

Тема урока: Массивы

Тип урока: Изучение нового материала

Структура урока:

Время	Что нужно сделать	Пояснения
5 минут	Организационная часть.	Отмечает присутствующих, вносит в журнал необходимую информацию. Отвечает на вопросы, если они есть. Повторение пройденного материала. Проверка и опрос по ДЗ.
25 минут	Теоретическая часть. Подготовка к практической работе.	Знакомство с массивами. Изучение основных возможностей работы с массивами.
5 минут	Перерыв.	Сквозное проветривание помещения.
40 минут	Практическая часть	Выполнение заданий.
5 минут	Подведение итогов занятия. Рефлексия	Подведение итогов занятия, обобщение материала, выдача ДЗ, пояснение ДЗ.

ХОД УРОКА

Вводное слово

Мы уже изучили числа и строки – типы данных, которые можно хранить и использовать в своих программах. Но одни лишь числа и строки – это как-то скучновато; не столь уж многое можно сделать со строкой как таковой. С помощью массивов JavaScript позволяет создавать и группировать данные более любопытными способами. А по сути своей массив – всего лишь список, где хранятся другие значения.

Теоретическая часть

Массив – это пронумерованная последовательность величин одинакового типа, обозначаемая одним именем. Элементы массива располагаются в последовательных ячейках памяти, обозначаются именем массива и индексом. Каждое из значений, составляющих массив, называется его компонентой (или элементом массива).

Индекс – это значение порядкового типа, определенного, как тип индекса данного массива.

Случайные числа – искусственно полученная последовательность реализаций случайной величины с заданным законом распределения.

Создание массива

Чтобы создать массив, используются квадратные скобки []. Фактически для задания пустого массива достаточно лишь пары квадратных скобок:

```
[];  
[]
```

Чтобы создать массив со значениями, нужно перечислить эти значения внутри квадратных скобок, разделяя их запятыми. Отдельные значения, хранящиеся в массиве, называют элементами. Пример:

```
var dinosaurs = [  
    "Тираннозавр",  
    "Велоцираптор",  
    "Стегозавр",  
    "Трицератопс",  
    "Брахиозавр",  
    "Птеранодон",  
    "Апатозавр",  
    "Диплодок",  
    "Компсогнат"  
];
```

Доступ к элементам массива

Чтобы получить доступ к элементам массива, используются квадратные скобки с индексом нужного элемента, как в этом примере:

```
dinosaurs[0];  
"Тираннозавр"  
dinosaurs[3];  
"Трицератопс"
```

Индекс – это номер элемента, в котором хранится значение. Аналогично символам в строке, первому элементу массива соответствует индекс 0, второму – 1, третьему – 2 и т. д. Поэтому, запросив индекс 0 в массиве `dinosaurs`, мы получили "Тираннозавр" (это первый элемент), а запросив индекс 3 – "Трицератопс" (четвертый элемент).

Создание и изменение элементов

Используя индекс в квадратных скобках, можно задавать или изменять значения элементов и даже добавлять новые элементы. Например:

```
dinosaurs[0] = "Тираннозавр рекс";
```

С помощью индексов также можно добавлять в массив элементы. Например:

```
var dinosaurs = [];
dinosaurs[0] = "Тираннозавр";
dinosaurs[1] = "Велоцираптор";
dinosaurs[2] = "Стегозавр";
dinosaurs[3] = "Трицератопс";
dinosaurs[4] = "Брахизавр";
dinosaurs[5] = "Птеранодон";
dinosaurs[6] = "Апатозавр";
dinosaurs[7] = "Диплодок";
dinosaurs[8] = "Компсогнат";

dinosaurs;
["Тираннозавр", "Велоцираптор", "Стегозавр", "Трицератопс", "Брахизавр", "Птеранодон", "Апатозавр", "Диплодок", "Компсогнат"]
```

Разные типы данных в одном массиве

Не обязательно, чтобы все элементы массива были одного типа. Например:

```
var dinosaursAndNumbers = [3, "динозавры", ["трицератопс", "стегозавр", 3627.5], 10];
```

Чтобы обратиться к элементам массива, вложенного в другой массив, нужно использовать вторую пару квадратных скобок. Например:

```
dinosaursAndNumbers[2];
["трицератопс", "стегозавр", 3627.5]
dinosaursAndNumbers[2][0];
"трицератопс"
```



Длина массива

Порой нужно знать, сколько в массиве элементов.

Для этого есть свойство `length` (длина), хранящее количество элементов в массиве.

Чтобы узнать длину массива, нужно просто добавить `.length` после его имени.

```
var maniacs = ["Якко", "Вакко", "Дот"];
maniacs[0];
"Якко"
maniacs[1];
"Вакко"
maniacs[2];
"Дот"
```

Чтобы узнать длину этого массива, нужно добавить `.length` к `maniacs`:

```
maniacs.length;  
3
```

Добавление элементов в массив

Чтобы добавить элемент к концу массива, можно воспользоваться методом `push`..

```
var animals = [];  
animals.push("Кот");  
1  
animals.push("Пес");  
2  
animals.push("Лама");  
3  
animals;  
["Кот", "Пес", "Лама"]  
animals.length;  
3
```

Запуск метода в программировании называется вызовом метода. При вызове метода `push` происходят две вещи. Во-первых, в массив добавляется элемент, указанный в скобках. Во-вторых, метод задает новую длину массива. Именно эти значения длины появляются в консоли после каждого вызова `push`.

Чтобы добавить элемент в начало массива, используется метод `.unshift(элемент)`:

```
animals;  
["Кот", "Пес", "Лама"]  
animals[0];  
"Кот"  
animals.unshift("Мартышка");  
4  
animals;
```

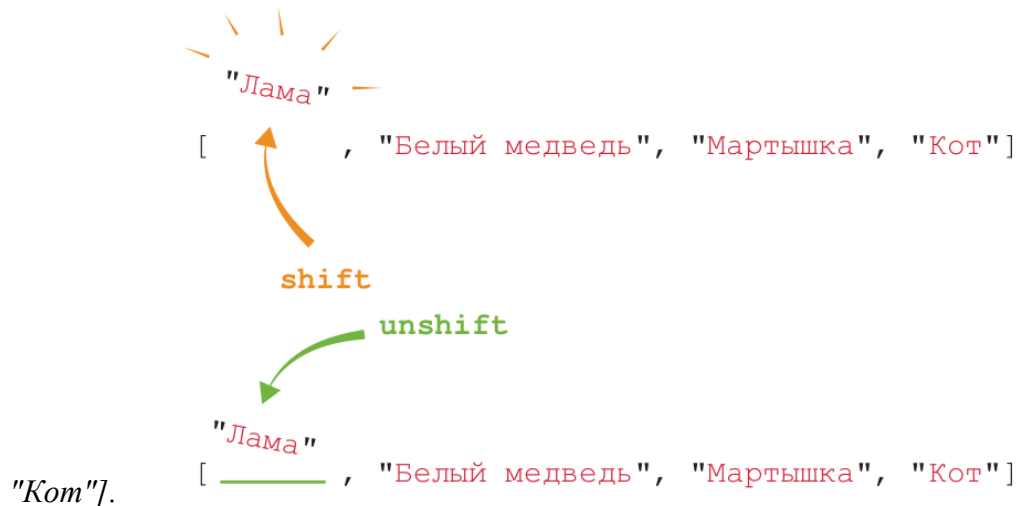
Удаление элементов массива

Убрать из массива последний элемент можно, добавив к его имени `.pop()`. Метод `pop` делает сразу два дела: удаляет последний элемент из массива и возвращает этот элемент в виде значения.

Чтобы удалить из массива первый элемент, вернув его значение, используется `.shift()`:

```
animals;  
["Лама", "Белый медведь", "Мартышка", "Кот"]  
var firstAnimal = animals.shift();  
firstAnimal;  
"Лама"  
animals;  
["Белый медведь", "Мартышка", "Кот"]
```

Метод `animals.shift()` работает аналогично `animals.pop()`, но элемент берется из начала массива. В начале этого примера массив `animals` имел вид `["Лама", "Белый медведь", "Мартышка", "Кот"]`. Вызов `.shift()` вернул первый элемент, "Лама", который мы сохранили в переменной `firstAnimal`. Поскольку `.shift()` не только возвращает элемент, но и удаляет его, в массиве `animals` осталось лишь `["Белый медведь", "Мартышка", "Кот"]`.



Объединение массивов

Пусть у нас есть два списка – список пушистых животных и список чешуйчатых животных – и мы хотим их объединить. Если поместить пушистых животных в массив `furryAnimals`, а чешуйчатых – в массив `scalyAnimals`, команда `furryAnimals.concat(scalyAnimals)` создаст новый массив, в начале которого будут элементы из первого массива, а в конце – из второго.

```
var furryAnimals = ["Альпака", "Кольцехвостый лемур", "Йети"];
var scalyAnimals = ["Удав", "Годзилла"];
var furryAndScalyAnimals = furryAnimals.concat(scalyAnimals);
furryAndScalyAnimals;
["Альпака", "Кольцехвостый лемур", "Йети", "Удав", "Годзилла"]
furryAnimals;
["Альпака", "Кольцехвостый лемур", "Йети"]
scalyAnimals;
["Удав", "Годзилла"]
```

Объединение нескольких массивов

Чтобы «склеить» два массива, создав таким образом новый массив, используется команда `firstArray.concat(otherArray)`. Метод `concat` создает массив, в котором элементы из `firstArray` будут расположены перед элементами из `otherArray`.

```
var furryAnimals = ["Альпака", "Кольцехвостый лемур", "Йети"];
var scalyAnimals = ["Удав", "Годзилла"];
var featheredAnimals = ["Ара", "Додо"];
var allAnimals = furryAnimals.concat(scalyAnimals, ←
featheredAnimals);
allAnimals;
["Альпака", "Кольцехвостый лемур", "Йети", "Удав", "Годзилла",
"Ара", "Додо"]
```

Поиск индекса элемента в массиве

Чтобы выяснить, какой у определенного элемента индекс в массиве, используется `.indexOf("элемент")`.

```
var colors = ["красный", "зеленый", "синий"];
colors.indexOf("синий");
2
colors.indexOf("зеленый");
1
```

Метод `indexOf` похож на квадратные скобки, только здесь все наоборот: команда `colors[2]` вернет "синий", а `colors.indexOf("синий")` вернет 2.

```
colors[2];
"синий"
colors.indexOf("синий");
2
```

Превращение массива в строку

Воспользовавшись методом `.join()`, можно соединить все элементы массива в одну большую строку.

```
var boringAnimals = ["Мартышка", "Кот", "Рыба", "Ящерица"];
boringAnimals.join();
"Мартышка, Кот, Рыба, Ящерица"
```

Что если нужно использовать в качестве разделителя не запятую?

```
var boringAnimals = ["Мартышка", "Кот", "Рыба", "Ящерица"];
boringAnimals.join(" - ");
"Мартышка - Кот - Рыба - Ящерица"
boringAnimals.join("*")
"Мартышка*Кот*Рыба*Ящерица"
boringAnimals.join(" и ")
"Мартышка и Кот и Рыба и Ящерица"
```

Случайный выбор

Использование `Math.random()`

Случайные числа можно генерировать с помощью специального метода `Math.random()`, который при каждом вызове возвращает случайное число от 0 до 1:

```
Math.random();
0.8945409457664937
Math.random();
0.3697543195448816
Math.random();
0.48314980138093233
```

Округление с помощью `Math.floor()`

И все же эти случайные значения нельзя использовать как индексы в массиве, поскольку индексы должны быть целыми числами, а не десятичными дробями. Чтобы исправить этот недостаток, нужен метод `Math.floor()`, округляющий число до ближайшего снизу целого значения (по сути, он просто отбрасывает все знаки после запятой).

```
Math.floor(3.7463463);
3
Math.floor(9.9999);
9
Math.floor(0.793423451963426);
0
```

Программа случайного выбора вариантов

Теперь можно создать массив с фразами, чтобы случайным образом выбирать их с помощью написанного ранее кода. Это и будет компьютерный «шар судьбы»! В комментариях указаны примеры вопросов, которые можно задать нашей программе.

```
var phrases = [
  "Звучит неплохо",
  "Да, это определенно надо сделать",
  "Не думаю, что это хорошая идея",
  "Может, не сегодня?",
  "Компьютер говорит нет"
];
// Мне выпить еще молочного коктейля?
phrases[Math.floor(Math.random() * 5)];
"Не думаю, что это хорошая идея"
// Мне пора делать домашнюю работу?
phrases[Math.floor(Math.random() * 5)];
"Может, не сегодня?"
```

Генератор случайных дразнилок

Можно усовершенствовать код выбора вариантов, создав программу, которая при каждом запуске генерирует случайную дразнилку!

```
var randomBodyParts = ["глаз", "нос", "череп"];
var randomAdjectives = ["воночая", "унылая", "дурацкая"];
var randomWords = ["муха", "выдра", "дубина", "мартышка", "крыса"];
// Выбор случайной части тела из массива randomBodyParts:
var randomBodyPart = randomBodyParts[Math.floor(Math.random() * 3)];
// Выбор случайного прилагательного из массива randomAdjectives:
var randomAdjective = randomAdjectives[Math.floor(Math.random() * 3)];
// Выбор случайного слова из массива randomWords:
var randomWord = randomWords[Math.floor(Math.random() * 5)];
// Соединяем случайные строки в предложение:
var randomInsult = "У тебя " + randomBodyPart + " словно " + ↵
randomAdjective + " " + randomWord + "!!!";
randomInsult;
"У тебя нос словно дурацкая выдра!!!"
```

Перерыв 5 минут.

Во время перерыва необходимо вывести всех ребят из аудитории. Открыть окна для сквозного проветривания. Нахождение преподавателя возле аудитории обязательно.

Практическая часть

После перерыва слушатели самостоятельно выполняют задания, преподаватель комментирует инструкции и помогает при необходимости.

1. Преобразуйте код дразнилки в более упрощенный.
2. Создайте массив и выведите все его элементы.
3. Создайте число и округлите его.
4. Дан массив вывести последний его элемент
5. Увеличьте дразнилку.

Подведение итогов занятия. Выдача домашнего задания и его разъяснение.

1. Что такое массив?
2. Что такое индекс?
3. Как узнать размер массива?