Absolutely! In JavaScript, what you refer to as "list methods" are typically methods available on **Arrays** — since JavaScript uses the `Array` object to work with lists.

Here's a **complete list** of **Array methods** in JavaScript (as of ECMAScript 2024). These include mutating, non-mutating, accessor, iteration, and utility methods.

🔁 Mutator Methods (Change the original array)

1. `push()`
2. `pop()`
3. `shift()`
4. `unshift()`
5. `splice()`
6. `sort()`
7. `reverse()`
8. `copyWithin()`
9. `fill()`

🔍 Accessor Methods (Do NOT change the original array)

10. `concat()`
11. `includes()`
12. `indexOf()`

13. `lastIndexOf()`
14. `join()`
15. `slice()`
16. `toString()`
17. `toLocaleString()`
18. `at()` *(ES2022)*

🔁 Iteration Methods

19. `forEach()`
20. `map()`
21. `filter()`
22. `reduce()`
23. `reduceRight()`
24. `some()`
25. `every()`
26. `find()`
27. `findLast()` *(ES2023)*
28. `findIndex()`
29. `findLastIndex()` *(ES2023)*

30. `flat()`

31. `flatMap()`

32. `entries()`

33. `keys()`

34. `values()`

🧪 Utility / Static Methods on `Array` constructor

35. `Array.from()`

36. `Array.isArray()`

37. `Array.of()`

🆕 New Methods in ECMAScript 2023+

38. `with()` *(Immutable element replacement)*

39. `toReversed()`

40. `toSorted()`

41. `toSpliced()`

42. `toSorted()` *(yes, part of ES2023 immutables)*

43. `group()` *(stage 3)*

44. `groupToMap()` *(stage 3)*

Would you like simple examples for each method as well? I can help create a full JavaScript notebook with examples per method too.