Arrays

This lesson will look at how to work with arrays in JavaScript. We will see how we can create arrays , access elements of an array and various other functionalities with arrays.

Introduction

*This is a quick revision of Javascript arrays for the sake of completion. If you understand Javascript arrays already, it will be **quite boring** for you.*

In most programming languages, an array is a linear collection of elements stored in a sequential order. We can iterate over arrays using array indices. Array indices typically start from 0 and go upto Array Length - 1. Such arrays are known as zero-based arrays. Javascript arrays are zero based arrays as well.

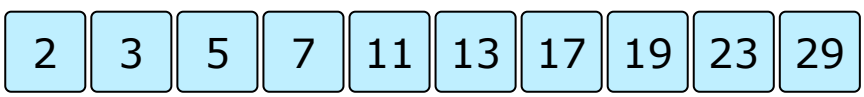
However, one unique thing about Javascript arrays is that they are actually objects and indices are just property names in that object. Indices are usually numbers and so Javascript arrays convert these numerical indices to string keys and then assign them as object properties. Therefore, if you access array[0], you are actually accessing property "0" of object array.

Here's how we can create an array in javascript.

```
var topics = [];
```

Initializing the array

You can also initialize the array by passing in the values when declaring the array. e.g. let's create an array of the first te.n prime numbers.



First 10 prime numbers

```
var primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29];
```

Adding elements to an existing array

Once you have created the array, you can add more elements to the array using **push** method.

```
var topics = [];
topics.push('Arrays');
topics.push('Stacks');
topics.push('Queues');
```

Length of the array

You can get the length of the array using the '**length**' property.

```
var topics = [];
console.log('[' + topics + '] has length = ' +
            topics.length);

topics.push('Arrays');
topics.push('Stacks');
topics.push('Queues');

console.log('[' + topics + '] has length = ' +
            topics.length);
```

Accessing Array Elements

Array elements are accessed using the indices ranging from 0 to length -1. Here's some code.

```
var topics = [];

topics.push('Arrays');
topics.push('Stacks');
topics.push('Queues');

for (var i = 0; i < topics.length; i++) {
  console.log(topics[i]);
```

```
}
```

The 'new' Array Syntax

Arrays can also be created with the new keyword in JavaScript as array is an object. In most cases, this way of creating arrays in inefficient and should be avoided. However, for learning, here's the syntax.

```
var topics = new Array('Arrays', 'Stacks', 'Quotes');
```

Let's have a quick quiz

Quiz 1

Q

Array ‘tutorial’ is defined as

```
var tutorial=['Arrays', 'Stacks', 'Queues'];
```

What’s the correct syntax to replace ‘Queues’ with ‘Linked Lists’?

Check Answers

Removing elements from an array

Elements of an existing array can be removed or permanently deleted by using the splice method. Splice method requires two arguments at the minimum, the index of first element to delete and the number of elements to delete starting from the given index.

```
var tutorials = ['Arrays', 'Stacks', 'Queues'];
console.log('Before Splice: ' + tutorials);
tutorials.splice(1,1);
console.log('After Splice: ' + tutorials);
```