JavaScript. Базовый уровень

Урок 2

Основные операторы JavaScript

Операторы и их приоритеты выполнения.
Условные операторы и циклы.

План урока

- Операторы.
- Принципы ветвления, блок-схемы.
- Конструкция if-else.

План урока

- Конструкция switch.
- Тернарный оператор.
- Функции.


Операторы в JavaScript

Оператор

Это наименьшая автономная часть языка программирования, то есть команда.

Операторы бывают:

- унарные;
- бинарные.

Унарный оператор

Применяется к одному операнду:

```
var x = 1;


x = -x; // унарный минус
```

Бинарный оператор

Применяется к двум операндам:

```
var a = 1;
var b = 2;


a + b; // бинарный плюс
```

У некоторых операторов есть особые названия:

- **инкремент** — означает увеличение операнда на установленный фиксированный шаг (как правило, единицу). Он же **a++** или **a+1**;
- **декремент** — обратная инкременту операция: **a--** или **a-1**;
- **конкатенация** — сложение строк. Обратной операции нет.

Приоритеты операторов

Оператор	Описание
. [ ] ( )	Доступ к полям, индексация массивов, вызовы функций и группировка выражений
++ -- - ~ ! delete new typeof void	Унарные операторы, тип возвращаемых данных, создание объектов, неопределенные значения
*** / %**	Умножение, деление, деление по модулю
+ - +	Сложение, вычитание, объединение строк
<< >> >>>	Сдвиг битов

Приоритеты операторов

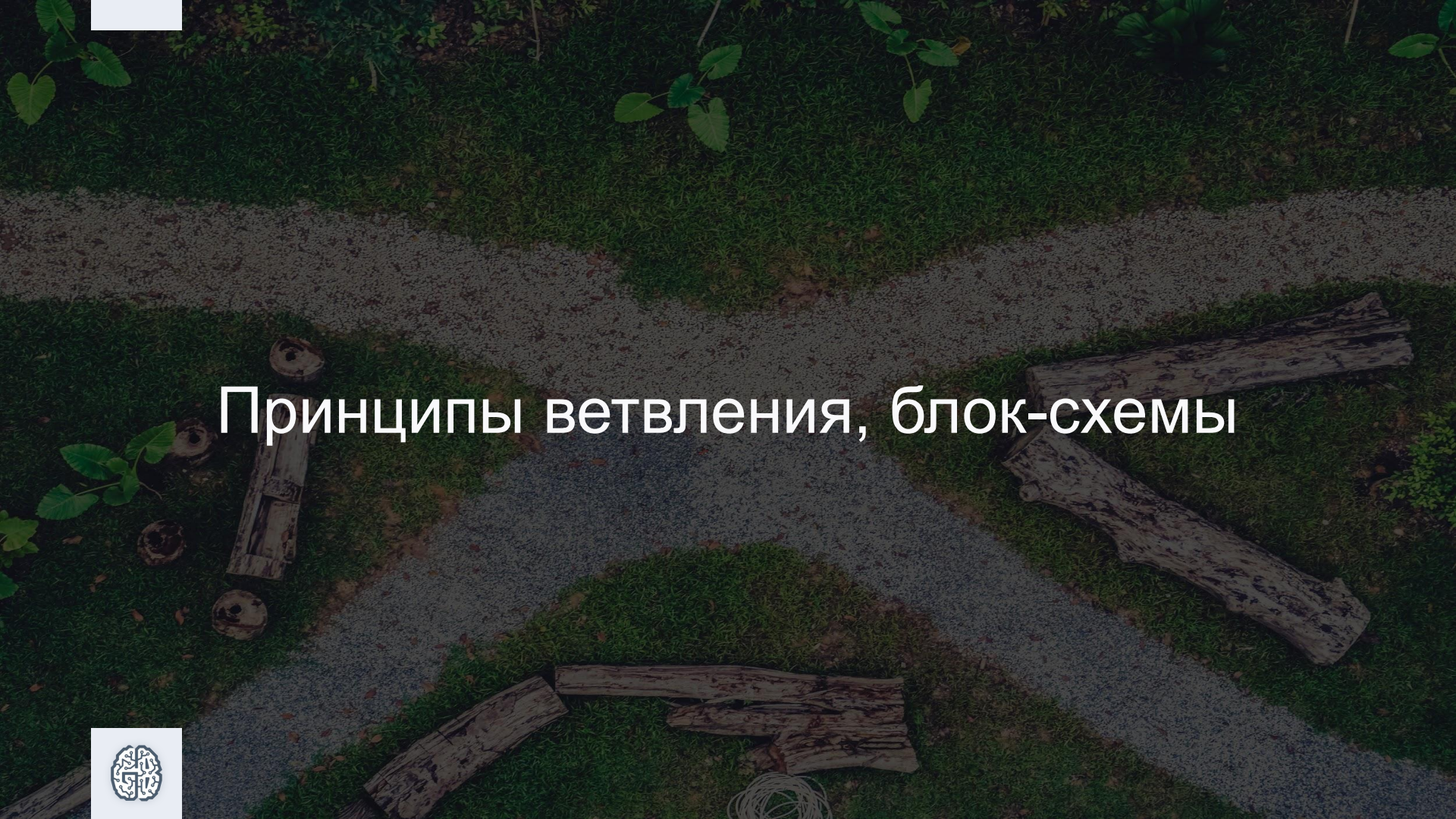
Оператор	Описание
< <= > >= instanceof	Меньше, меньше или равно, больше, больше или равно, instanceof
== != === !==	Равенство, неравенство, строгое равенство, строгое неравенство
&	Побитовое И
^	Побитовое исключающее ИЛИ
|	Побитовое ИЛИ

Приоритеты операторов

Оператор	Описание
&&	Логическое И
||	Логическое ИЛИ
?:	Условный оператор
= OP=	Присваивание, присваивание с операцией (например += и &=)
,	Вычисление нескольких выражений

Пример

```
var a = 5 * 3 - 7;
```

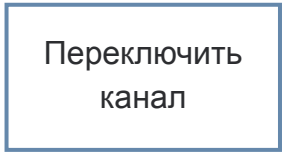

Принципы ветвления, блок-схемы

Ветвление

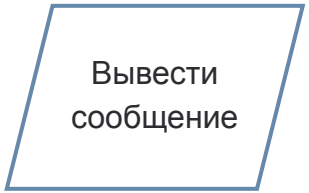
«Если случится событие А, то я выполню действие Б».

Для ветвления в программировании применяются специальные операторы, обеспечивающие выполнение определенной команды или набора команд только при условии истинности логического выражения или их группы.

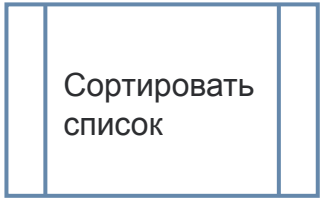
Блок-схемы



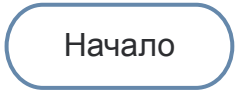
процесс



данные

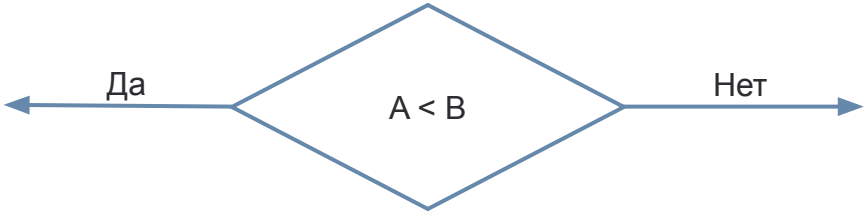


предопределенный процесс



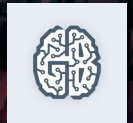
терминатор

Блок-схемы

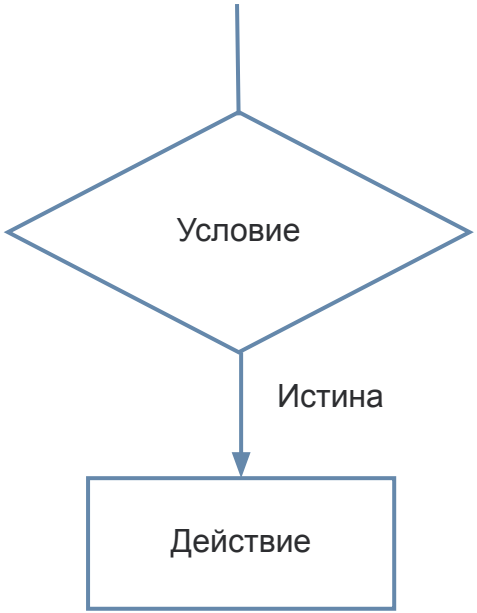


Ветвление

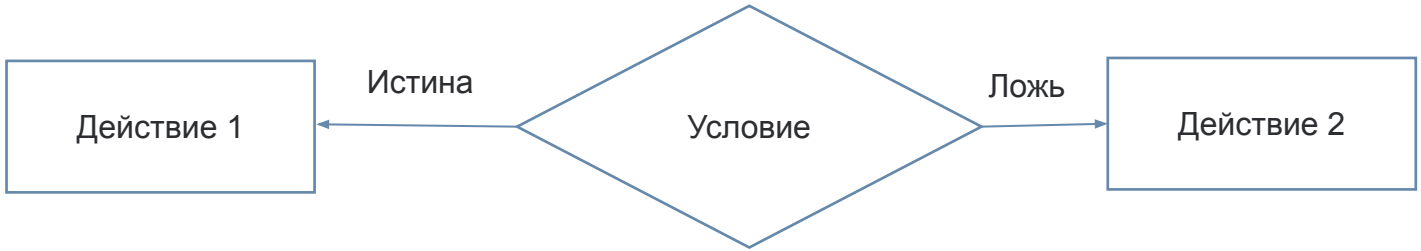
Конструкция if-else



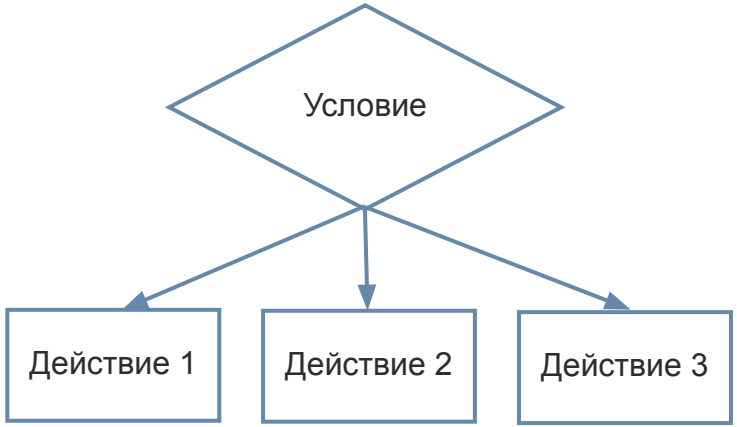
Оператор if



Оператор else

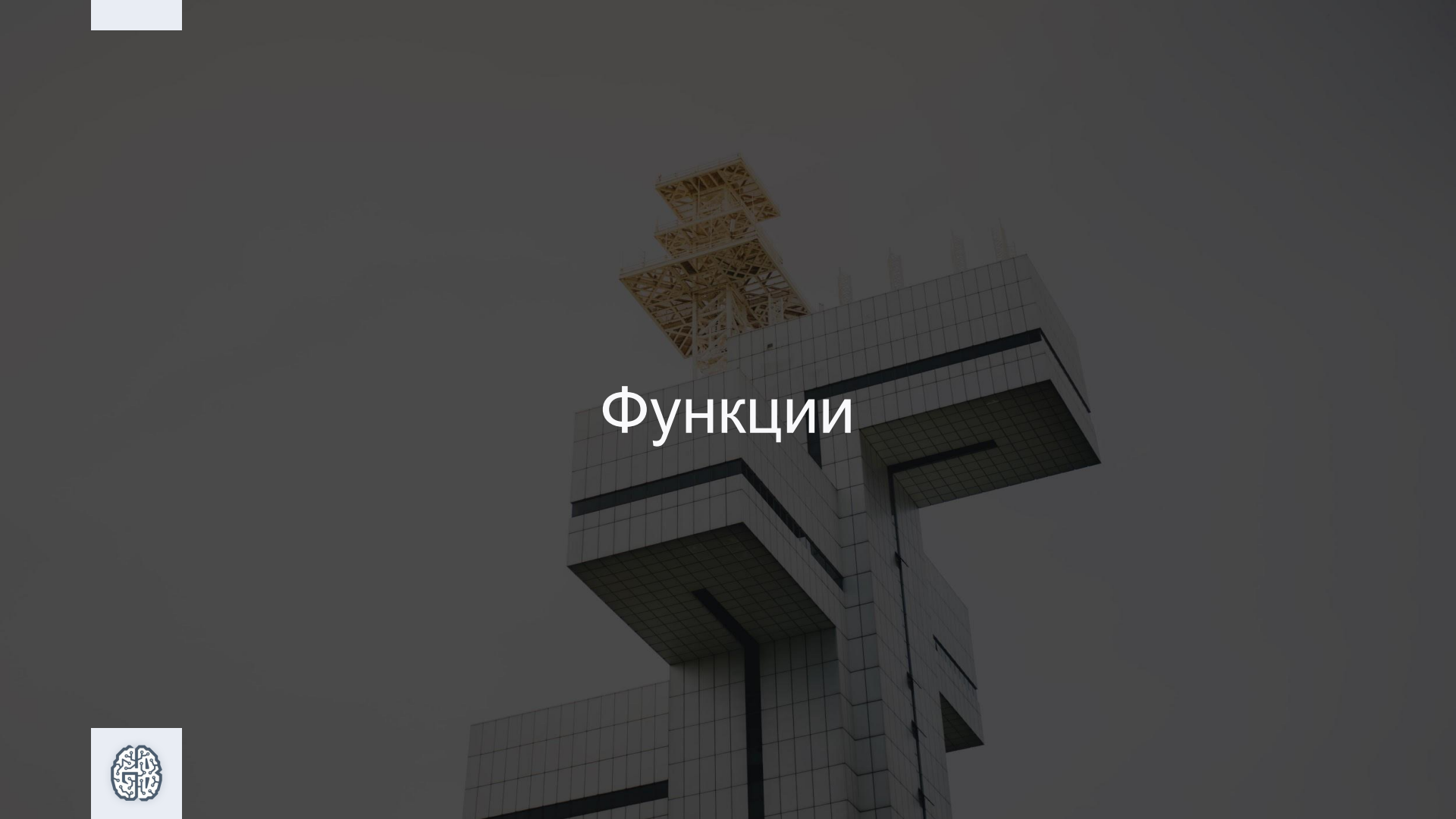


Оператор else if



- Оператор **switch**.
- Тернарный оператор.

Функции

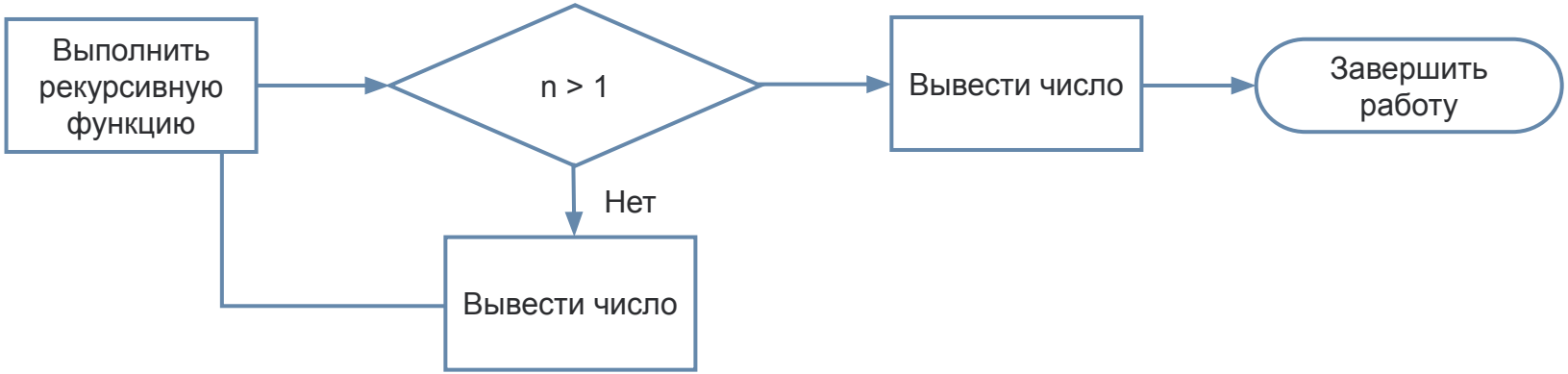


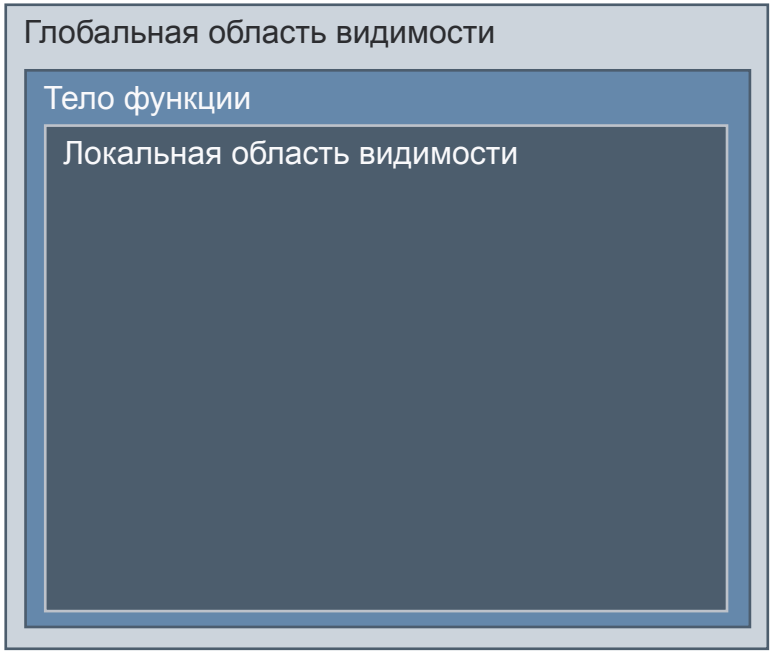
Функции

Функция – это блок кода, к которому можно обращаться из разных частей скрипта.

Рекурсия

«Чтобы понять рекурсию, нужно сначала понять рекурсию» (автор неизвестен)





Области видимости

Практическое задание

Практическое задание

1, 2. Проверить код, представленный в практическом задании в методичке.

Практическое задание

3. Объявить две целочисленные переменные — **a** и **b** и задать им произвольные начальные значения. Затем написать скрипт, который работает по следующему принципу:
   - если **a** и **b** положительные, вывести их разность;
   - если **a** и **b** отрицательные, вывести их произведение;
   - если **a** и **b** разных знаков, вывести их сумму;

Ноль можно считать положительным числом.

Практическое задание

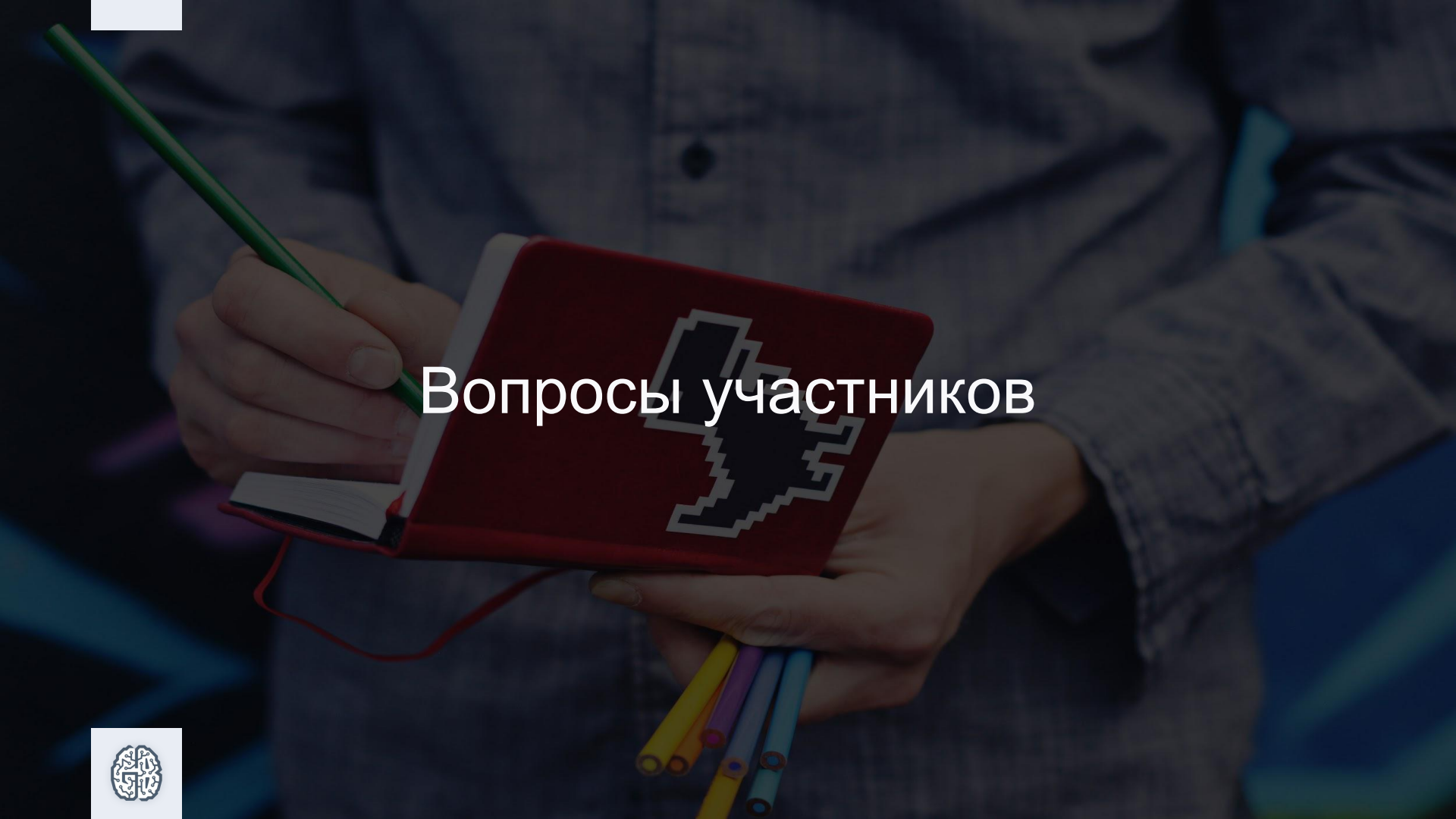
4. Присвоить переменной **a** значение в промежутке [0..15]. С помощью оператора **switch** организовать вывод чисел от **a** до 15.
5. Реализовать четыре основные арифметические операции в виде функций с двумя параметрами. Обязательно использовать оператор **return**.

Практическое задание

6. Реализовать функцию с тремя параметрами: **function mathOperation(arg1, arg2, operation)**, где **arg1**, **arg2** — значения аргументов, operation — строка с названием операции. В зависимости от переданного значения выполнить одну из арифметических операций (использовать функции из пункта 3) и вернуть полученное значение (применить **switch**).

Практическое задание

7. * Сравнить **null** и **0**. Объяснить результат.
8. * С помощью рекурсии организовать функцию возведения числа в степень. Формат: **function power(val, pow), где val** — заданное число, **pow** —— степень.


Вопросы участников