1. Using the shift() Method to Remove the First Element of an Array

The shift() method removes the first element from an array and shifts all other elements one position to the left. It returns the element that was removed.

Example:

```
let pets = ["dog", "cat", "bird"];
let removedPet = pets.shift();  // Removes the first element, "dog"
console.log(pets);  // ["cat", "bird"]
console.log(removedPet);  // "dog"
```

- **Before**: ["dog", "cat", "bird"]
- **After**: The first element "dog" is removed, leaving ["cat", "bird"].
- shift() modifies the original array by removing the first item.

2. Using the unshift() Method to Add Elements to the Beginning of an Array

The unshift() method adds one or more elements to the beginning of an array. It returns the new length of the array.

Example:

```
let pets = ["cat", "bird"];
pets.unshift("fish", "ferret");  // Adds "fish" and "ferret" to the beginning
console.log(pets);  // ["fish", "ferret", "cat", "bird"]
```

- **Before**: ["cat", "bird"]
- **After**: ["fish", "ferret", "cat", "bird"]
- unshift() modifies the original array by adding new elements at the start.

3. Using the splice() Method to Add or Remove Elements Anywhere in an Array

The splice() method allows you to modify an array by inserting, deleting, or replacing elements. It takes three arguments:

- The **first argument** is the index at which to start modifying the array.
- The **second argument** is the number of elements to remove starting from the index.
- The **optional third argument** is one or more elements to add at the specified position.

Example 1: Adding and Removing Elements

```
let pets = ["dog", "cat", "fly", "bug", "ox"];
pets.splice(2, 2, "pig", "duck", "emu");  // Adds "pig", "duck", "emu" after "cat" and removes "fly" and "bug"
console.log(pets);  // ["dog", "cat", "pig", "duck", "emu", "ox"]
```

- **Before**: ["dog", "cat", "fly", "bug", "ox"]
- **After**: Starting at index 2, we remove two elements ("fly" and "bug") and insert "pig", "duck", and "emu". Result: ["dog", "cat", "pig", "duck", "emu", "ox"]

Example 2: Adding Elements Without Removing Any

```
let pets = ["dog", "cat", "fly", "bug", "ox"];
pets.splice(2, 0, "pig", "duck", "emu");  // Adds "pig", "duck", "emu" without removing any elements
console.log(pets);  // ["dog", "cat", "pig", "duck", "emu", "fly", "bug", "ox"]
```

- **Before**: ["dog", "cat", "fly", "bug", "ox"]
- **After**: Starting at index 2, we add "pig", "duck", and "emu", but do not remove any elements. Result: ["dog", "cat", "pig", "duck", "emu", "fly", "bug", "ox"]

Example 3: Removing Elements Without Adding Any

```
let pets = ["dog", "cat", "fly", "bug", "ox"];
pets.splice(2, 2);  // Removes two elements starting at index 2 ("fly" and "bug")
console.log(pets);  // ["dog", "cat", "ox"]
```

- **Before**: ["dog", "cat", "fly", "bug", "ox"]
- **After**: Starting at index 2, we remove two elements ("fly" and "bug"), leaving the result ["dog", "cat", "ox"].

4. Using the slice() Method to Copy Elements from an Array

The slice() method is used to create a shallow copy of a portion of an array without modifying the original array. It returns a new array with the selected elements.

The slice() method takes two arguments:

- The **first argument** is the index of the first element to include in the new array (inclusive).
- The **second argument** is the index of the element **after** the last element to include (exclusive).

Example: Copying Elements

```
let pets = ["dog", "cat", "fly", "bug", "ox"];
let noPets = pets.slice(2, 4);  // Copies "fly" and "bug" to the new array
console.log(noPets);  // ["fly", "bug"]
console.log(pets);  // ["dog", "cat", "fly", "bug", "ox"]
```

- **Before**: ["dog", "cat", "fly", "bug", "ox"]
- **After**: The slice() method creates a new array noPets containing elements from index 2 (inclusive) to index 4 (exclusive). The result: ["fly", "bug"]. The original array pets is unchanged.

Key Points to Remember About slice():

- The first index is **inclusive**, meaning it includes the element at that index.
- The second index is **exclusive**, meaning it does not include the element at that index.

- The slice() method does **not modify** the original array; it returns a new array.
- If you omit the second argument, slice() will copy all elements from the starting index to the end of the array.

Summary of Array Methods:

1. **shift()**: Removes the first element from the array.
2. **unshift()**: Adds one or more elements to the beginning of the array.
3. **splice()**: Adds or removes elements from anywhere in the array, with the option to replace elements.
    - splice(startIndex, numElementsToRemove, newElement1, newElement2, ...)
4. **slice()**: Creates a shallow copy of a portion of the array without modifying the original array.
    - slice(startIndex, endIndex)