

JavaScript Nested for Loops Notes

Overview

- **Nested for Loops:** Loops inside loops to process data combinations.

- Example: Combine first and last names into full names:

```
var firstNames = ["BlueRay ", "Upchuck ", "Lojack ", "Gizmo ", "Do-Rag "];
var lastNames = ["Zzz", "Burp", "Dogbone", "Droop"];
var fullNames = [];
for (var i = 0; i < firstNames.length; i++) {
    for (var j = 0; j < lastNames.length; j++) {
        fullNames.push(firstNames[i] + lastNames[j]);
    }
}
```

- Result: `fullNames` has 20 combinations (e.g., "BlueRay Zzz", "BlueRay Burp", etc.).

How It Works

- Outer loop (`i`) iterates over `firstNames`.
- Inner loop (`j`) iterates over `lastNames` for each `i`.
- Inner loop completes all iterations for each outer loop iteration (like minute and second hands of a clock).

Key Points

- Use different counter variables (`i`, `j`) for clarity.
- Indent inner loops 2 spaces for readability.
- Use `array.length` for dynamic loop limits.
- Multiple nesting levels are possible.

Style

- Use `===` for comparisons.
- Always use curly brackets `{}` for blocks, no semicolon after.