**Array Methods**

**Array join()**

The join() method returns a new string by concatenating all of the elements in an array, separated by a specified separator.

**join() Syntax**

The syntax of the join() method is:

arr.join(separator)

Here, arr is an array.

**Example**

var message = ["JavaScript", "is", "fun."];

// join all elements of array using space

let joined Message = message. join(" ");

console.log(joined Message);

**// Output: JavaScript is fun.**

**join() Parameters**

The join() method takes in:

separator (optional) - A string to separate each pair of adjacent elements of the array. By default, it is comma ,.

**join() Return Value**

Returns a String with all the array elements joined by separator.

**Example: Using join() method**

var info = ["Terence", 28, "Kathmandu"];

var info\_str = info.join(" | ");

// join() does not change the original array

console.log(info); // [ 'Terence', 28, 'Kathmandu' ]

// join() returns the string by joining with separator

console.log(info\_str); // Terence | 28 | Kathmandu

// empty argument = no separator

var collection = [3, ".", 1, 4, 1, 5, 9, 2];

console.log(collection.join("")); // 3.141592

var random = [44, "abc", undefined];

console.log(random.join(" and ")); // 44 and abc and

**Output**

[ 'Terence', 28, 'Kathmandu' ]

Terence | 28 | Kathmandu

3.141592

44 and abc and

**Concat()**

The JavaScript concat() method is used to merge two or more arrays into a single array. It does not modify the original arrays but creates a new array containing the elements of the input arrays in the order they were passed to the method.

**Here's the syntax for the concat() method:**

var new Array = array1.concat(array2, array3, ..., array N);

Basic Example of concat()

Consider the following arrays:

const array1 = [1, 2, 3];

const array2 = ['a', 'b', 'c'];

To merge these arrays using the concat() method, you can do the following:

Var merged Array = array1.concat(array2);

console.log(merged Array); **// Output: [1, 2, 3, 'a', 'b', 'c']**

In this example, merged Array is a new array containing the elements of array1 and array2 in the order they were passed to the concat() method.

Merging Multiple Arrays

The concat() method allows you to merge multiple arrays at once. Consider the following example:

Var array1 = [1, 2, 3];

Var array2 = ['a', 'b', 'c'];

Var array3 = [true, false];

const merged Array = array1.concat(array2, array3);

console.log(merged Array); **// Output: [1, 2, 3, 'a', 'b', 'c', true, false]**

In this example, we merged three arrays into a single array using the concat() method.

Merging Nested Arrays

The concat() method can also merge arrays containing nested arrays. However, it will not flatten the nested arrays. Here's an example:

Var array1 = [1, 2, 3];

Var array2 = [['a', 'b'], 'c'];

Var merged Array = array1.concat(array2);

console.log(merged Array); **// Output: [1, 2, 3, ['a', 'b'], 'c']**

As you can see, the nested array ['a', 'b'] remains unchanged in the merged array.

**index. of()**

The index Of() method of Array instances returns the first index at which a given element can be found in the array, or -1 if it is not present.

Var beasts = ['ant', 'bison', 'camel', 'duck', 'bison'];

console.log(beasts. index Of('bison'));

**// Expected output: 1**

// Start from index 2

console.log(beasts. Index Of('bison', 2));

**// Expected output: 4**

console.log(beasts. index Of('giraffe'));

**// Expected output: -1**

**Syntax**

Index Of(search Element)

Index Of(search Element, from Index)