JavaScript Array Methods Cheat Sheet (Non-Basic)

Methods that return True / False

some(callback) - At least one element passes the test.
Return: Returns true/false | [yes] Stops early
Example: ['Apple', 'Banana'].some(fruit => fruit.length > 5) // true

every(callback) - All elements pass the test.
Return: Returns true/false | [yes] Stops early
Example: [2, 4, 6].every(n => n % 2 === 0) // true

includes(value, fromIndex?) - Checks if value exists in array.
Return: Returns true/false | [no] Checks all if needed
Example: [1, 2, 3].includes(2) // true

Array.isArray(value) - Checks if a value is an array.
Return: Returns true/false | [no] N/A
Example: Array.isArray([1,2,3]) // true

Methods that return First Matching Element

find(callback) - First element that matches.
Return: Returns element or undefined | [yes] Stops early
Example: [10, 20, 30].find(n => n > 15) // 20

findLast(callback) - Last element that matches.
Return: Returns element or undefined | [yes] Stops early
Example: [10, 20, 30].findLast(n => n > 15) // 30

findIndex(callback) - Index of first match.
Return: Returns index or -1 | [yes] Stops early
Example: [10, 20, 30].findIndex(n => n > 15) // 1

findLastIndex(callback) - Index of last match.
Return: Returns index or -1 | [yes] Stops early
Example: [10, 20, 30].findLastIndex(n => n > 15) // 2

at(index) - Element at given index (can be negative).
Return: Returns element | [no]
Example: [1, 2, 3].at(-1) // 3

Methods that return All Matching Elements

filter(callback) - All matching elements.
Return: Returns new array | [no]
Example: [1, 2, 3, 4].filter(n => n > 2) // [3, 4]

map(callback) - Transforms each element.
Return: Returns new array | [no]
Example: [1, 2, 3].map(n => n * 2) // [2, 4, 6]

flat(depth) - Flattens nested arrays.
Return: Returns new array | [no]
Example: [[1, 2], [3, 4]].flat() // [1, 2, 3, 4]

flatMap(callback) - Maps then flattens one level.
Return: Returns new array | [no]
Example: [1, 2].flatMap(n => [n, n*2]) // [1, 2, 2, 4]

Methods that return Single Calculated Value

reduce(callback, initialValue) - Reduces to a single value.
Return: Returns any type | [no]
Example: [1, 2, 3].reduce((a, b) => a + b, 0) // 6

reduceRight(callback, initialValue) - Reduces from right to left.
Return: Returns any type | [no]
Example: [1, 2, 3].reduceRight((a, b) => a + b, 0) // 6

Methods that return New Ordered Array

sort(callback) - Sorts array (mutates).
Return: Returns same array | [no]
Example: [3, 1, 2].sort() // [1, 2, 3]

toSorted(callback) - Sorted copy (no mutate).
Return: Returns new array | [no]
Example: [3, 1, 2].toSorted() // [1, 2, 3]

reverse() - Reverses array (mutates).
Return: Returns same array | [no]
Example: [1, 2, 3].reverse() // [3, 2, 1]

toReversed() - Reversed copy (no mutate).
Return: Returns new array | [no]
Example: [1, 2, 3].toReversed() // [3, 2, 1]

Methods that return Indexes

indexOf(value, fromIndex?) - Index of first match or -1.
Return: Returns number | [no]
Example: [1, 2, 3].indexOf(2) // 1

lastIndexOf(value, fromIndex?) - Index of last match or -1.
Return: Returns number | [no]
Example: [1, 2, 2, 3].lastIndexOf(2) // 2