

JS LIST METHODS

Returns a Boolean value, based on whether the list contains the element specified as an argument.

```
["h", "e", "|", "|", "o"].includes("h")
<< true
["h", "e", "|", "|", "o"].includes("f")
<< false
```

Returns the last element in a list

```
["my", "array"].pop()
<< "array"
```

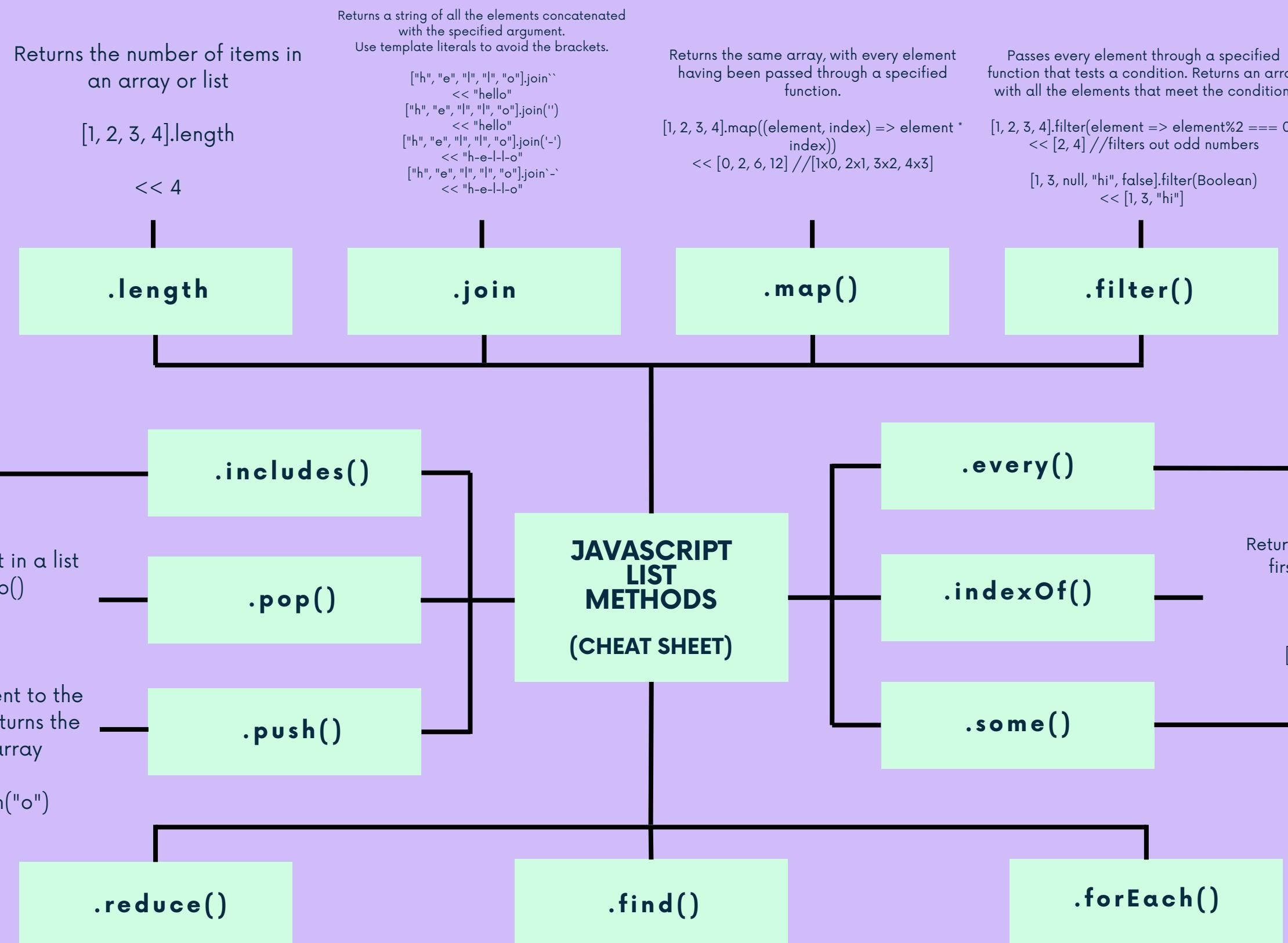
Adds a specified element to the end of an array and returns the length of the new array

```
["h", "e", "|", "|"].push("o")
<< 5
```

Passes each element of the list through a function. With every pass, the element contributes to building up a value. This value is defined at the start, and is returned once all elements have been passed through.

```
[1, 2, 3, 4, 5].reduce((sum, element) => sum + element, 0)
<< 15 // 1 + 2 + 3 + 4 + 5
```

Note: unlike the other methods that use functions, the initial value must be defined as the parameter first (before element), and defined in a comma after the function.



IMPORTANT NOTE:

All methods that require functions as arguments have three inbuilt parameters: the element of the list, the index of the current element, and the array itself.

For example,
`.map((element, index, array) => *magic*)`

The parameters must be defined in that order, and you do not need all of them each time. However, you must have the element.