JavaScript Essentials

Arrays

Table of Contents

Lesson Objectives

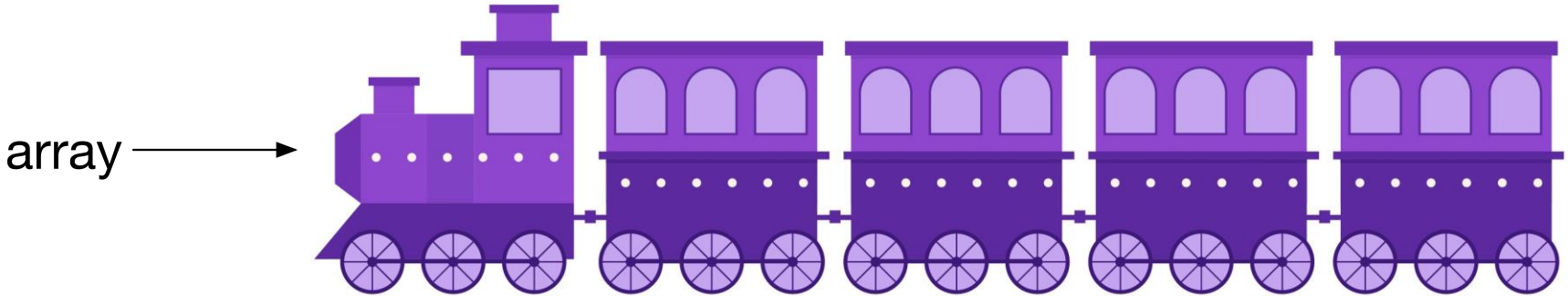
- Understand array – a neat way of storing a list of data items
- Able to create an array, retrieve, add and remove items stored in an array

Section 1

Overview – What is an array?

Overview – What is an array?

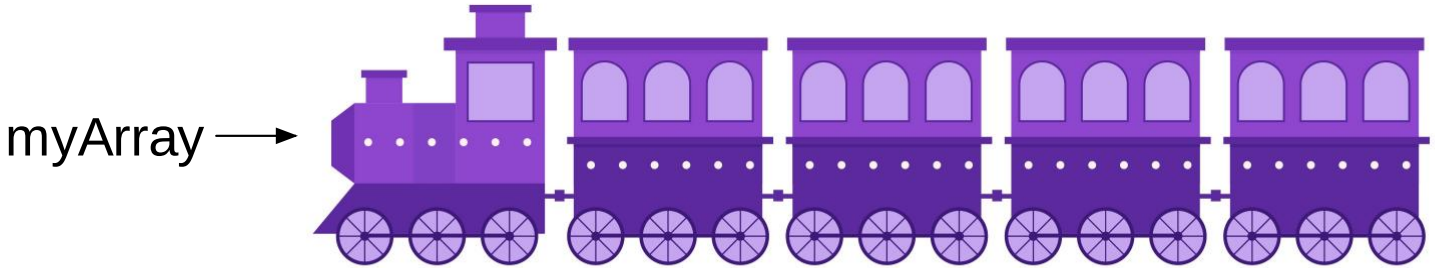
- Arrays are generally described as “list-like objects”
- Think of Array as a train of data:



Overview – What is an array?

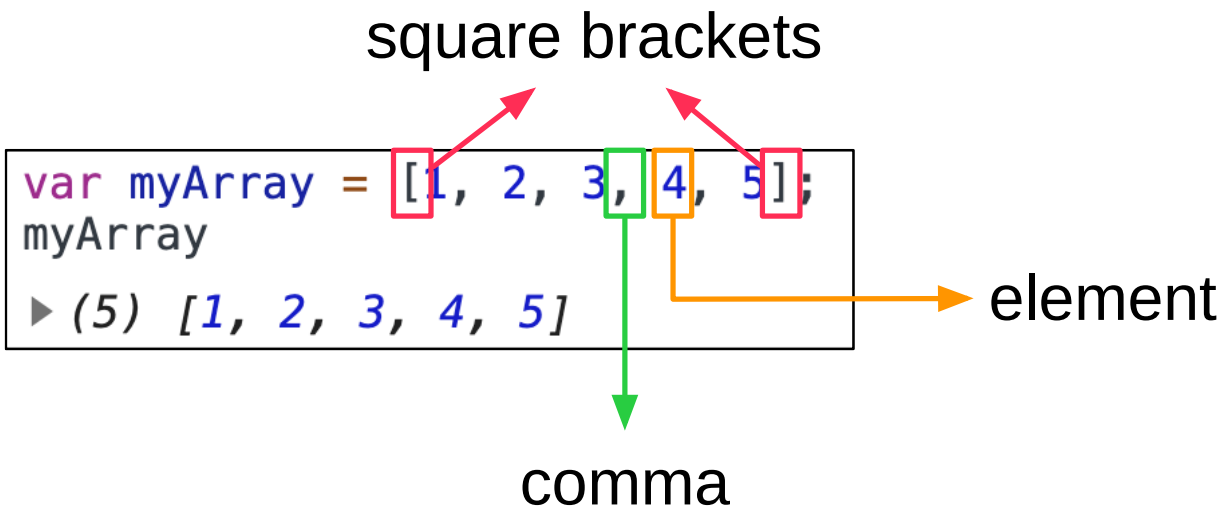
- Array objects can be stored in variables and dealt with in much the same way as any other type of value

```
var myArray = [1, 2, 3, 4, 5];
myArray
▶ (5) [1, 2, 3, 4, 5]
```



Overview – Create an array

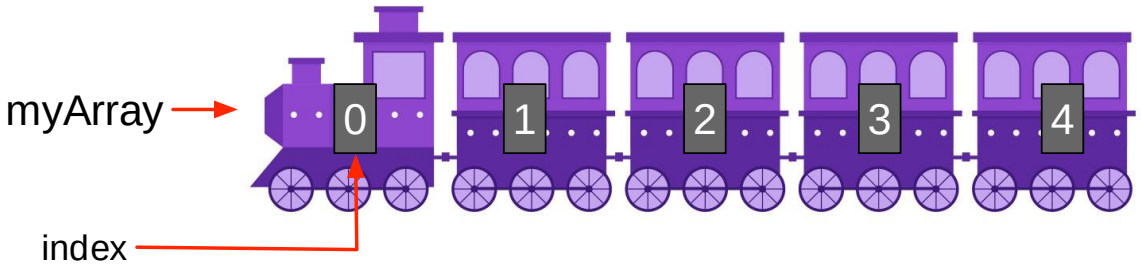
- Syntax: Arrays consist of square brackets and elements that are separated by commas.



Overview – Access and modify items

- Syntax: access individual items in the array using bracket
- **Computers start counting from 0!**

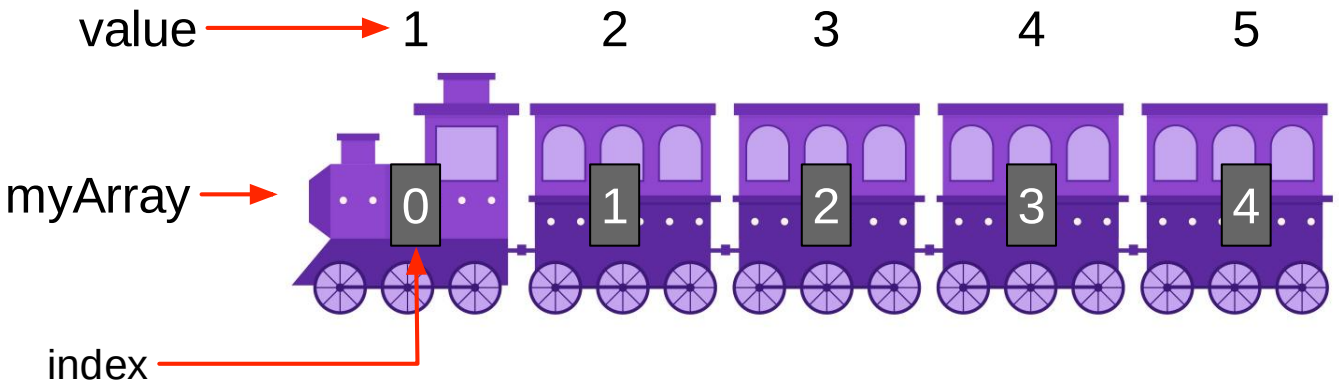
```
var myArray = [1, 2, 3, 4, 5];
myArray
▶ (5) [1, 2, 3, 4, 5]
```



Overview – Access and modify items

- Syntax: access individual items in the array using bracket
- To access 2nd element use: myArray[1];

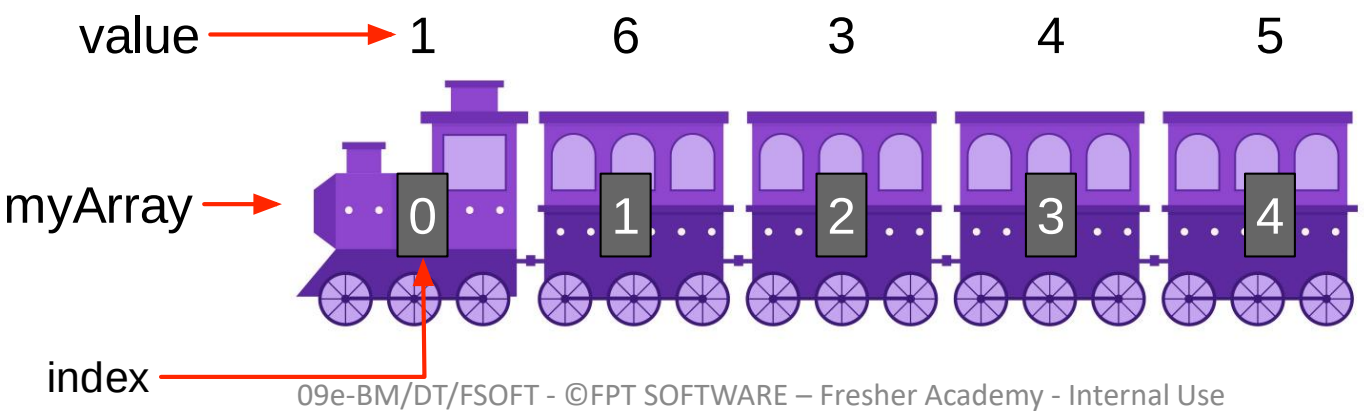
```
var myArray = [1, 2, 3, 4, 5];
myArray[1];
2
```



Overview – Access and modify items

- Syntax: access individual items in the array using bracket
- To assign new value 2nd element use: myArray[1] = 6;

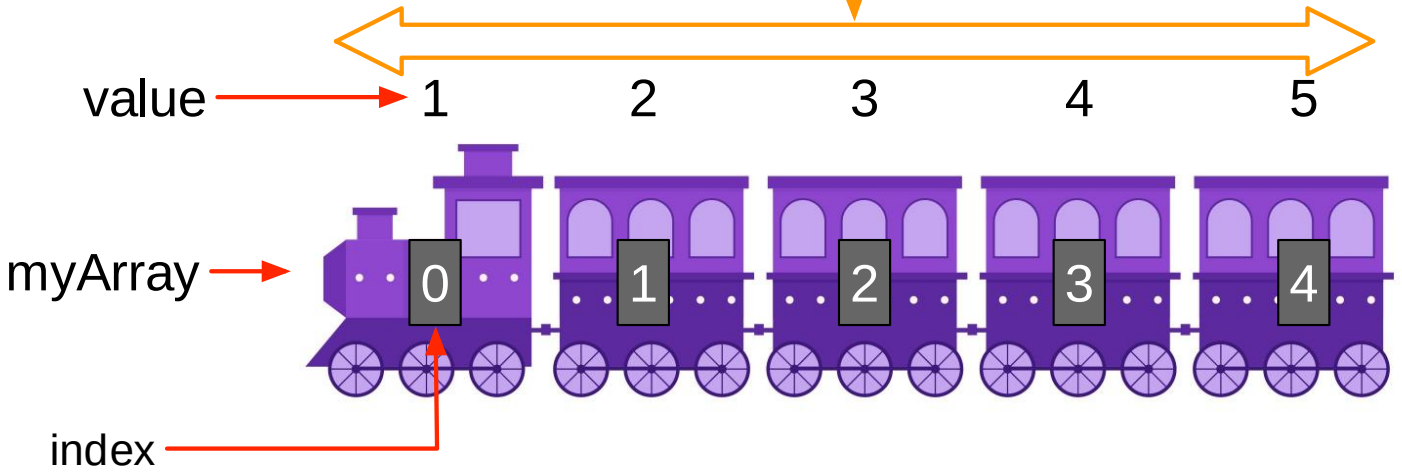
```
var myArray = [1, 2, 3, 4, 5];
myArray[1] = 6;
myArray;
▶ (5) [1, 6, 3, 4, 5]
```



Overview – Find the length of an array

- Syntax: access by using the **length** property

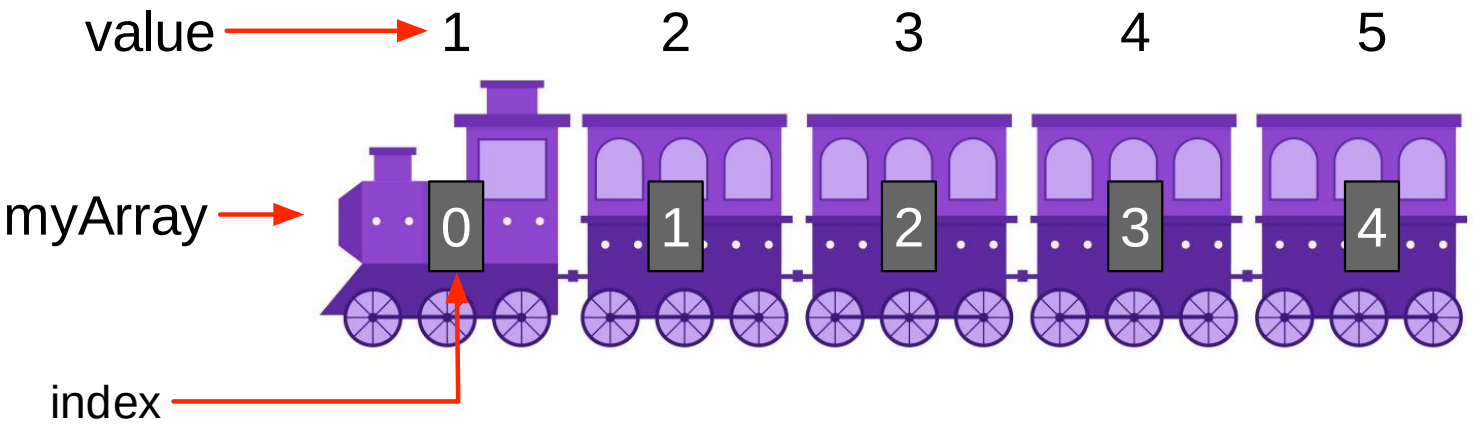
```
var myArray = [1, 2, 3, 4, 5];
myArray.length;
5
```



Overview – Find the length of an array

- Syntax: access last item by using the **length - 1**

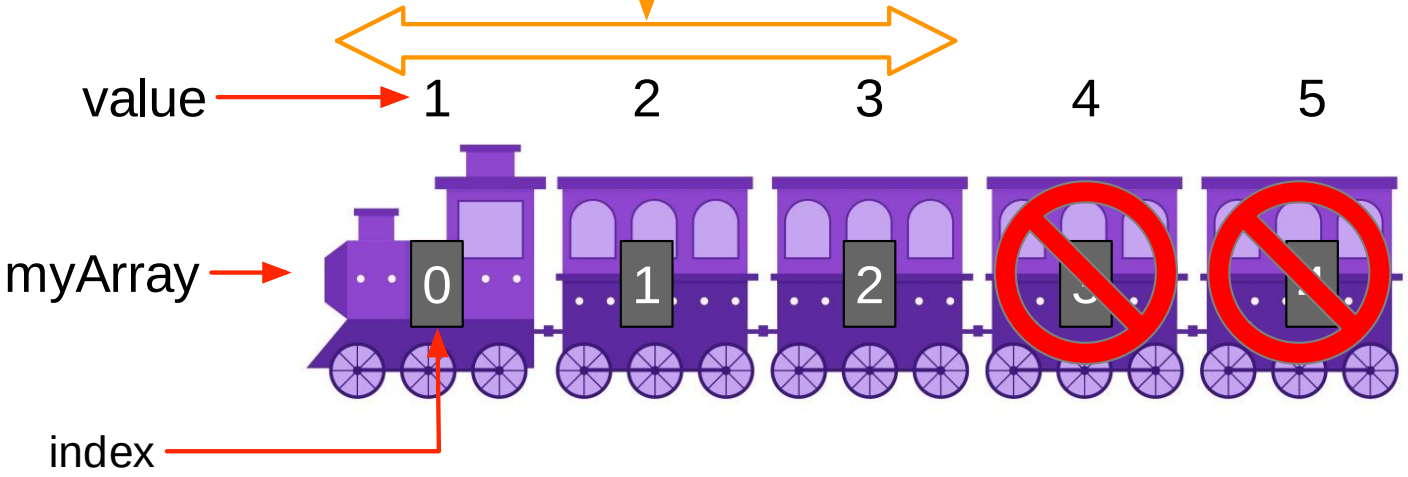
```
var myArray = [1, 2, 3, 4, 5];
myArray[myArray.length - 1];
5
```



Overview – Find the length of an array

- Syntax: reduce **length** will shrink the array

```
var myArray = [1, 2, 3, 4, 5];
myArray.length = 3;
myArray;
▶ (3) [1, 2, 3]
```



Overview – Summary

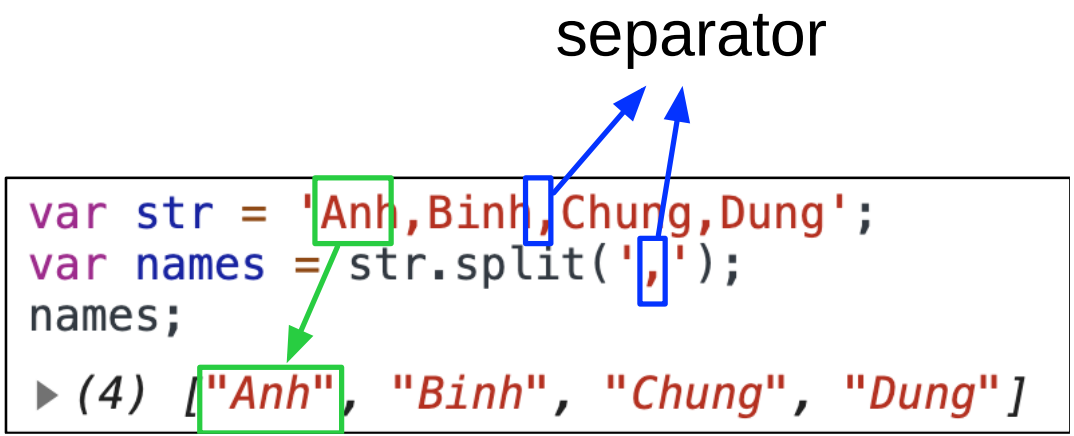
- **Array** provide a neat way of storing a list of data items under a single variable
- **Arrays** are generally described as "list-like objects"; they are basically single objects that contain multiple values stored in a list
- **Array** objects can be stored in variables and dealt with in much the same way as any other type of value
- The difference being that we can access each value inside the list individually
- **Always** remember: Computer starts at 0 (not 1)

Section 2

Useful Array methods

Useful methods – Convert string to array

- Convert string to array with **split()** method



Useful methods – Convert string to array

- Practice with below string:

```
var myData = 'Manchester,London,Liverpool,Birmingham,Leeds,Carlisle';
```

- Do the following:
  1. Convert to array string
  2. Retrieve the 1st, 3rd and last element
  3. Remove last element
  4. Join back the using .join() on Array

Useful methods – Add or remove items

- First of all, to add or remove an item at the end of an array we can use push() and pop() respectively.

```
var myArray = [1, 2, 3, 4, 5];

var p = myArray.push(6);
// add and return value of item was added
console.log(myArray, p);
```

```
▶ (6) [1, 2, 3, 4, 5, 6] 6
```

```
var myArray = [1, 2, 3, 4, 5];

var last = myArray.pop(); // remove last
console.log(myArray, last);
```

```
▶ (4) [1, 2, 3, 4] 5
```

Useful methods – Add or remove items

- Practice using array from previous session:
  1. Using push() to add ’Cardiff’
  2. Using push() to add 'Bradford', 'Brighton’ at same time
  3. Check the length of array after the method call completes
  4. Removing the last item from the array and save it to a variable

Useful methods – Add or remove items

- unshift() and shift() work in exactly the same way as push() and pop(), respectively, except that they work on the beginning of the array, not the end.

```
var myArray = [1, 2, 3, 4, 5];

var p = myArray.unshift(6);
// same like push be add to the beginning
console.log(myArray, p);
```

```
▶ (6) [6, 1, 2, 3, 4, 5] 6
```

Useful methods – Add or remove items

- unshift() and shift() work in exactly the same way as push() and pop(), respectively, except that they work on the beginning of the array, not the end.

```
var myArray = [1, 2, 3, 4, 5];

var first = myArray.shift();
// same like pop be remove from the beginning
console.log(myArray, first);
```

```
▶ (4) [2, 3, 4, 5] 1
```

Useful methods - Summary

- Use strings.split to convert a strings into an array
- To add and remove item (at the end) use push and pop
- To add and remove item (at the beginning) use shift and unshift
- **Note:** shift and unshift might affect the performance of the program

Section 3

Practice time

Practice 1: Arrays manipulation

- Underpants — $6.99
- Socks — $5.99
- T-shirt — $14.99
- Trousers — $31.99
- Shoes — $23.99

Total: $83.95

Practice 2: Arrays manipulation

Thank you

Q&A