

JAVASCRIPT-CORE-S7-Arrays

JS session 7 Arrays C10 TR

Training Clarusway

Pear Deck - February 3, 2022 at 7:30PM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1



Use this space to take notes:

Slide 2

Table of Contents



- ▶ Arrays Definition
- ▶ Array Properties & Methods

CLARUSWAY
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 3

Your Response



Did you finish Javascript Core pre-class material?

YES NO



Students choose an option

Pear Deck Interactive Slides

Use this space to take notes:

Slide 4

Play Kahoot!

CLARUSWAY
WAY TO REINVENT YOURSELF

Link(s) on this slide:

- <https://create.kahoot.it/details/8-arrays/5606e44b-f3d9-465f-9a20-c33353271dbf>

Use this space to take notes:

Slide 5

1 Arrays Definition



CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 6

▶ Arrays Definition



The array is a single variable in JavaScript that is used to store various elements. It is often used when a list of elements is stored and accessed by a single variable. Unlike most languages where the array is a multiple variable relations, there is a single variable in JavaScript array that stores multiple elements.



Spaces and line breaks are not important. A declaration can span multiple lines.



You can also generate an array and assigns values to it in the following example:

CLARUSWAY
WAY TO REINVENT YOURSELF



```
var colors = ["Red", "Blue",  
             "Yellow"];
```

```
var colors  
= [  
  "Red",  
  "Blue",  
  "Yellow"  
];
```

8

Use this space to take notes:

Slide 7



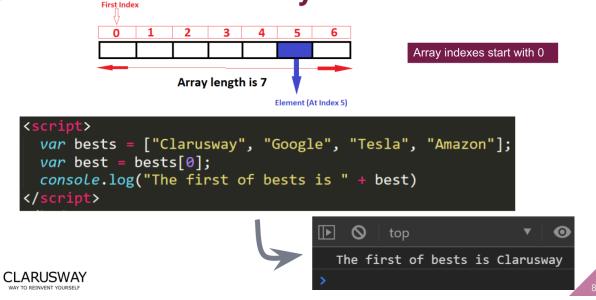
▶ Access the Array of Elements

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 8

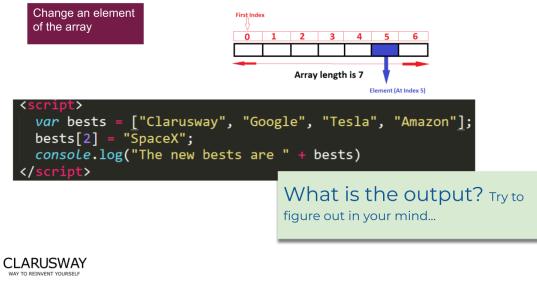
▶ Access the Array of Elements ➤



Use this space to take notes:

Slide 9

▶ Access the Array of Elements ➤

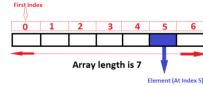


Use this space to take notes:

Slide 10

▶ Access the Array of Elements ➤

Change an element
of the array



```
<script>
  var bests = ["Clarusway", "Google", "Tesla", "Amazon"];
  bests[2] = "SpaceX";
  console.log("The new bests are " + bests)
</script>
```

CLARUSWAY



10

Use this space to take notes:

Slide 11



3 ▶ Arrays are Objects

CLARUSWAY

Use this space to take notes:

Slide 12

▶ Arrsys are Objects



Arrays are a particular type of object. The JavaScript type operator returns "object" for arrays.



However, it is better to define JavaScript arrays like arrays

Objects are using names
to control their "members"

Array
var name = ["John", "James", 24];

Object
var name = {firstName:"John", lastName:"James", age:24};

CLARUSWAY
WAY TO REINVENT YOURSELF

12

Use this space to take notes:

Slide 13



4 The *length* Property

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 14

► The *length* Property



The length property returns the number of elements.

```
<script>
  var names = ["John", "James", "Walter", "Edward"];
  console.log(names.length);
</script>
```



```
[x] ⓘ top
4
>
```

14

Use this space to take notes:

Slide 15



5 ► The *concat()* Method

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 16

► The `concat()` Method



The `concat()` method is used to combine two or more arrays.



This method does not modify the current arrays but returns a new array that contains the elements of the joined arrays

```
<script>
  var dogs = ["Bulldog", "Beagle", "Rottweiler"];
  var cats = ["Ragdoll", "Sphynx", "Birman"];
  console.log(dogs.concat(cats));
  var birds = ["Parrot", "Budgerigar", "Cockatoos"];
  console.log(cats.concat(dogs,birds));
</script>
```

▶ (6) ["Bulldog", "Beagle", "Rottweiler", "Ragdoll", "Sphynx", "Birman"]
▶ (9) ["Ragdoll", "Sphynx", "Birman", "Bulldog", "Beagle", "Rottweiler", "Parrot", "Budgerigar", "Cockatoos"]

CLARUSWAY
WAY TO REINVENT YOURSELF



16

Use this space to take notes:

Slide 17



6 ► The `sort()` Method

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 18

► The `sort()` Method



The `sort()` method sorts arrays alphabetically

```
<script>
  var names = ["John", "James", "Aaron", "Walter", "Edward"];
  console.log(names.sort());
</script>
```



CLARUSWAY

```
] ⓘ top ▾ ⓘ Filter
▶ (5) ["Aaron", "Edward", "James", "John", "Walter"]
```

18

Use this space to take notes:

Slide 19

► The `sort()` Method



The `sort()` method **doesn't** sorts numbers in arrays

```
<script>
  var numbers = [45, 16, 78, 8, 25];
  console.log(numbers.sort());
</script>
```



CLARUSWAY

```
] ⓘ top ▾ ⓘ Filter
▶ (5) [16, 25, 45, 78, 8]
>
```

19

Use this space to take notes:

Slide 20



7 The *push()* Method

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 21

▶ The *push()* Method



 **push()** The *push()* method adds a new element to the end of an array.

```
<script>
  var names = ["John", "James", "Aaron", "Walter", "Edward"];
  names.push("McCarthy") // Adds new element to name
  console.log(names);
</script>
```



21

Use this space to take notes:

Slide 22



7 The *pop()* Method

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 23

▶ The *pop()* Method



 **pop()** The *pop()* method allows you to remove the last element from an array

```
<script>
  var names = ["John", "James", "Aaron", "Walter", "Edward"];
  names.pop() // Removes last element from name
  console.log(names);
</script>
```

CLARUSWAY
WAY TO REINVENT YOURSELF

23

Use this space to take notes:

Slide 24

► The *pop()* Method

 **pop()** The *pop()* method allows you to remove the last element from an array

```
<script>
var names = ["John", "James", "Aaron", "Walter", "Edward"];
name = names.pop() // Removed element can assign
console.log(name);
console.log(names);
</script>
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Edward
► (4) ["John", "James", "Aaron", "Walter"]

24

Use this space to take notes:

Slide 25



8 ▶ *The shift()* and *unshift()* Methods

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 26

► ***shift()* Method**



The `shift()` method removes the first element from an array and returns the string that removed element

```
<script>
  var names = ["John", "James", "Aaron", "Walter"];
  shiftedName = names.shift();
  console.log(shiftedName);
  console.log(names);
</script>
```



```
John
▶ (3) ["James", "Aaron", "Walter"]
```

CLARUSWAY
WAY TO REINVENT YOURSELF



28

Use this space to take notes:

Slide 27

► ***unshift()* Method**



The `unshift()` method adds a new element to the beginning of an array and returns the new array length.

```
<script>
  var names = ["James", "Aaron", "Walter"];
  unshiftedName = names.unshift("John");
  console.log(unshiftedName);
  console.log(names);
</script>
```



```
4
▶ (4) ["John", "James", "Aaron", "Walter"]
```

CLARUSWAY
WAY TO REINVENT YOURSELF



27

Use this space to take notes:

Slide 28



9

► *splice()* Method

CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 29

► *splice()* Method



The *splice()* method is used to add new elements to an array and delete elements from an array

```
<script>
  var names = ["James", "Aaron", "Walter"];
  names.splice(1, 2, "John", "Edward");
  console.log(names);
</script>
```



► (3) ["James", "John", "Edward"]



- The first parameter (1) determines the position of the first element to delete and starting position to insert.
- The second parameter (2) determines the number of elements to delete.
- The remaining parameters ("John", "Edward") determines the new elements to be added.

CLARUSWAY
WAY TO REINVENT YOURSELF

29

Use this space to take notes:

Slide 30

► ***splice()* Method**

```
<script>
  var names = ["James", "Aaron", "Walter"];
  names.splice(1, 3, "John", "Edward", "McCarthy");
  console.log(names);
</script>
```



The screenshot shows a browser's developer tools console window. At the top, there are icons for zoom, refresh, and filter, followed by the word 'top'. Below that, a dropdown menu shows '(4)'. The main area of the console displays the array: `["James", "John", "Edward", "McCarthy"]`.

CLARUSWAY
WAY TO REINVENT YOURSELF

30

Use this space to take notes:

Slide 31



10 ► ***slice()* Method**



CLARUSWAY
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 32

► slice() Method



The slice() method slices a piece of an array into a new array.



The original array will not be modified.

```
<script>
  var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
  var citrus = fruits.slice(1, 3);
  console.log(citrus);
  console.log(fruits)
</script>
```

What is the output? Try to figure out in your mind...

CLARUSWAY
WAY TO REINVENT YOURSELF

- The first parameter (1) determines the starting position of the new array.
- The second parameter (3) determines the ending (not included end) position of the new array.

32

Use this space to take notes:

Slide 33

► slice() Method



The slice() method slices a piece of an array into a new array.



The original array will not be modified.

```
<script>
  var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
  var citrus = fruits.slice(1, 3);
  console.log(citrus);
  console.log(fruits)
</script>
```

↳ ▶ (2) ["Orange", "Lemon"]
▶ (5) ["Banana", "Orange", "Lemon", "Apple", "Mango"]

CLARUSWAY
WAY TO REINVENT YOURSELF

- The first parameter (1) determines the starting position of the new array.
- The second parameter (3) determines the ending (not included end) position of the new array.

33

Use this space to take notes:

Slide 34



11 *indexOf()* and *lastIndexOf()* Method

CLARUSWAY
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 35

► *indexOf()* Methods



The *indexOf()* method returns the position of a specified element in an array at the first occurrence

This method returns -1 if the element is not found

```
1 | var colors = ["Red", "Yellow", "Green", "Blue", "Pink", "Green"];
2 | var x = colors.indexOf("Green", 3);
3 |
4 | console.log(x);
```

Output

```
5
```

- The first parameter ("Green") determines the item to search for.
- The second parameter (3) determines the starting position of the search.

CLARUSWAY
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 36

▶ *lastIndexOf()* Methods

○ The *lastIndexOf()* method returns the last index of a specified element in an array and returns -1 if the element is not found.

○ In this method, the search begins at the stated position, or at the end if no starting position is specified, and the search ends at the beginning of the array.

```
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango", "Banana", "Orange", "Apple"];
var last = fruits.lastIndexOf("Banana");
console.log(last);
</script>
```

What is the output? Try to figure out in your mind...

CLARUSWAY
WAY TO REINVENT YOURSELF

38

Use this space to take notes:

Slide 37

▶ *lastIndexOf()* Methods

○ The *lastIndexOf()* method returns the last index of a specified element in an array and returns -1 if the element is not found.

○ In this method, the search begins at the stated position, or at the end if no starting position is specified, and the search ends at the beginning of the array.

```
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango", "Banana", "Orange", "Apple"];
var last = fruits.lastIndexOf("Banana");
console.log(last);
</script>
```



CLARUSWAY
WAY TO REINVENT YOURSELF

37

Use this space to take notes:

Slide 38

► *lastIndexOf()* Methods



The *lastIndexOf()* method returns the last index of a specified element in an array and returns -1 if the element is not found.



In this method, the search begins at the stated position, or at the end if no starting position is specified, and the search ends at the beginning of the array.

```
<script>
var fruits = ["Banana", "Orange", "Mango", "Apple", "Mango", "Banana", "Orange", "Apple"];
var last = fruits.lastIndexOf("Apple", 4);
console.log(last);
</script>
```

What is the output? Try to figure out in your mind...

CLARUSWAY
WAY TO REINVENT YOURSELF

38

Use this space to take notes:

Slide 39

► *lastIndexOf()* Methods

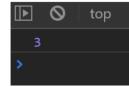


The *indexOf()* method returns the last index of a specified element in an array and returns -1 if the element is not found.



In this method, the search begins at the stated position, or at the end if no starting position is specified, and the search ends at the beginning of the array.

```
<script>
var fruits = ["Banana", "Orange", "Mango", "Apple", "Mango", "Banana", "Orange", "Apple"];
var last = fruits.lastIndexOf("Apple", 4);
console.log(last);
</script>
```



39

Use this space to take notes:

Slide 40

THANKS!
Any questions?



CLARUSWAY
WAY TO REINVENT YOURSELF

40

Use this space to take notes: