(7.6.1) JS Basics - Arrays - .filter() Method

JavaScript is a versatile language with a rich set of array manipulation methods, and one of the most powerful and commonly used methods is .filter(). This method allows you to create a new array containing elements from an existing array that meet specific criteria. It's a fantastic tool for extracting, sorting, or transforming data in a clean and concise way.

The Basics

The .filter() method is available on arrays and takes a callback function as its argument. This callback function is called for each element in the array, and if it returns true, the element is included in the new filtered array. If it returns false, the element is excluded.

Here's a simple example that demonstrates how to use .filter():

```
const numbers = [1, 2, 3, 4, 5, 6];

// Create a new array with even numbers
const evenNumbers = numbers.filter((num) => num % 2 === 0);

console.log(evenNumbers); // Output: [2, 4, 6]
```

In this example, the callback function (num) => num % 2 === 0 checks if each element is even. If the condition is met, the element is included in the evenNumbers array.

Using .filter() with Objects

You can also use .filter() with arrays of objects. For instance, if you have an array of objects representing people and you want to filter out only those who are adults, you can do it like this:

```
const people = [
{ name: "Alice", age: 25 },
{ name: "Bob", age: 17 },
{ name: "Charlie", age: 30 },
{ name: "David", age: 15 },
];
```

```
// Filter out adults (age >= 18)
const adults = people.filter((person) => person.age >= 18);

console.log(adults);
// Output:
// [
// { name: "Alice", age: 25 },
// { name: "Charlie", age: 30 }
// ]
```

Wrapping Up

The .filter() method is a valuable tool in JavaScript for working with arrays. It provides a clean and efficient way to extract elements that meet specific criteria, making your code more readable and maintainable. By combining .filter() with other array methods, you can build powerful data transformation pipelines for a wide range of tasks. So go ahead and master the .filter() method to become a more proficient JavaScript developer.