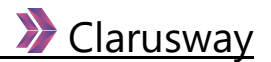


Hands-on JavaScript Arrays



Purpose of the this hands-on training is to teach the students JavaScript Arrays.

Learning Outcomes

At the end of the this hands-on training, students will be able to use;

- Arrays
- Array **length** Property
- Array Methods

Outline

- What is an Array?
- Defining an Array
- Defining an Array with **new** Keyword
- Access the Elements of an Array
- Changing an Array Element
- Access the Full Array
- Arrays are Special type of Objects
- Array Properties and Methods

Part 1 - What is an Array?

- In JavaScript, array is a single variable that is used to store different elements. It is often used when we want to store list of elements and access them by a single variable.

Part 2 - Defining an Array?

Syntax :

```
var nameOfArray = [item1, item2, ...];
```

Let's give an Array example:

Part 3 - Defining an Array with **new** Keyword

- You can initialize an array with Array constructor syntax using new keyword.

Syntax :

```
var arrayName = new Array(element1, element2, element3, ...elementN);
```

¶ The two examples above do exactly the same. There is no need to use `new Array()`. For **simplicity**, **readability** and **execution speed**, use the first one (the array literal method).

Part 4 - Accessing Elements of an Array

You access an array element with index number.

Syntax :

```
var element = nameOfArray[indexNumber];
```

¶ Array indexes start with 0.

Part 5 - Changing an Array Element

You can change an element by giving a new value.

Syntax :

```
nameOfArray[indexNumber] = "newElement";
```

Part 6 - Accessing Full Array

- You can access the full array with array name.

Syntax :

```
console.log(nameOfArray);
```

Part 7 - Arrays are Special type of Objects

- The typeof operator in JavaScript returns "object" for arrays.

- How to check if a variable is an array in JavaScript?

Syntax :

```
Array.isArray(value);
```

Part 8 - Array Properties and Methods

- The real strength of JavaScript arrays are the built-in array properties and methods.
- The `length` Property

- Accessing the Last Array Element

- The `push()` Method

- Also new element can be added to an array using `length` property

- The `toString()` Method

`toString()` method converts an array to a string of (comma separated) array values.

- The `join()` Method

The `join()` method is used to join the elements of an array into a string. The elements of the string will be separated by a specified separator and its default value is a comma(,).

- The `pop()` Method

Remove an item from the end of an array with `pop()` method.

`pop()` returns the removed item.

- The `push` Method

Add items to the end of the an array

`push()` returns the new array length.

- The `shift()` Method

Remove an item from the beginning of an array

`shift()` returns the removed item.

- The `unshift()` Method

Add items to the beginning of an array.

`unshift()` returns the new array length.

- Deleting Elements

Since JavaScript arrays are objects, elements can be deleted by using the JavaScript operator `delete`.

⚠ Using `delete` may leave undefined holes in the array. Use `pop()` or `shift()` instead.

- The `splice()` Method

The `splice()` method can be used to add new items to an array.

First parameter of `splice` method defines the position where new elements should be added.

Second parameter of `splice` method defines how many elements should be removed.

Rest of the parameters define the new items to be added.

- The `concat()` Method

The `concat()` method creates a new array by merging (concatenating) existing arrays:

⚠ The `concat()` method does not change the existing arrays. It always returns a new array.

The `concat()` method can take any number of array arguments:

- The `slice()` Method

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

The `slice()` method can take two arguments like `slice(2, 4)`.