Урок 8

Массивы

Массив (Array) в JavaScript является глобальным объектом, который используется для создания массивов; которые представляют собой высокоуровневые спископодобные объекты.

Создание массива

```
let fruits = ['Яблоко', 'Банан'];

console.log(fruits.length); //длина массива
// 2
```

Доступ к элементу массива по индексу

```
let first = fruits[0];
// Яблоко

let last = fruits[fruits.length - 1];
// Банан
```

Добавление элемента в конец массива

```
let newLength = fruits.push('Апельсин');
// ["Яблоко", "Банан", "Апельсин"]
```

Удаление последнего элемента массива

```
let last = fruits.pop(); // удалим Апельсин (из конца)
// ["Яблоко", "Банан"];
```

Удаление первого элемента массива

```
let first = fruits.shift(); // удалим Яблоко (из начала)
// ["Банан"];
```

Добавление элемента в начало массива

```
let newLength = fruits.unshift('Клубника') // добавляет в начало
// ["Клубника", "Банан"];
```

Поиск номера элемента в массиве

```
fruits.push('Манго');
// ["Клубника", "Банан", "Манго"]
let pos = fruits.indexOf('Банан');
// 1
```

Удаление элемента с определённым индексом

```
let removedItem = fruits.splice(pos, 1); // так можно удалить элемент

// ["Клубника", "Манго"]
```

Удаление нескольких элементов, начиная с определённого индекса

```
let vegetables = ['Капуста', 'Репа', 'Редиска', 'Морковка'];
console.log(vegetables);
// ["Капуста", "Репа", "Редиска", "Морковка"]

let pos = 1, n = 2;

let removedItems = vegetables.splice(pos, n);
// так можно удалить элементы, n определяет количество элементов для
удаления,
// начиная с позиции(pos) и далее в направлении конца массива.

console.log(vegetables);
// ["Капуста", "Морковка"] (исходный массив изменён)

console.log(removedItems);
// ["Репа", "Редиска"]
```

Программа, которая считывает три числа через prompt и добавляет их в массив.

```
let arr = [];
let num1 = Number(prompt('Enter the number.'));
let num2 = Number(prompt('Enter the number.'));
let num3 = Number(prompt('Enter the number.'));
arr.push(num1, num2, num3);
console.log(arr);
```

Задание:

1. Создайте массив styles с элементами «Джаз» и «Блюз».
2. Добавьте «Рок-н-ролл» в конец.
3. Замените значение в середине на «Классика». Ваш код для поиска значения в середине должен работать для массивов с любой длиной.
4. Удалите первый элемент массива и покажите его.
5. Вставьте Рэп и Регги в начало массива.

Массив по ходу выполнения операций:
Джаз, Блюз
Джаз, Блюз, Рок-н-ролл
Джаз, Классика, Рок-н-ролл
Классика, Рок-н-ролл
Рэп, Регги, Классика, Рок-н-ролл

Решение

```
let styles = ["Джаз", "Блюз"];
styles.push("Рок-н-ролл");
styles[Math.floor((styles.length - 1) / 2)] = "Классика";
alert(styles.shift());
styles.unshift("Рэп", "Регги");
```

Задание: Написать программу, в которой объявлен массив с 5 числовыми элементами. Программа должна заполнить новый пустой массив квадратами чисел из первого массива.
Пример:
Исходный массив [1, 4, 2, 5, 3]
Итоговый массив [1, 16, 4, 251, 9]

Циклы

Последовательность инструкций, предназначенная для многократного исполнения, называется **телом цикла**. Единичное выполнение тела цикла называется **итерацией**. Выражение, определяющее, будет в очередной раз выполняться итерация или цикл завершится, называется **условием выхода** или условием окончания цикла (либо условием продолжения в зависимости от того, как интерпретируется его истинность — как признак необходимости завершения или продолжения цикла). Переменная, хранящая текущий номер итерации, называется **счётчиком** итераций цикла или просто счётчиком цикла. Цикл не обязательно содержит счётчик, счётчик не обязан быть один — условие выхода из цикла может зависеть от нескольких изменяемых в цикле переменных, а может определяться внешними условиями (например, наступлением определённого времени), в последнем случае счётчик может вообще не понадобиться.

Исполнение любого цикла включает первоначальную инициализацию переменных цикла, проверку условия выхода, исполнение тела цикла и обновление переменной цикла на каждой итерации. Кроме того, большинство языков программирования предоставляет средства для досрочного управления циклом, например, операторы завершения цикла, то есть выхода из цикла независимо от истинности условия выхода (break) и операторы пропуска итерации (continue).

Синтаксис СИ подобного цикла.

```
// Цикл FOR

// Синтаксис
for (Начало; Условие; Шаг) {
    // Тело цикла
    // Тут будет выполняться код
}
```

Пример:

```
// Пример
for (let num = 0; num < 5; num++) {
    console.log(num);
}

/*
Работа цикла for:
1) Выполняется начало - let num = 0
2) Выполняется условие - num < 5
3) Если условие true выполняется
    тело цикла - console.log(num)
4) Выполняется шаг - num++
Повтор начиная с пункта №2
*/
```

Break - Досрочно прекращает работу

```
// Директива break

let num = 0;
for (; num < 5; num++) {
    console.log(num);
    if (num == 2) break;
}
console.log(`Работа окончена, num = ${num}`);
```

Continue - Инструкция continue прерывает выполнение текущей итерации текущего или отмеченного цикла, и продолжает его выполнение на следующей итерации.

```
// Директива continue

let num = 0;
for (; num < 5; num++) {
    if (num == 2) continue;
    console.log(num);
}
```

Пример вывода всех элементов массива при помощи СИ подобного цикла:

```
let arr = [];

for (let num = 0; num < 3; num++){
    arr.push(Number(prompt('Array')));
}

console.log(arr);
```

Задача: Написать цикл, который выводит только положительные числа из массива
Задача: Написать цикл, который выводит только четные числа

Синтаксиса цикла от большего к меньшему

```
let arr = [];

for (let num = 10; num > 0; num--){
    arr.push(num);
}

console.log(arr);
```

Задача: Найти сумму всех элементов массива

```
let arr = [1, 2, 3, 4, 5];

let sum = 0;

for(let i = 0; i < arr.length; i++){
    sum = sum + arr[i];
}

console.log(sum);
```

Форма обратной связи:

https://airtable.com/shr7Lo4UqIWKhQzTR?prefill_lessons=recc1eciHM5YvPYek