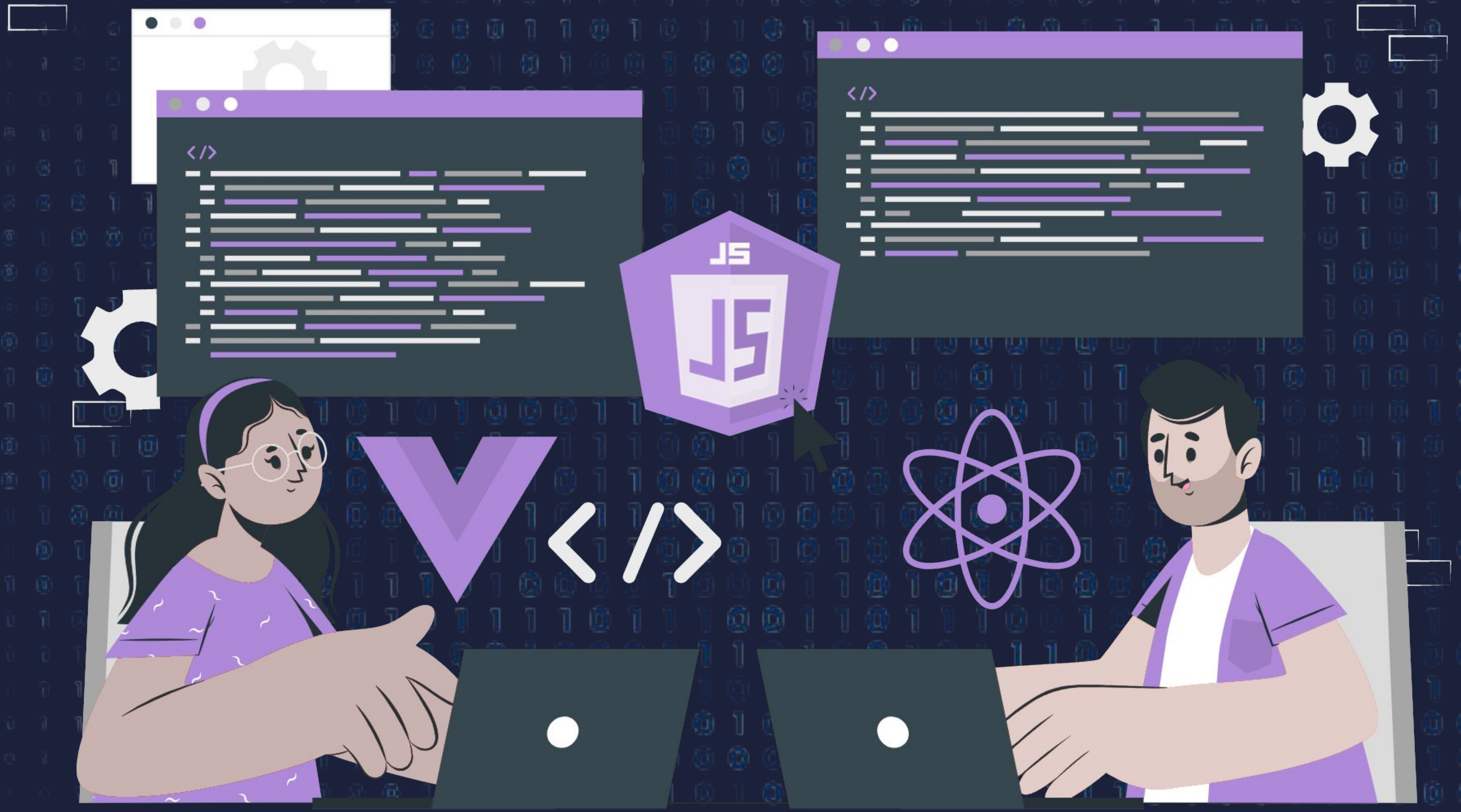
PW SKILLS

For each, map, filter



Lecture CheckList

1. Introduction to forEach.
2. Syntax.
3. Implementation.
4. Introduction to map method.
5. Syntax.
6. Implementation.
7. Introduction to filter method.
8. Syntax.
9. Implementation.

Introduction to forEach.

Earlier when ever there was a necessity of iterating over arrays, developers majorily used for loops. But, using for loop can be slow and error-prone when we deal with large complex arrays. So, forEach was standardized with the release of ECMAScript 5 (ES5) in 2009.

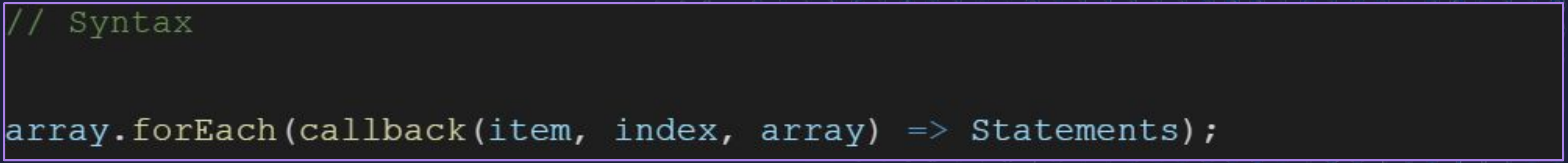
The introduction of the forEach() method in ES5 provided a more convenient and efficient way to iterate over the elements of an array. It allowed developers to perform a specific function on each element of an array without having to manually write a for loop.

Syntax

The following are the parameters of forEach() in Javascript:

- callbackFunction: This argument contains the method that will be called for each array element. It is a mandatory parameter.
- item: the array item on iteration.
- index: This parameter is optional and contains the index number of each element.
- array: This parameter is optional and contains the entire array on which the Array is being used.

The return value of this method is always undefined.



```
// Syntax

array.forEach(callback(item, index, array) => Statements);
```

Use Case

Let’s imagine you are assigned with task to display all the items added to cart in an ecommerce application. You can do it in multiple ways but let’s assume we are asked to do it using forEach.

Use Case

Now let’s try to use all the parameters. Let’s print the item name, the position in which the item was added to cart and the total cart items.

Introduction to map method.

The map method in javascript was introduced to iterate and perform operations on array items in a more concise and readable way.

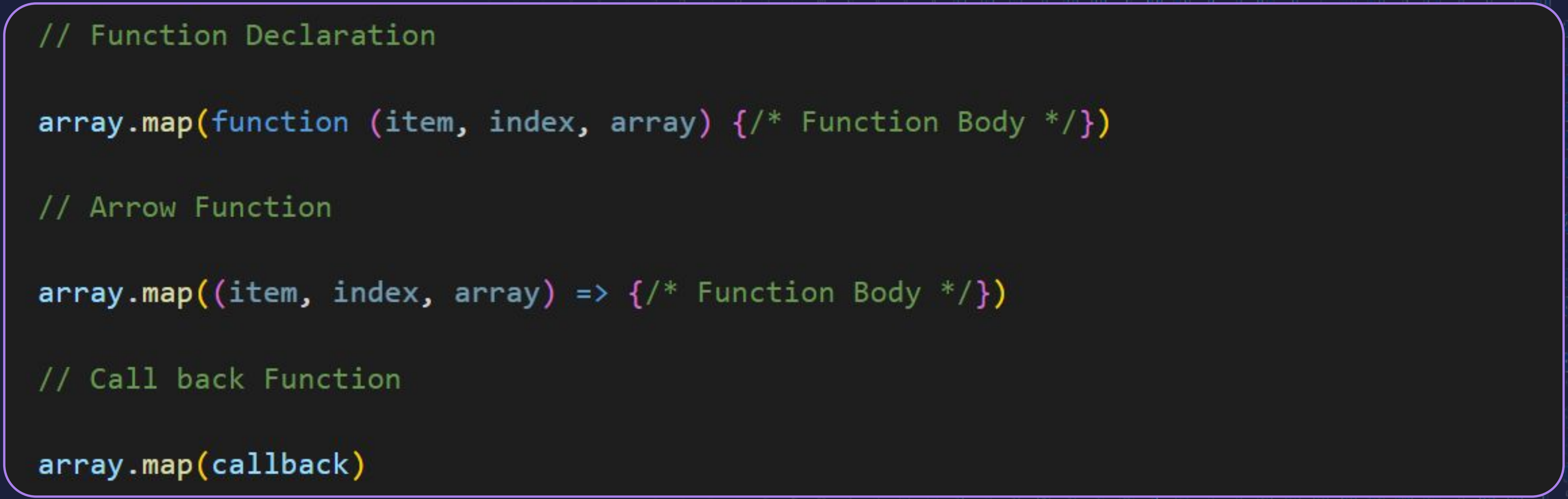
The map method of the array takes a function as a parameter and returns a new array that contains the result of the function performed on each array item.

The map() method is used over other iterative methods such as for loops or forEach() based on the applications. map() allows us to easily operate on the array items. It returns a new array containing the result of these operations. As it creates a new array, the array on which the map method is applied would be unaltered and less prone to error.

Syntax

The map method can be applied to an array in the following ways:

1. Function declaration.
2. Arrow Function.
3. Callback Function.



```
// Function Declaration

array.map(function (item, index, array) {/* Function Body */})

// Arrow Function

array.map((item, index, array) => {/* Function Body */})

// Call back Function

array.map(callback)
```

Use Case

Let’s now look at a use case of the map method. Imagine you are working at an e-commerce site. Imagine the website is storing the price as strings and you are assigned a task to convert all the prices in the cart to numbers so the total can be calculated.

Introduction to filter method.

As the name suggests the filter method is used to filter the data present in an array. On daily basis in every application, we use filtering. We filter the products in a shopping site based on criteria, we filter the posts on any social media based on the date posted or based on the keywords. So, filtering the data is one of the important part.

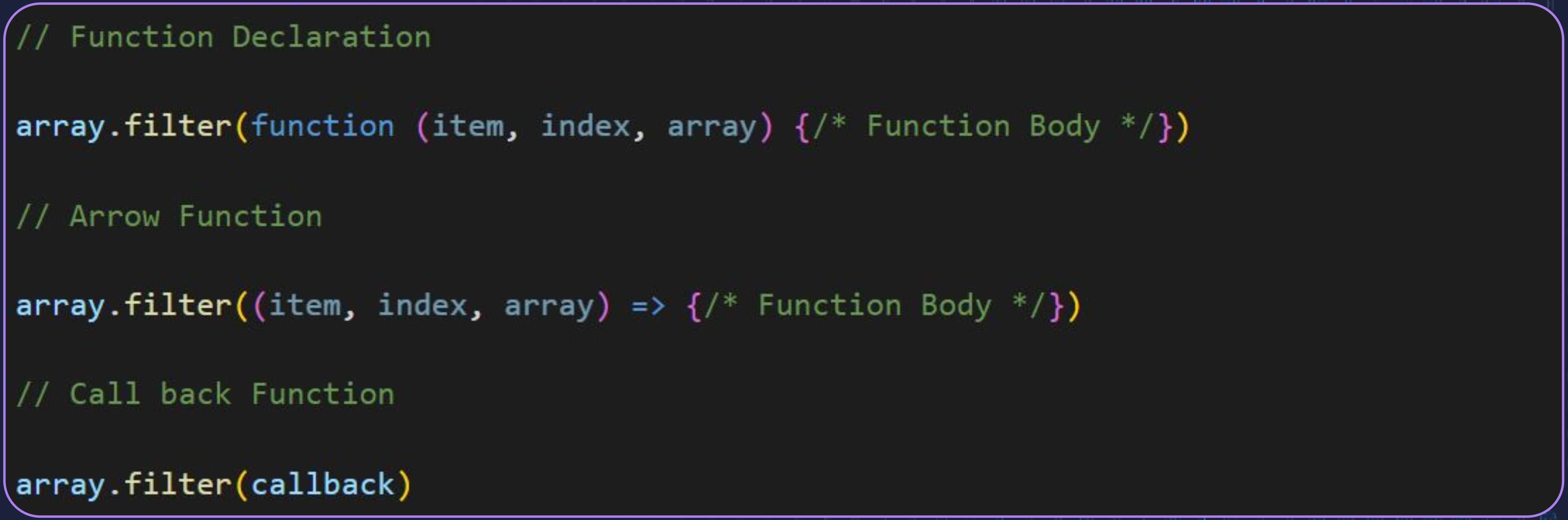
The filter method is used to create a new array whose items are the result of the filtering criteria of the original array. Only those values which satisfy the given criteria are added to a new array, and that array is returned. The original array does not get changed.

Whenever the filter method is applied to an array, we pass a filter function to it. The filter function iterates over all the elements of the given array and passes each element to the callback function. If the callback function returns true, then the element is added to the result array.

Syntax

The filter method can be applied to an array in the following ways:

1. Function declaration.
2. Arrow Function.
3. Callback Function.



```
// Function Declaration

array.filter(function (item, index, array) {/* Function Body */})

// Arrow Function

array.filter((item, index, array) => {/* Function Body */})

// Call back Function

array.filter(callback)
```

Use Case

Let’s imagine you are working for an e-commerce organization and you are assigned to return the long usernames from the data of the username. Long usernames in this case are usernames whose length is greater than 5 characters.

PW SKILLS

THANK YOU