Bing

The code you’ve provided has a couple of issues. Let’s break them down:

1. **Order of Function Calls**:
    - The `filter()` method is called before the `divisibleByTen` function is defined.
    - When the `filter()` method runs, it tries to use `divisibleByTen`, but it’s not yet defined, resulting in an error.
    - To fix this, move the definition of `divisibleByTen` above the line where you call `filter()`.
2. **Incorrect Usage of `filter()`**:
    - The purpose of the `filter()` method is to create a new array containing elements that meet a specified condition.
    - Your `divisibleByTen` function does not return a boolean (which is expected by `filter()`).
    - Instead, it returns the first element that satisfies the condition (i.e., the first number divisible by 10).
    - To correctly use `filter()`, the callback function should return `true` or `false` based on whether the element meets the condition.

Here’s an improved version of your code:

```javascript
const givenArray = [30, 50, 79, 79, 90, 101, 30];

const divisibleByTen = array => {
    return array.filter(arr => arr % 10 === 0);
};

const expectedArray = divisibleByTen(givenArray);
console.log(expectedArray);
```

In this corrected version:

- We define `divisibleByTen` before using it in the `filter()` method.
- The `divisibleByTen` function now correctly returns an array of elements that are divisible by 10.

Now your code should work as expected! 🚀