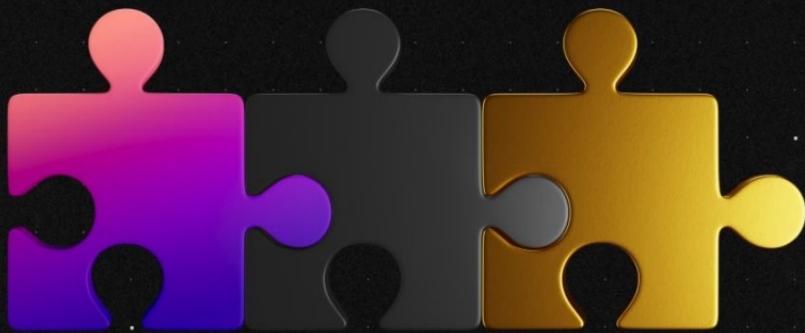


JS

For loops, array reduce & method chaining in **JavaScript**

→ like and share



For Loops



```
const files = [ 'foo.txt ',  
  '.bar', ' ', 'baz.foo' ];  
let filePaths = [];  
  
for (let file of files) {  
  const fileName = file.trim();  
  if(fileName) {  
    const filePath =  
      `~/cool_app/${fileName}`;  
    filePaths.push(filePath);  
  }  
}
```

Key features

1. Less common nowadays, due to functional programming being more popular.
2. Control over the iteration, such as skipping over elements or early returns.
3. Resulting array needs to be declared beforehand, outside the loop.
4. Uses `Array.prototype.push()` or the spread (...) operator to add elements.
5. $O(N)$ complexity, each element will be iterated over only once.



Array reduce



```
const files = [ 'foo.txt' ,  
'.bar' , ' ' , 'baz.foo' ];  
const filePaths =  
  files.reduce((acc, file) => {  
    const fileName = file.trim();  
    if(fileName) {  
      const filePath =  
        `~/cool_app/${fileName}`;  
      acc.push(filePath);  
    }  
    return acc;  
}, []);
```

Key features

1. Uses `Array.prototype.reduce()` with an empty array as the initial value.
2. More common nowadays, due to functional programming being more popular.
3. Less control over the iteration, cannot skip elements or return early.
4. Can be chained with other methods, if necessary.
5. Uses `Array.prototype.push()` or the spread (...) operator to add elements.
6. $O(N)$ complexity, each element will be iterated over only once.



Method chaining

```
const files = [ 'foo.txt', '.bar', ' ', 'baz.foo' ];
const filePaths = files
  .map(file => file.trim())
  .filter(Boolean)
  .map(fileName => `~/cool_app/${fileName}`);
```

Key features

1. Uses `Array.prototype.map()` and `Array.prototype.filter()`.
2. More common nowadays, due to functional programming being more popular.
3. Less control over the iteration, cannot skip elements or return early.
4. Declarative, easier to read and refactor, chain can grow as necessary.
5. Does not use `Array.prototype.push()` or the spread (...) operator.
6. $O(cN)$ complexity, c iterations per element, (c : length of the chain).



Do you find it helpful?

let me know down in the
comments !



Slobodan Gajić

Content Creator



[FOLLOW FOR MORE](#)