JavaScript Array Methods

1. Concat > Merges two or more arrays.
   Returns a new array containing the combined elements.
   eg: [1,2,3].concat([4,5,6]) creates a new array
   [1,2,3,4,5,6] ← without modifying.

2. every > Tests wheather all elements in the array pass a test implemented by the provided function.
   Returns a boolean (true if all elements pass, false otherwise).
   eg: [1,2,3].every(num => num = 0)
   returns true because all numbers are > 0

3. fill > Fills all elements of an array with a static value.
   Returns the modified array.
   eg: [1,2,3].fill(0) changes the array to [0,0,0], replacing all elements with 0.

4. Find > Returns the first element in the array that satisfies the provided testing function.
   Returns: The value of the first element that passes the test, or undefined if no element passes.
   eg: [1,2,3,4,5].find(num >= num > 3) returns 4, the first ~~element~~ number greater than 3.

5. FindIndex > Returns the index of the first element in the array that satisfies the provided testing function. Which returns the index of the first element that passes the test, or -1 if no element passes.

example: [1,2,3,4,5]. findIndex (num => num >3) returns 3, the index of 4 (the first number > 3).

6. Flat > Creates a new array with sub-array elements concatenated recursively up to the specified depth. Returns: A new array with the sub-arrays flattened.
example: [1, [2,3], [4,[5,[]]]]. flat(2) flattens the array to depth 2, resulting in [1,2,3,4,5,6].

7. includes > Determines wheather an array includes a certain value among its entries. Returns a boolean (true if the value is found, false otherwise)
example: [1,2,3,4,5]. includes (3) returns true because 3 is in the array.

8. indexOf > Returns the first index at which a given element can be found in the array. Which returns: the index of the element, or -1 if it is not present.
example: [1,2,3,4,5] indexOf (3) returns 2, ~~the index of 3~~ the index of 3 in the array.

9. join > Creates and returns a new string by concatenating all elements in a array. A example ['Hello', 'World'] join (' ') returns "Hello World", joining the elements with a space.

10. lastIndexOf > Returns the last index at which a given element can be found in the array.
example: [1,2,3,2,1]. lastIndexOf (2) returns 3,

the index of the last occurrence of 2.

11. pop > Returns the last element from an array.
example: For ['apple', 'banana', 'pizza'], pop() returns 'pizza' and modifies the array to ['apple','banana']

12. push > Adds one or more elements to the end of an array.
example: Fruits.push('grape') adds 'grape' to the end of the Fruits array and returns its new length.

13. reverse > Reverses the order of the elements in an array in place.
example: [1,2,3].reverse() modifies the array to [3,2,1]

14. unshift > Adds one or more elements to the beginning of an array. The new length of the array as return.
example: Fruits.unshift('mango') adds 'mango' to the beginning of the fruits array and returns its new length.

15. shift > Removes the first element from an array.
example: For ['mango','apple','banana'], shift() returns 'mango' and modifies the array to ['apple','banana']

16. slice > Returns a shallow copy of a portion of an array
example: [1,2,3,4,5].slice(1,4) returns [2,3,4], extracting elements from index 1 to 3 (4 is exclusive)

17. Some > Tests whether at least one element in the array passes the test implemented by the provided Function.
example: [1, 2, 3, -4, 5].some (num => num < 0) returns turn because at least one number (-4) is less than 0

19. Sort > Sorts the elements of an array in the place.
example: [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5].sort ((a, b) => a - b)
sorts the array in the ascending order.

20. splice > Changes the contents of an array by removing or replacing existing elements and/or adding new elements in place.
example: months.splice (1, 0, 'Feb') inserts 'Feb' at index 1 without removing any elements.

21. toString > Returns a string representing the array & its elements
example: [1, 2, 3].toString() returns the string "1, 2, 3"

22. Filter > Creates a new array with all elements that pass the test implemented by the provided function
example: [1, 2, 3, 4, 5].filter (num => num % 2 === 0) returns [2, 4], containing only the even numbers.

23. reduce > Executes a reducer function on each element of the array, resulting in a single output value.
example: [1, 2, 3, 4, 5].reduce ((acc, curr) => acc + curr, 0) sums all numbers in the array, returning 15.

24. map > Creates a new array with the results of calling a provided function on every element in the array. And returns a new array with each element being the result of the callback function.

example: [1,2,3].map(num => num * 2)
returns [2,4,6], doubling each number.

25. ForEach > Executes a provided function once for each array element.

example: [1,2,3].ForEach(num => console.log(num*2))
logs 2,4 and 6 to the console but doesn't return anything.