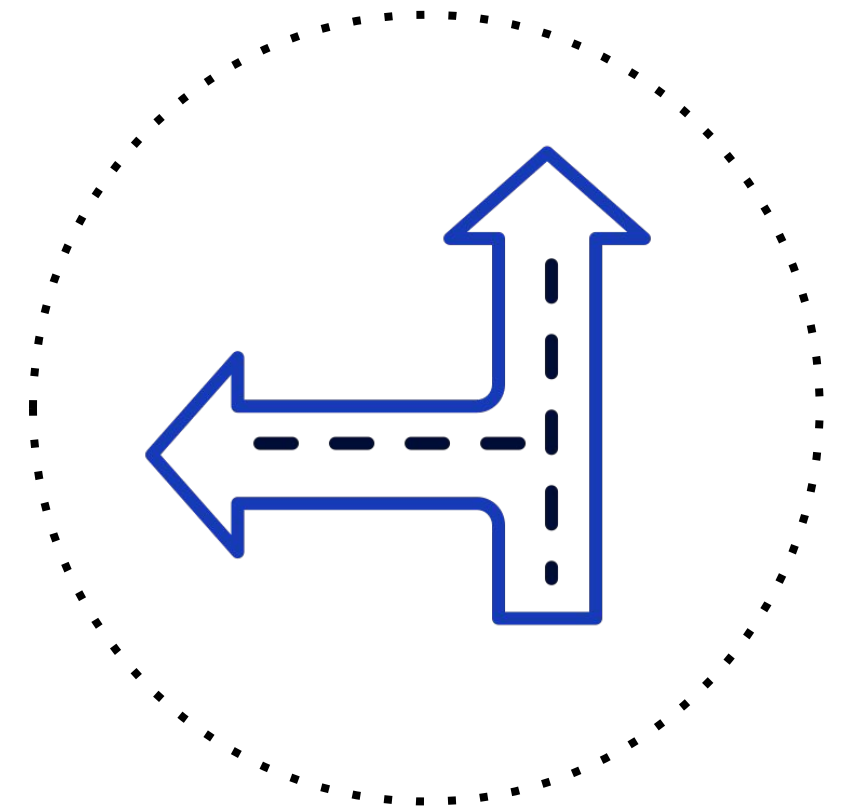
Указатель – это переменная, которая хранит не значение, а ссылку на память со значением. Несколько указателей могут ссылаться на один и тот же объект.



Указатели

```
function swap(string a, string b) {  
    string temp = a;  
    a = b;  
    b = temp;  
}
```

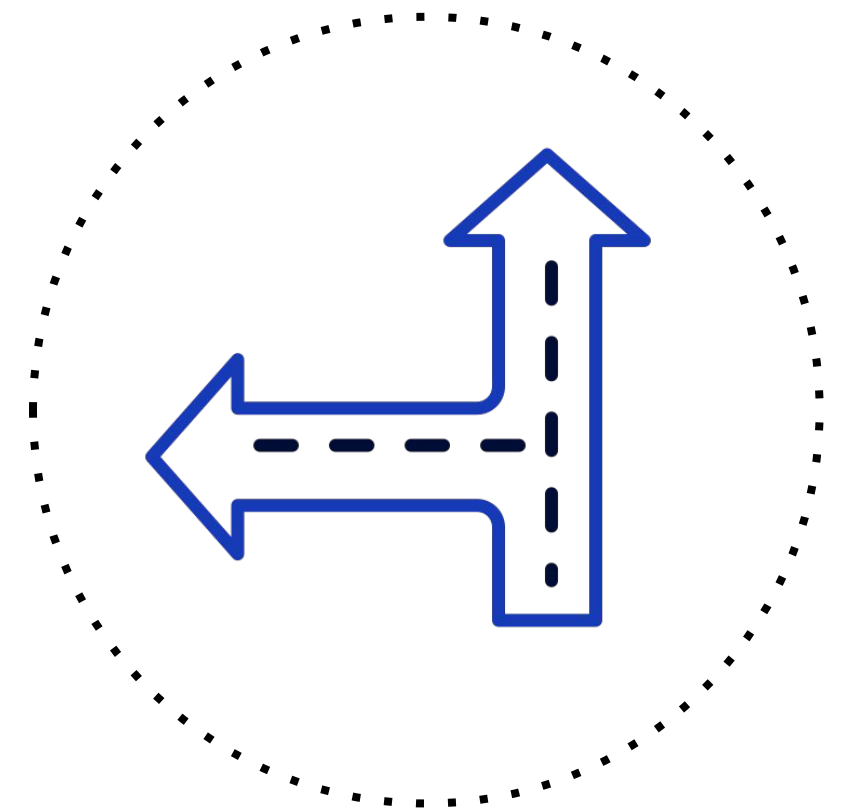
```
string name1 = "Artem";  
string name2 = "Alexey";  
swap(name1, name2);
```



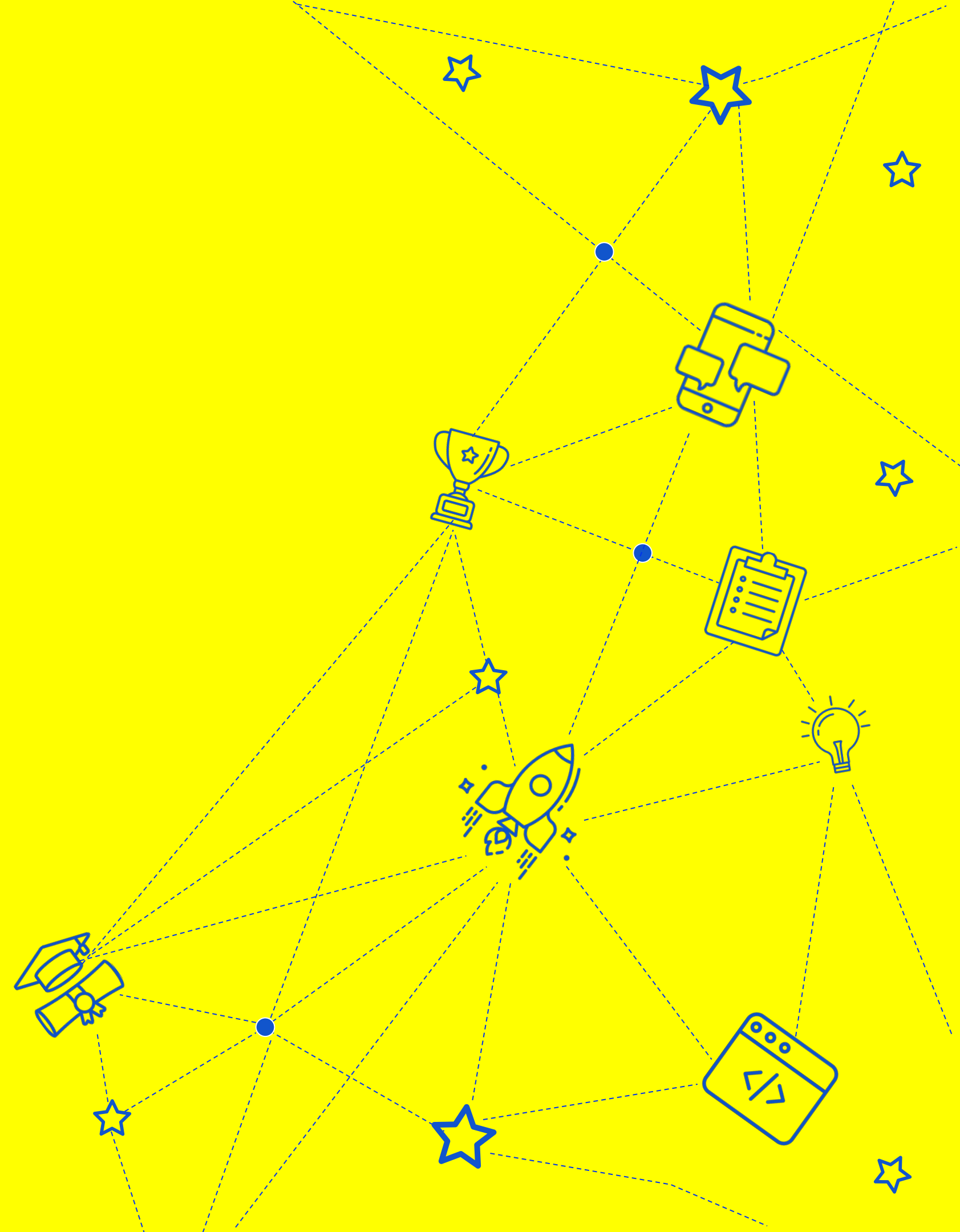
Указатели

```
function swap(string *a, string *b) {  
    string temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
string name1 = "Artem";  
string name2 = "Alexey";  
swap(name1, name2);
```

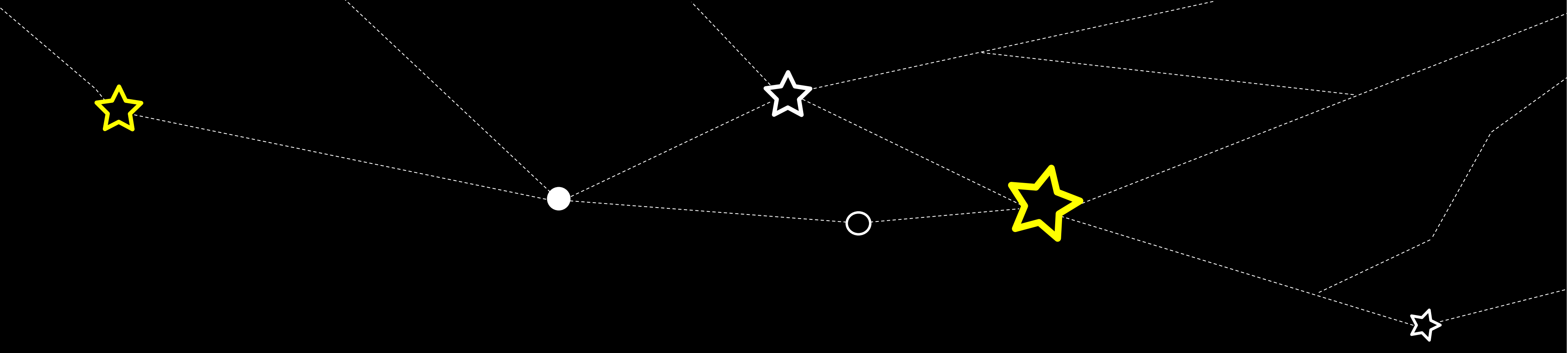


Подведем итоги



Итоги

- ★ Каждый язык программирования имеет ввод и вывод
- ★ Необходимо выучить синтаксис, чтобы писать на языке
- ★ Псевдокод это упрощенный и универсальный язык без строгих правил.
- ★ Основные типы переменных **string, number, boolean;**
- ★ Используйте ключевые слова **if** и **else** для написания условий
- ★ Функции помогают декомпозировать и структурировать программу;



СПАСИБО ЗА ВНИМАНИЕ

Артём Максимов



@tema_maksimov