

Lesson 06 Demo 02

Creating a Function to Filter the Product List

Objective: To create a function to filter an array of product objects based on specific criteria

Tools Required: Visual Studio Code and Node.js

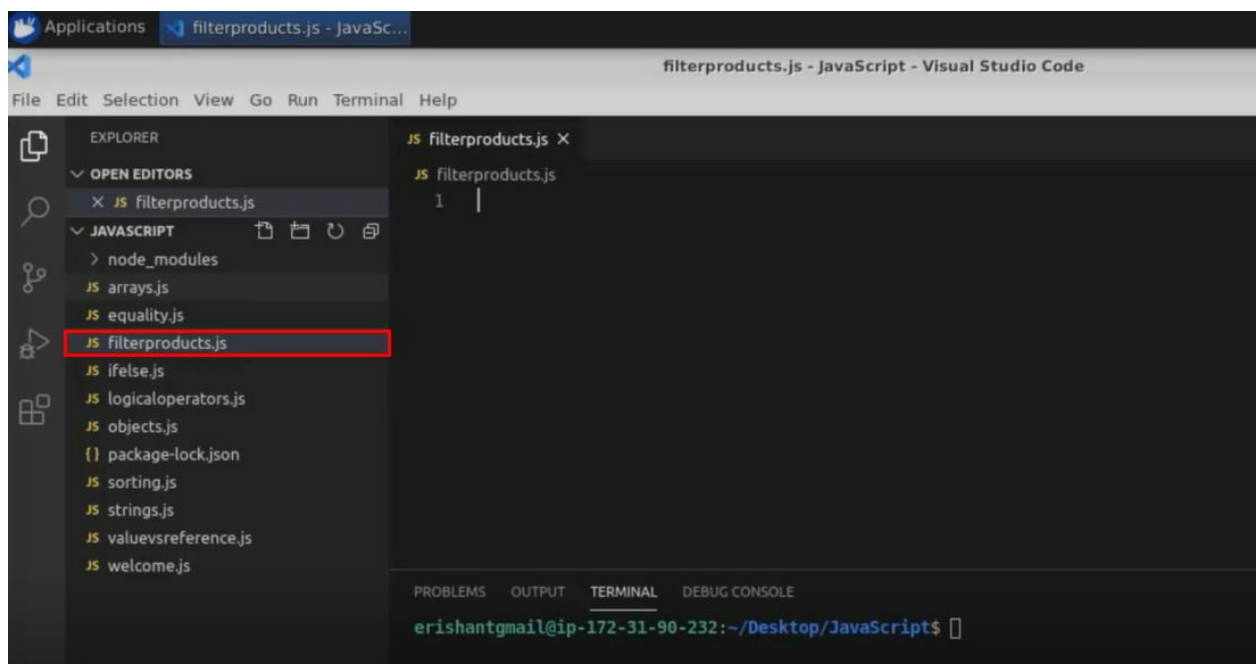
Prerequisites: None

Steps to be followed:

1. Define the filter function

Step 1: Define the filter function

- 1.1 Open Visual Studio Code and create a new JavaScript file named **filterproducts.js**



1.2 Copy the code from the previous snippet, **sorting.js**, and paste it into the **filterproducts.js** file

```

21  }
22  }
23
24  function printArray(array){
25    for(let item of array){
26      console.log(item);
27    }
28  }
29
30  function sortOnPrice(product1, product2){
31    //return product1.price - product2.price;
32    return product2.price - product1.price;
33  }
34
35
36  products.sort(sortOnPrice);

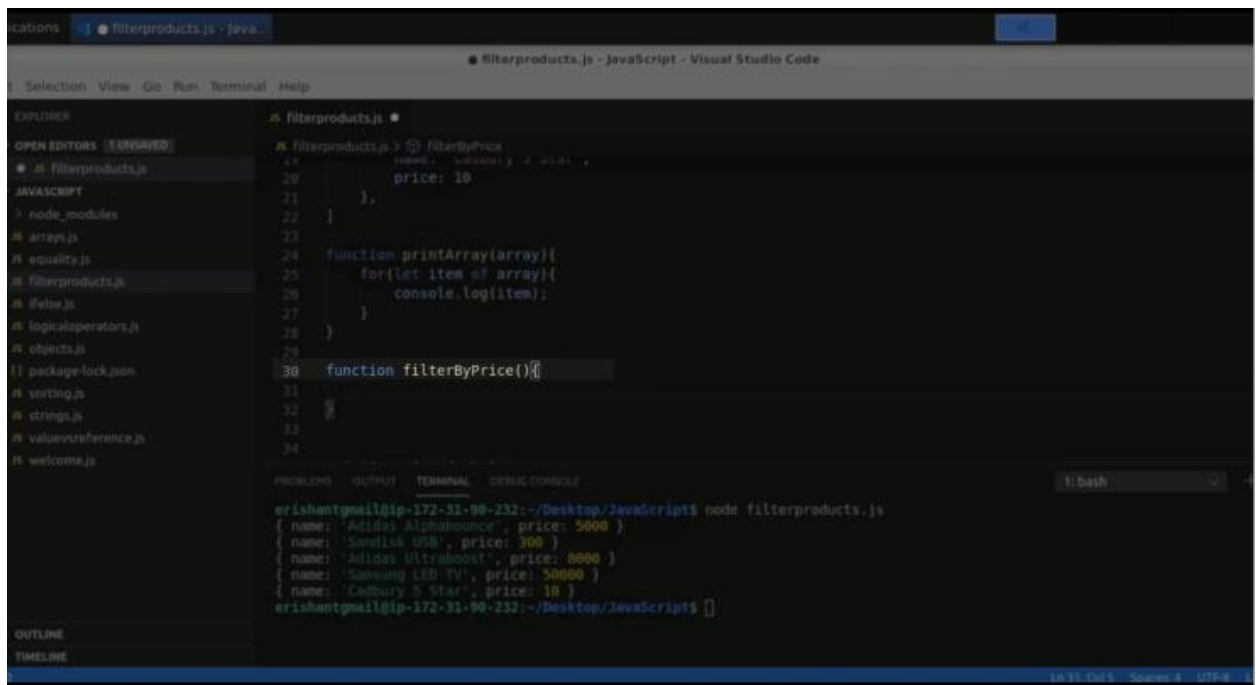
```

```

24  function printArray(array){
25    for(let item of array){
26      console.log(item);
27    }
28  }
29
30  function sortOnPrice(product1, product2){
31    //return product1.price - product2.price;
32    return product2.price - product1.price;
33  }
34
35
36  products.sort(sortOnPrice);
37
38  printArray(products);

```

1.3 Define a function called **filterByPrice** inside the **filterproducts.js** file



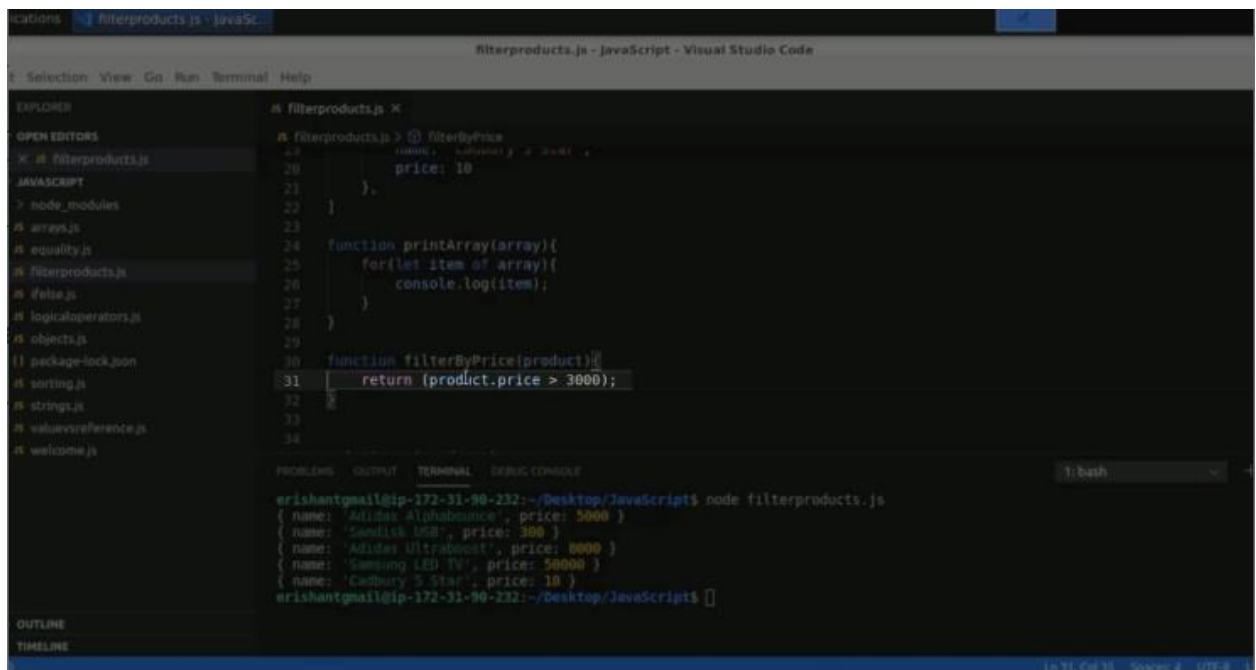
The screenshot shows the Visual Studio Code editor with the file `filterproducts.js` open. The file contains the following code:

```
1  // Filter products by price
2  const products = [
3    { name: 'Adidas Alphabounce', price: 5000 },
4    { name: 'Sandisk USB', price: 300 },
5    { name: 'Adidas Ultraboost', price: 8000 },
6    { name: 'Samsung LED TV', price: 50000 },
7    { name: 'Cadbury 3 Star', price: 10 }
8  ];
9
10 function printArray(array){
11   for(let item of array){
12     console.log(item);
13   }
14 }
15
16 function filterByPrice(){}
17
```

The terminal output shows the result of running `node filterproducts.js`:

```
erishant@gmail@ip-172-31-90-232:~/Desktop/Javascript$ node filterproducts.js
{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Cadbury 3 Star', price: 10 }
erishant@gmail@ip-172-31-90-232:~/Desktop/Javascript$
```

1.4 Within the **filterByPrice** function, define a return function that takes one product as input and checks if the product's price is greater than a certain threshold



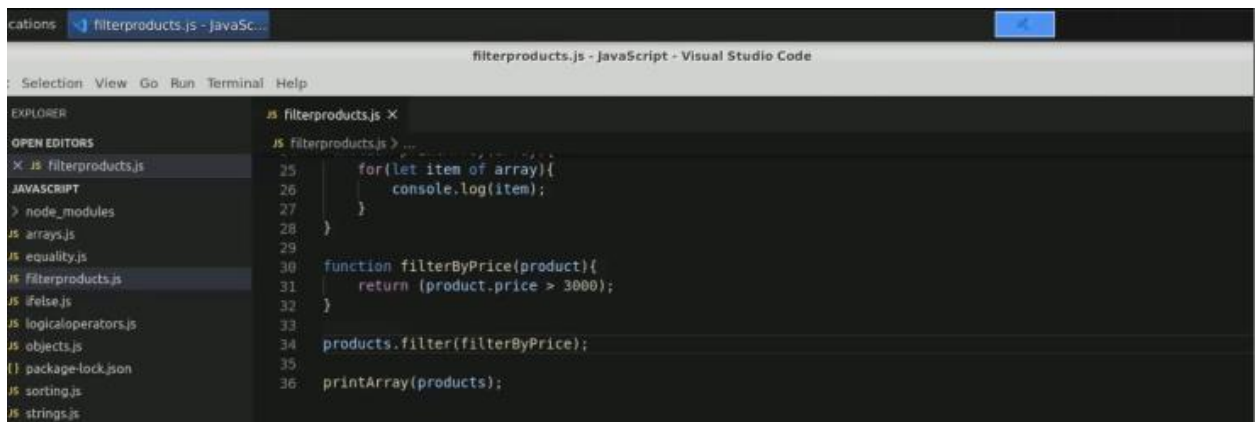
The screenshot shows the Visual Studio Code editor with the file `filterproducts.js` open. The file contains the following code:

```
1  // Filter products by price
2  const products = [
3    { name: 'Adidas Alphabounce', price: 5000 },
4    { name: 'Sandisk USB', price: 300 },
5    { name: 'Adidas Ultraboost', price: 8000 },
6    { name: 'Samsung LED TV', price: 50000 },
7    { name: 'Cadbury 3 Star', price: 10 }
8  ];
9
10 function printArray(array){
11   for(let item of array){
12     console.log(item);
13   }
14 }
15
16 function filterByPrice(product){
17   return (product.price > 3000);
18 }
19
```

The terminal output shows the result of running `node filterproducts.js`:

```
erishant@gmail@ip-172-31-90-232:~/Desktop/Javascript$ node filterproducts.js
{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Cadbury 3 Star', price: 10 }
erishant@gmail@ip-172-31-90-232:~/Desktop/Javascript$
```

1.5 Use the **filter** function and pass the **filterByPrice** function as an input



```

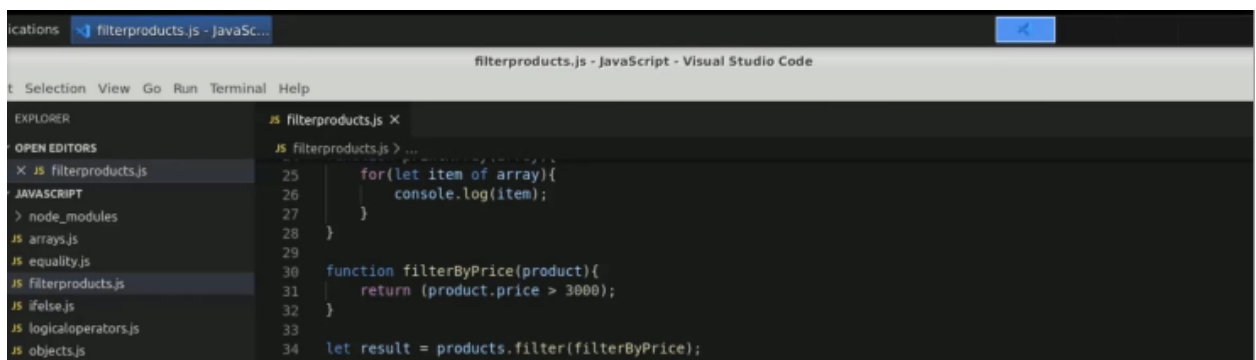
filterproducts.js - JavaScript - Visual Studio Code

EXPLORER
  filterproducts.js
  arrays.js
  equality.js
  filterproducts.js
  ifelse.js
  logicaloperators.js
  objects.js
  package-lock.json
  sorting.js
  strings.js

OPEN EDITORS
  filterproducts.js

25   for(let item of array){
26     console.log(item);
27   }
28 }
29
30 function filterByPrice(product){
31   return (product.price > 3000);
32 }
33
34 products.filter(filterByPrice);
35
36 printArray(products);
  
```

1.6 Save the filtered products in a variable such as **result**



```

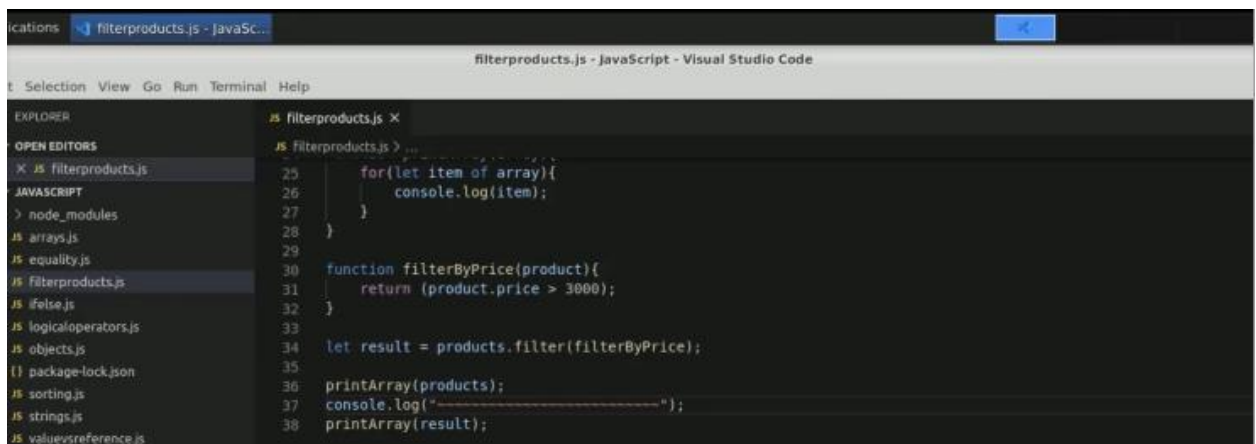
filterproducts.js - JavaScript - Visual Studio Code

EXPLORER
  filterproducts.js
  arrays.js
  equality.js
  filterproducts.js
  ifelse.js
  logicaloperators.js
  objects.js

OPEN EDITORS
  filterproducts.js

25   for(let item of array){
26     console.log(item);
27   }
28 }
29
30 function filterByPrice(product){
31   return (product.price > 3000);
32 }
33
34 let result = products.filter(filterByPrice);
  
```

1.7 Write a statement to display the filtered results on the console. Add a log statement as a delimiter or separator for better visibility of the results



```

filterproducts.js - JavaScript - Visual Studio Code

EXPLORER
  filterproducts.js
  arrays.js
  equality.js
  filterproducts.js
  ifelse.js
  logicaloperators.js
  objects.js
  package-lock.json
  sorting.js
  strings.js
  valuesreference.js

OPEN EDITORS
  filterproducts.js

25   for(let item of array){
26     console.log(item);
27   }
28 }
29
30 function filterByPrice(product){
31   return (product.price > 3000);
32 }
33
34 let result = products.filter(filterByPrice);
35
36 printArray(products);
37 console.log("-----");
38 printArray(result);
  
```

1.8 Run the program using the **node filterproducts.js** command to see the filtered results where the product price is greater than **3000**

```

25     for(let item of array){
26         console.log(item);
27     }
28 }
29
30 function filterByPrice(product){
31     return (product.price > 3000);
32 }
33
34 let result = products.filter(filterByPrice);
35
36 printArray(products);
37 console.log("-----");
38 printArray(result);

```

```

{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Adidas Ultraboot', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Cadbury 5 Star', price: 10 }

{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Adidas Ultraboot', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }

```

1.9 Modify the filtering rule inside the **filterByPrice** function to **product.price > 7000**

```

25     for(let item of array){
26         console.log(item);
27     }
28 }
29
30 function filterByPrice(product){
31     return (product.price > 7000);
32 }
33
34 let result = products.filter(filterByPrice);
35
36 printArray(products);
37 console.log("-----");
38 printArray(result);

```

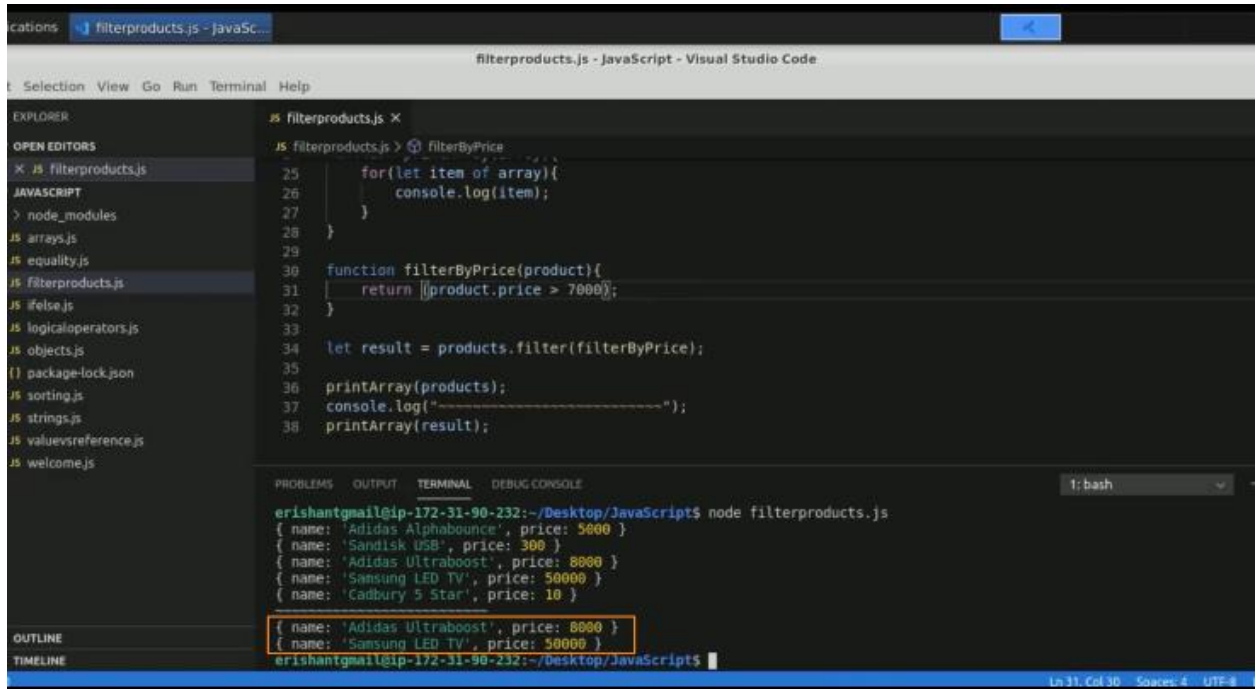
```

{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Adidas Ultraboot', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Cadbury 5 Star', price: 10 }

{ name: 'Adidas Ultraboot', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }

```

- 1.10 Run the program using the command **node filterproducts.js** to see the updated filtered results where the product price is greater than **7000**



The screenshot shows the Visual Studio Code interface with the file `filterproducts.js` open. The code defines a `filterByPrice` function and uses `products.filter` to filter an array of product objects. The terminal output shows the filtered results, with the last two items highlighted by a red box.

```
filterproducts.js - JavaScript - Visual Studio Code
t Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
X JS filterproducts.js
JAVASCRIPT
> node_modules
JS arrays.js
JS equality.js
JS filterproducts.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
() package-lock.json
JS sorting.js
JS strings.js
JS valuesreference.js
JS welcome.js

JS filterproducts.js X
JS filterproducts.js > filterByPrice
25   for(let item of array){
26     console.log(item);
27   }
28 }
29
30 function filterByPrice(product){
31   return product.price > 7000;
32 }
33
34 let result = products.filter(filterByPrice);
35
36 printArray(products);
37 console.log("-----");
38 printArray(result);

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: bash
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node filterproducts.js
{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Cadbury 5 Star', price: 10 }

{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

By following these steps, you have successfully created a function to filter an array of product objects based on specific criteria, allowing for more precise data manipulation and retrieval in your JavaScript applications.