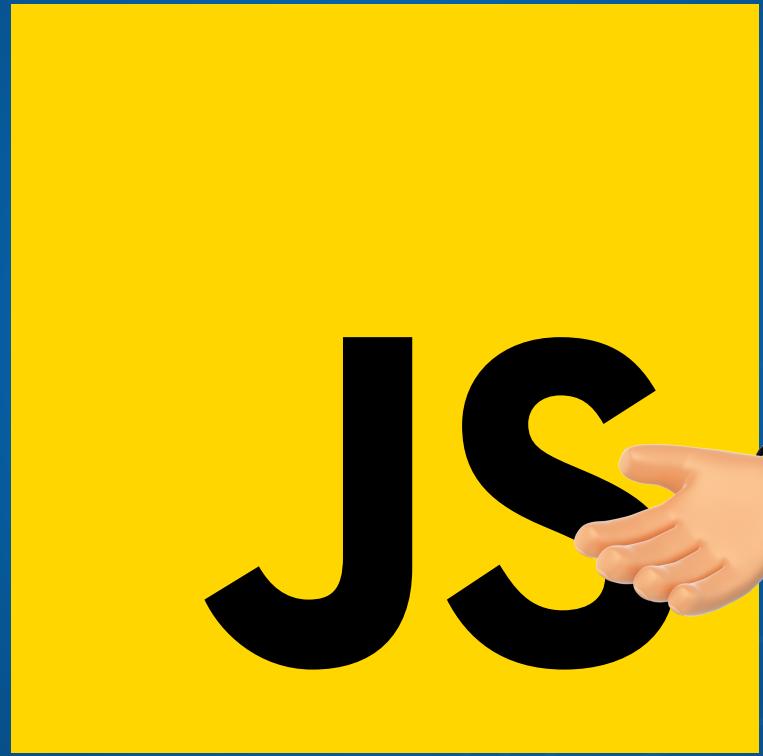


JavaScript callbacks



Kacper Hernacki
[@kacper.hernacki](https://twitter.com/kacper.hernacki)

Intro

By definition, a callback is a function that you pass into another function as an argument for executing later.



Kacper Hernacki
@kacper.hernacki

The following defines a filter() function that accepts an array of numbers and returns a new array of odd numbers:

```
function filter(numbers) {  
  let results = [];  
  for (const number of numbers) {  
    if (number % 2 != 0) {  
      results.push(number);  
    }  
  }  
  return results;  
}  
  
let numbers = [1, 2, 4, 7, 3, 5, 6];  
console.log(filter(numbers));
```



Kacper Hernacki
@kacper.hernacki

How it works:

- **First, define the filter() function that accepts an array of numbers and returns a new array of the odd numbers.**



Kacper Hernacki
@kacper.hernacki

- **Second, define the numbers array that has both odd and even numbers.**
- **Third, call the filter() function to get the odd numbers out of the numbers array and output the result.**



Kacper Hernacki
@kacper.hernacki

If you want to return an array that contains even numbers, you need to modify the filter() function.

```
●●●  
function isOdd(number) {  
    return number % 2 != 0;  
}  
  
function filter(numbers, fn) {  
    let results = [];  
    for (const number of numbers) {  
        if (fn(number)) {  
            results.push(number);  
        }  
    }  
    return results;  
}  
let numbers = [1, 2, 4, 7, 3, 5, 6];  
console.log(filter(numbers, isOdd));
```



Kacper Hernacki
@kacper.hernacki

**the filter() function now is
more generic and reusable,
it is made by :**

- First, extracting the logic
in the if block and wrap it
in a separate function.**
- Second, passing the
function to the filter()
function as an argument.**



Kacper Hernacki
@kacper.hernacki

Basically `isOdd()` function is a callback.

Because the `filter()` function accepts a function as an argument, it's called a high-order function.



Kacper Hernacki
@kacper.hernacki

Moreover, a callback can be an anonymous function, which is a function without a name like this:



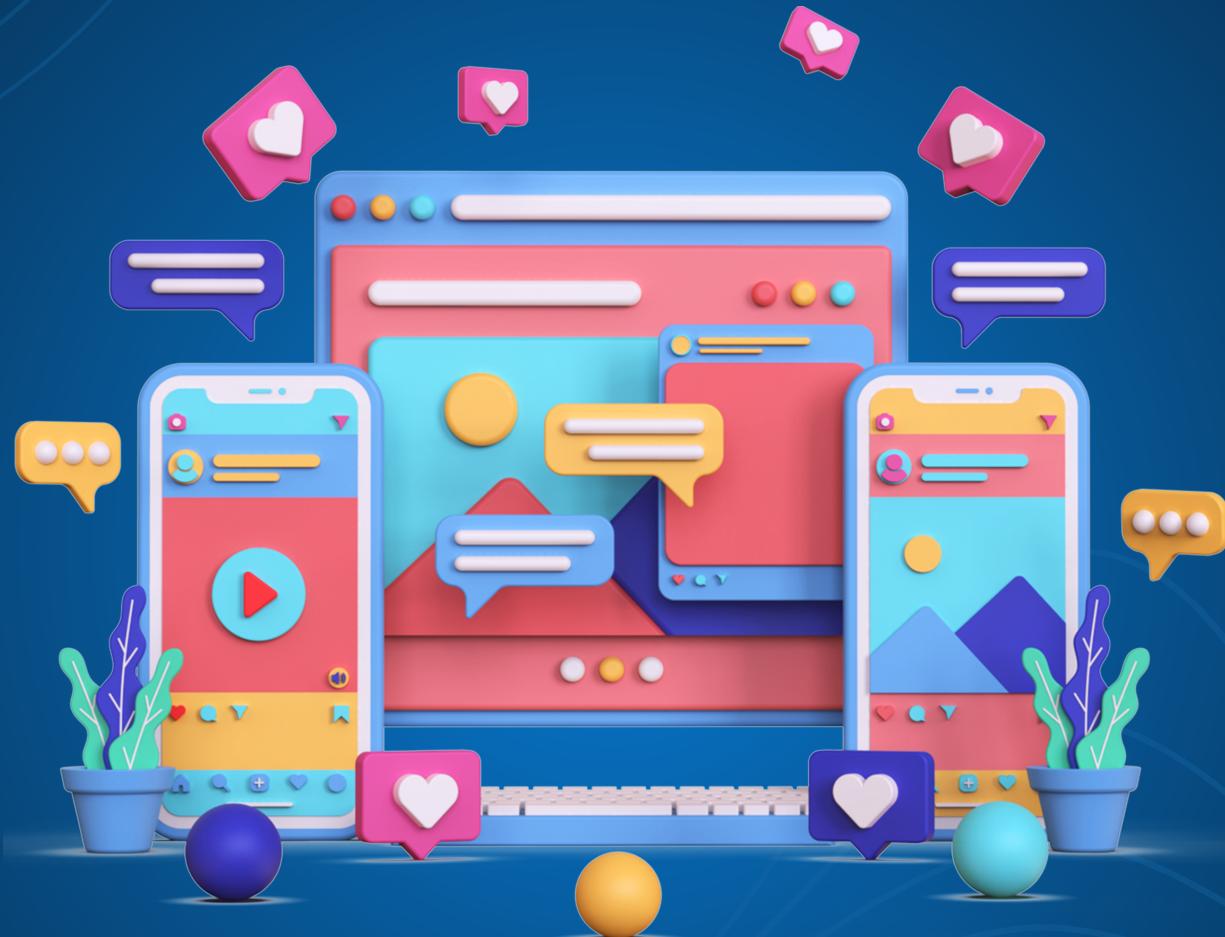
```
function filter(numbers, callback) {  
  let results = [];  
  for (const number of numbers) {  
    if (callback(number)) {  
      results.push(number);  
    }  
  }  
  return results;  
}  
  
let numbers = [1, 2, 4, 7, 3, 5, 6];  
  
let oddNumbers = filter(numbers, function (number) {  
  return number % 2 != 0;  
});  
  
console.log(oddNumbers);
```



Kacper Hernacki
@kacper.hernacki

Did you find it helpful?

leave a comment or share



Kacper Hernacki
@kacper.hernacki