1. Какими способами можно объявлять массивы в JS?

Есть 2 способа объявления массива через квадратные скобки и через new Array

1. let fruits = ["Яблоко", "Апельсин", "Слива"];

2. let arr = *new Array*("Яблоко", "Груша", "и тд");

1. Приведите 3 примера, из тех, которые не были озвучены в материалах, что могло бы быть массивом?

*Как может выглядеть ваш ответ: например, список книг в приложении "Библиотека" - это массив. Список людей в паспортном столе - это массив. Список полей в форме регистрации - это массив.*

* 1. Список учеников в классе
  2. Список любимых песен
  3. Список популярных туристических городов
  4. Список полетов авиа кампании

1. Каким еще способом, кроме pop и shift можно удалять элементы из массивов?

Метод arr.splice(str) – это универсальный «швейцарский нож» для работы с массивами. Умеет всё: добавлять, удалять и заменять элементы.

Его синтаксис:

arr.splice(index[, deleteCount, elem1, ..., elemN])

1. Можно ли пропускать части **for**? Что получится, если написать for(;;)?

Любая часть for может быть пропущена.

Для примера, мы можем пропустить начало если нам ничего не нужно делать перед стартом цикла.

Вот так:

let i = 0; // мы уже имеем объявленную i с присвоенным значением

for (; i < 3; i++) { // нет необходимости в "начале"

alert( i ); // 0, 1, 2

}

Можно убрать и шаг:

let i = 0;

for (; i < 3;) {

alert( i++ );

}

Это сделает цикл аналогичным while (i < 3).

А можно и вообще убрать всё, получив бесконечный цикл:

for (;;) {

// будет выполняться вечно

}

1. Самостоятельно разберитесь, как работает цикл while и приведите два примера кода с его использованием.

Цикл while имеет следующий синтаксис:

while (condition) {

// код также называемый "телом цикла"

}

Код из тела цикла выполняется, пока условие condition истинно.

Например, цикл ниже выводит i, пока i < 3:

let i = 0;

while (i < 3) { // выводит 0, затем 1, затем 2

alert( i );

i++;

}

Одно выполнение тела цикла по-научному называется *итерация*. Цикл в примере выше совершает три итерации.

Если бы строка i++ отсутствовала в примере выше, то цикл бы повторялся (в теории) вечно. На практике, конечно, браузер не позволит такому случиться, он предоставит пользователю возможность остановить «подвисший» скрипт, а JavaScript на стороне сервера придётся «убить» процесс.

Любое выражение или переменная может быть условием цикла, а не только сравнение: условие while вычисляется и преобразуется в логическое значение.

Например, while (i) – более краткий вариант while (i != 0):

let i = 3;

while (i) { // когда i будет равно 0, условие станет ложным, и цикл остановится

alert( i );

i--;

}

**Фигурные скобки не требуются для тела цикла из одной строки**

Если тело цикла состоит лишь из одной инструкции, мы можем опустить фигурные скобки {…}:

let i = 3;

while (i) alert(i--);

1. Какой получится массив, если создать его вот так new Array(5)?

Получится пустой массив с 5 элементами

1. Как вывести чётные числа от 2 до 10 при помощи цикла for?

for (let i = 2; i<=10 ; i=i+2){

alert(i); }

1. Каков будет результат выполнения этого кода? Почему?

let arr = ["a", "b"];

arr.push(function() {

alert( arr );

})

arr[2](); // ?

в конце массива добавиться функция function

и массив станет : a b function

arr[2](); // будет равняться заданной функции function

1. Три основных способа перебора элементов массива?
2. Цикл **for (традиционный)**
3. Цикл **for ... of**
4. Метод **forEach** (появился в ES5)
5. Как можно выбрать все инпуты из вашей формы регистрации из прошлого ДЗ с помощью querySelector\*?
6. Самостоятельно разберитесь, как можно проще всего сделать сортировку в массиве на JS? *Пожалуйста, не усложняйте ответ на этот вопрос* 😸

let arr = [5, 2, 1, -10, 8];

arr.sort((a, b) => b - a);

alert( arr );

1. Как можно принудительно остановить выполнение цикла?

Обычно цикл завершается при вычислении *условия* в false.

Но мы можем выйти из цикла в любой момент с помощью специальной директивы break.

Например, следующий код подсчитывает сумму вводимых чисел до тех пор, пока посетитель их вводит, а затем – выдаёт:

let sum = 0;

while (true) {

let value = +prompt("Введите число", '');

if (!value) break; // (\*)

sum += value;

}

alert( 'Сумма: ' + sum );

Директива break в строке (\*) полностью прекращает выполнение цикла и передаёт управление на строку за его телом, то есть на alert.