Arrays

ARRAYS

- An array is an ordered list of values
- Its created using square brackets:

const myArray = [];

- To access a specific value in an array, we write its position in the array (index).
- To assign a specific value to an element in an array, we assign by specifying it's index

array[index] = value;

- We can create an array literal using square brackets that already contain some

const avengers = ['Captain America', 'Iron Man', 'Thor', 'Hulk'];

Removing Values from Arrays

The delete operator will remove an item from an array but the value will be replaced by a

value of undefined:

delete array[index];

Destructuring an array is the concept of taking values out of an array and presenting them as individual values.

```
const [x,y] = [1,2];
    x
      << 1
    y
      << 2
```

Array Properties and Methods

- To find the length of an array, we can use the length property

array.length

- To remove the last item from an array, we can use the pop() method. The method returns the last item of the array, but the array no longer contains that item.

array.pop()

- The shift() method works in a similar way to the pop() method, but this removes the first item in the array: *array.shift()*

- The push method appends a new value to the end of the array. The return value is the new length of the array: *array.push(value)*

- The unshift method is similar to the push() method, but this appends a new item to the beginning of the array: array.unshift(value)

- The concat method can be used to merge an array with one or more arrays

  array1.concat(array2)

- The join method can be used to turn the array into a string that comprises all the items in the array, separated by a named character, if not specified, a comma will be used:

  array.join(' & ');

  << 'value 1 & Value 2 & Value 3

- The slice method creates a subarray, effectively chopping out a slice of an original array, starting at one position and finishing at another: *array.slice(startIndex,endIndex);*

- The splice method removes items from an array then inserts new items in their place.

  array.splice(startIndex,howMany,'replacement-value');

- We can reverse the order of an array using the reverse method.

  array.reverse();

- We can sort the order of an array using the sort method.

  array.sort();

- ES6 also introduced the includes() method. This returns a Boolean value depending on whether the array contains a particular element or not:

  array.includes(value);

Multidimensional Arrays (array of arrays)

- To access the values in a multidimensional array, we use two indices: one to refer to the item's place in the outer array, and one to refer to its place in the inner array.

- The spread operator that we met earlier can be used to flatten multi-dimensional arrays.

SETS

- Set is a data structure that represents a collection of unique values.

- An empty set is created using the new operator

  const list = new Set();

- Values can be placed into a set using the add method: ***list.add(value);***

- All non-primitive values, such as arrays and objects, are considered unique values, even if they contain the same values.

Set Properties and Methods

- The number of values in a set can be found using the size method
- The clear() method can be used to remove all values from a set
- The has() method can be used to check if a value is in a set. This returns a boolean value of true or false.
- The delete method can be used to remove a value from a set. This returns a boolean value of true if the value was removed from the set, or false if the value wasn't in the set and couldn't be removed
- A set can be converted into an array by placing the set, along with the spread operator directly inside an array literal.

Additional Notes

- A memory leak occurs when a program retains references to values that can no longer be accessed in its memory.
- Weak sets avoid this situation by garbage collecting any references to a "dead object" that's had its original reference removed.
- To create a weak set, the new operator and the WeakSet() constructor in the same way that we created a set:

```
const weak = new WeakSet();
```