Array Access and Modification

Where we learn about the array methods of **at()**; **copyWithin()**; **fill()**; **pop()**; **push()**; **reverse()**; **shift()**; **splice()**; and **unshift()**

Array Access and Modification: at()

The **at()** method returns an indexed element from an array. According to MDN Web Docs, the **at()** method is equivalent to the bracket notation when index is non-negative. For example, *array[0]* and *array.at(0)* both return the first item.

The syntax for **at()**is as follows:

array.at(index)

Array Access and Modification: copyWithin()

The **copyWithin()** method is a built-in JavaScript array method that allows you to copy a sequence of elements within an array and paste them at a different location within the same array.

The syntax for **copyWithin()** is as follows:

array.copyWithin(target, start, end)

- target (required): The index at which to paste the copied elements.
- start (optional): The index at which to start copying elements from.
- end (optional): The index at which to stop copying elements from (exclusive). If not provided, it will default to the length of the array.

Array Access and Modification: fill()

The **fill()** method is a built-in JavaScript array method that allows you to fill all or a portion of an array with a static value.

The syntax for **fill()** is as follows:

array.fill(value, start, end)

- value (required): The value to fill the array with.
- start (optional): The index at which to start filling elements. If not provided, it will default to 0.
- end (optional): The index at which to stop filling elements (exclusive). If not provided, it will default to the length of the array.

Array Access and Modification: pop()

The **pop()** method is a built-in JavaScript array method that removes the last element from an array and returns that element.

The syntax for **pop()** is as follows:

array.pop()

Array Access and Modification: push()

The **push()** method is a built-in JavaScript array method that adds one or more elements to the end of an array and returns the new length of the array.

The syntax for **push()** is as follows:

array.push(element1, element2)

Array Access and Modification: reverse()

The **reverse()** method is a built-in JavaScript array method that reverses the order of the elements in an array. It modifies the original array in-place.

The syntax for **reverse()**is as follows:

array.reverse()

Array Access and Modification: shift()

The **shift()** method is a built-in JavaScript array method that removes the first element from an array and shifts all remaining elements to a lower index, effectively reducing the length of the array by one.

The syntax for **shift()**is as follows:

array.shift()

Array Access and Modification: splice()

The **splice()** method is a built-in JavaScript array method that allows you to modify an array by adding, removing, or replacing elements. It can be used to add or remove elements at any position within the array. The **splice()** method modifies the original array in-place and returns the removed elements as a new array.

The syntax for **splice()**is as follows:

array.splice(startIndex, deleteCount, item1, item2, ..., itemN)

- startIndex (required): The index at which to start modifying the array.
- deleteCount (optional): The number of elements to remove starting from the startIndex. If set to 0, no elements are removed.
- item1, item2, ..., itemN (optional): Elements to be added to the array at the startIndex.

Array Access and Modification: unshift()

The **unshift()** method is a built-in JavaScript array method that adds one or more elements to the beginning of an array.

The syntax for **unshift()**is as follows:

array.unshift(element1, element2, element3)