**Функции: определение, параметры, возвращаемые значения**

**5.1. Что такое функция?**

**Функция** — это блок кода, который можно вызвать по имени. Она может принимать **аргументы**, выполнять действия и **возвращать результат**.

Функции:

* Делают код **модульным и читаемым**
* Позволяют **повторно использовать** логику
* Изолируют действия (важно для масштабирования и тестов)

**5.2. Объявление функции**

js

КопироватьРедактировать

function sayHello() {

console.log("Привет, Гриша!");

}

Вызов:

js

КопироватьРедактировать

sayHello();

**5.3. Параметры и аргументы**

js

КопироватьРедактировать

function greet(name) {

console.log("Привет, " + name + "!");

}

greet("Гриша"); // Привет, Гриша!

greet("Мария"); // Привет, Мария!

Можно задать **несколько параметров**:

js

КопироватьРедактировать

function sum(a, b) {

console.log(a + b);

}

**5.4. Возвращаемое значение: return**

js

КопироватьРедактировать

function square(x) {

return x \* x;

}

let result = square(5); // 25

После return выполнение функции останавливается.

**5.5. Функции vs console.log**

🔸 console.log() — выводит результат  
🔸 return — **передаёт** значение, которое можно сохранить, передать дальше, использовать

**5.6. Функции как выражения (анонимные)**

js

КопироватьРедактировать

const sayHi = function(name) {

return "Привет, " + name;

};

**5.7. Стрелочные функции (arrow functions)**

js

КопироватьРедактировать

const sum = (a, b) => a + b;

const greet = name => `Привет, ${name}`;

⚠️ Стрелочные функции **не имеют своего this**, отличаются от обычных в контексте объектов и классов.

**5.8. Значения по умолчанию**

js

КопироватьРедактировать

function greet(name = "гость") {

console.log("Привет, " + name);

}

greet(); // Привет, гость

**5.9. Вложенные функции и область видимости**

js

КопироватьРедактировать

function outer() {

let msg = "Внешняя";

function inner() {

console.log("Доступ из внутренней: " + msg);

}

inner();

}

**5.10. Callback-функции**

Функции можно передавать как аргументы другим функциям:

js

КопироватьРедактировать

function processUser(name, callback) {

callback(name);

}

processUser("Гриша", function(n) {

console.log("Добро пожаловать, " + n);

});

**5.11. Частые ошибки**

❌ Нет return, хотя нужен результат  
❌ Путают console.log() и return  
❌ Переиспользуют переменные внутри функции → конфликты  
❌ Отсутствие параметров → undefined

**5.12. Глоссарий**

| **Термин** | **Значение** |
| --- | --- |
| Функция | Код, выполняемый по вызову |
| Параметр | Формальное имя переменной в определении |
| Аргумент | Конкретное значение при вызове |
| return | Возврат значения |
| Анонимная функция | Функция без имени |
| Callback | Функция, переданная другой функции |

**5.13. Вопросы для самопроверки**

1. Что делает return?
2. В чём разница между параметром и аргументом?
3. Как записать стрелочную функцию?
4. Что произойдёт, если не передать аргумент?
5. Что такое callback?