ScopeTreasure Courses

JAVASCRIPT COURSES · LESSON 3.8 · CODE SCOPE SERIES

Arrays

One name, many boxes: indexes from zero, length, changing items, push/pop/shift/unshift, and loops that turn arrays into page content.

Introduction

A menu has six items. With plain variables you need six boxes: `item1`, `item2`, `item3`... and the pain only grows. An **array** solves this with one simple idea: **one name, many ordered boxes**. One variable holds the whole list, and each item has a number — its **index**.

In this lesson you will meet the most common shock for beginners — **index starts at zero** — learn `length`, change items, add and remove with `push` and `pop`, glance at `shift` and `unshift`, and keep the promise from Lesson 3.6: a loop walking an array to build real page content.

(Roman Urdu: Array matlab aik naam ke neeche bohat se khaanay — aur ginti hamesha zero se shuru hoti hai.)

Key points

- **One array = one name, many boxes:** `let menu = ["Chai", "Coffee", "Lassi"];` — the items sit inside square brackets, separated by commas.
- **INDEX STARTS AT ZERO:** the first item is `menu[0]`, the second is `menu[1]`, the third is `menu[2]`. There is no item at the count you expect — asking `menu[3]` on a 3-item array gives `undefined`. That is not an error; the box simply does not exist.
  (Roman Urdu: Pehli cheez ka number 0 hai, 1 nahi — yehi sab se bari beginner ghalti hai.)
- **`length` counts the items:** `menu.length` is `3`. And the **last item is always at `length - 1`** — here, `menu[menu.length - 1]` is `"Lassi"`.
- **Change an item by its index:** `menu[1] = "Cold Coffee";` replaces the value in that box. The array itself was declared once; its boxes stay changeable.
- **`push` adds at the end; `pop` removes from the end:** `menu.push("Green Tea")` grows the array. `menu.pop()` removes the last item **and returns it**, so you can store what was removed.
  (Roman Urdu: push aakhir mein jorta hai, pop aakhir se nikaal kar wapas deta hai.)
- **`shift` and `unshift` work at the start:** `shift()` removes and returns the **first** item; `unshift("Chai")` adds at the front. Same idea as push/pop, other end of the line.
- **Loops love arrays:** a `for` loop from `0` to `length - 1` visits every item in order — the pattern behind menus, product lists, and galleries.

Code examples

Example 1 — Indexes, length, and the zero shock

```
let menu = ["Chai", "Coffee", "Lassi"];

console.log(menu[0]);
console.log(menu[1]);
console.log(menu[2]);
console.log(menu[3]);
console.log(menu.length);
console.log(menu[menu.length - 1]);
```

What it does: Reads the menu box by box. The first three reads give the three drinks. The fourth read asks for index `3` — a box that does not exist in a 3-item array. The last two lines show the safe way to count items and reach the final one.

What you will see: six lines in the Console: `Chai`, `Coffee`, `Lassi`, `undefined`, `3`, and `Lassi`. That `undefined` is the classic beginner shock — and now it will never surprise you again.

Example 2 — push, pop, and changing an item

```
let menu = ["Chai", "Coffee", "Lassi"];

menu[1] = "Cold Coffee";
console.log(menu);

menu.push("Green Tea");
console.log(menu);
console.log(menu.length);

let removed = menu.pop();
console.log(removed);
console.log(menu);
```

What it does: First, index `1` is replaced — Coffee becomes Cold Coffee. Then `push` adds Green Tea at the end. Finally `pop` removes the last item and hands it back, so it can be stored in `removed` and printed.

What you will see: five outputs in the Console: `["Chai", "Cold Coffee", "Lassi"]`, then `["Chai", "Cold Coffee", "Lassi", "Green Tea"]`, then `4`, then `Green Tea`, then the array back to `["Chai", "Cold Coffee", "Lassi"]`.

Example 3 — The loop promise kept: array into menu items

```
<!DOCTYPE html>
<html>
<head>
  <title>ChaiPoint Menu</title>
</head>
<body>
  <h1>Our Menu</h1>
  <ul id="menuList"></ul>

  <script>
    let menu = ["Chai", "Coffee", "Lassi"];
    let list = document.getElementById("menuList");
    let html = "";

    for (let i = 0; i < menu.length; i++) {
      html = html + "<li>" + menu[i] + "</li>";
    }

    list.innerHTML = html;
  </script>
</body>
</html>
```

What it does: This is Lesson 3.6's promise kept — a loop walks the array and builds one list item per drink, then puts the finished HTML into the page's empty list. Add a drink to the array and the page grows by itself; no HTML editing needed.

What you will see: the page shows the heading "Our Menu" and a bullet list with three items: Chai, Coffee, Lassi. Push a fourth drink into the array, reload, and four bullets appear.

Quick recap

- An array is **one name, many ordered boxes** — and the first box is index **0**.
- `length` counts items; the last item lives at `length - 1`; asking beyond it gives `undefined`.
- `push`/`pop` work at the end, `shift`/`unshift` at the start — and loops turn arrays into page content.

Common mistakes

1. **Counting indexes from 1** — asking for `menu[1]` when you mean the first item gives you the second one. Fix: always count `0, 1, 2` — first item is index `0`.
2. **Asking for an index that does not exist** — `menu[3]` on a 3-item array is `undefined`, and using it in maths or text creates strange bugs. Fix: the last safe index is `menu.length - 1`.

3. **Expecting pop to just delete** — `pop()` also **returns** the removed item, and `push` returns the new length, not the array. Fix: store the return value when you need it, and print the array itself to see its contents.

Practice

1. Make an array of five fruits. Print the first item, the last item (using `length - 1`), and the total count. Then change the third item to a new fruit and print the full array.
2. Start with an empty array `let cart = [];`. Use `push` to add three items, print the array, then use `pop` twice, printing the removed item each time. Finish with a `for` loop that prints every remaining item with its index.

Practice quiz

1. fruits = ['a','b','c']; code reads fruits[3] and gets undefined — what is the fix?

- Use fruits[2], the last real index of a 3-item array
- Use fruits[3] with quotes
- Add a fourth fruit
- Change fruits to Fruits

Answer: Use fruits[2], the last real index of a 3-item array

2. What does this print?

```
let arr = ["a", "b", "c"];
console.log(arr.length);
```

- 3
- 2
- c
- Error

Answer: 3

Watch the video lesson: https://youtu.be/ky2-nxfl-Bw

Read online: https://scopetreasure.com/notes/js-arrays · More free tools and notes at https://scopetreasure.com