
How the Nested Loop Works:

- The **outer loop** (i) iterates through each first name.
- The **inner loop** (j) iterates through each last name for every first name.
- For each combination, we concatenate the names and store them in the fullNames array.

Corrected and Improved Code:

```
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]); // Combine names and store in the array
    }
}

// Display the generated rapper names
console.log(fullNames);
```

Output (20 Rap Names)

```
[
  "BlueRay Zzz", "BlueRay Burp", "BlueRay Dogbone", "BlueRay Droop",
  "Upchuck Zzz", "Upchuck Burp", "Upchuck Dogbone", "Upchuck Droop",
  "Lojack Zzz", "Lojack Burp", "Lojack Dogbone", "Lojack Droop",
  "Gizmo Zzz", "Gizmo Burp", "Gizmo Dogbone", "Gizmo Droop",
  "Do-Rag Zzz", "Do-Rag Burp", "Do-Rag Dogbone", "Do-Rag Droop"
]
```

Key Takeaways:

1. **Nested Loops**: The inner loop runs **completely** for every iteration of the outer loop.
2. **String Concatenation**: firstNames[i] + lastNames[j] merges the two name parts.
3. **Dynamic Expansion**: If you add more names to either list, the code still works without modification.

4. **Scalability**: You can increase the number of rapper names easily by adding more options.
