

Lesson 6 Demo 1

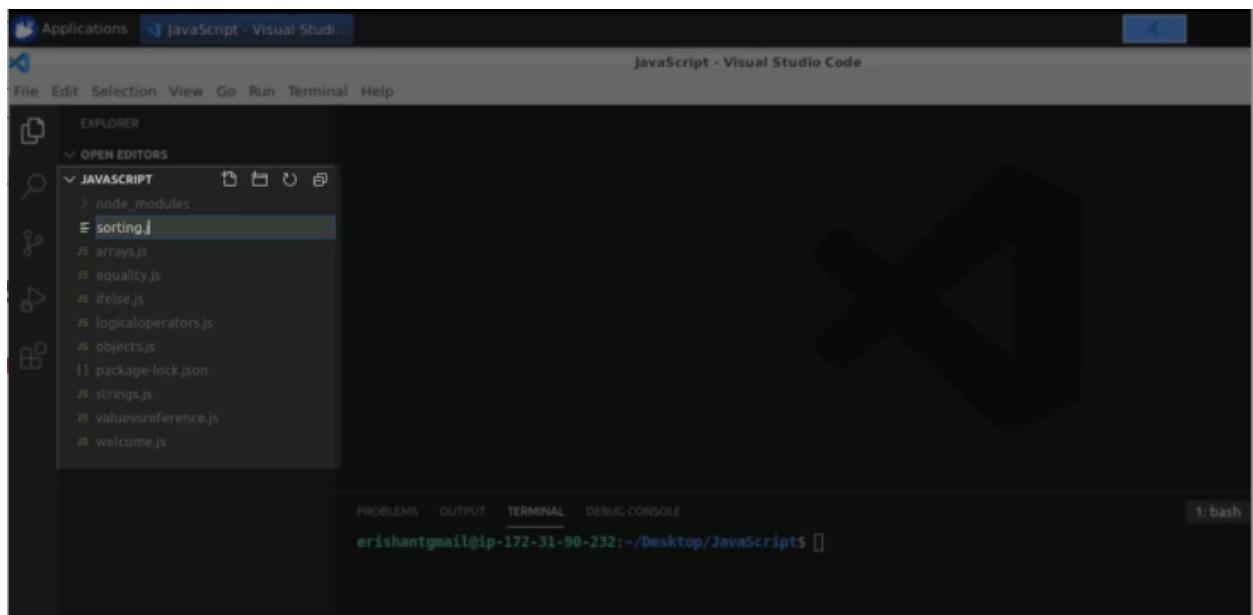
Creating Function to Sort the Product List Based on the Price

Objective: To create function to sort the product list based on the price

Prerequisites: None

Steps to perform:

1. Create a file named 'sorting.js'.



2. Create a variable called 'products'.

The screenshot shows a Visual Studio Code interface. The title bar says "simplilearnlabs.com:42001/guacamole/#/client/REVOQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacad...". The editor tab is "sorting.js - JavaScript ...". The code editor contains the following JavaScript code:

```
JS sorting.js •
JS sorting.js > (0) products
1 let products = [
2   [
3     [
4       [
5     ]
]
```

The Explorer sidebar on the left shows files like arrays.js, equality.js, ifelse.js, logicaloperators.js, objects.js, package-lock.json, sorting.js, strings.js, valuevsreference.js, and welcome.js.

3. Create an object with a name and price.

The screenshot shows the same Visual Studio Code interface. The code editor now contains:

```
JS sorting.js •
JS sorting.js > (0) products
1 let products = [
2   [
3     [
4       [
5         {
6           name: 'Adidas Alphabounce',
7           price: 5000
8         }
9       ]
10    ]
11  ]
12 ]
```

The Explorer sidebar remains the same.

4. Create some more objects with different names and prices.

```

    sorting.js
    6      [
    7        {
    8          name: 'Sandisk USB',
    9          price: 300
   10     },
   11     {
   12       name: 'Adidas Ultraboost',
   13       price: 8000
   14     },
   15     {
   16       name: 'Samsung LED TV',
   17       price: 50000
   18     },
   19     {
   20       name: 'Cadbury 5 Star',
   21       price: 10
   22   ],
   23
   24   function printArray(array){
   25     for(let item of array){
   26       console.log(item);
   27     }
   28   }
   29
   30   printArray(products);
  
```

- Print the array using the print array function using for loop.

```

    sorting.js
    19       name: 'Cadbury 5 Star',
   20       price: 10
   21     ],
   22   ],
   23
   24   function printArray(array){
   25     for(let item of array){
   26       console.log(item);
   27     }
   28   }
   29
   30   printArray(products);
  
```

- Run the program.

Applications sorting.js - JavaScript - ... sorting.js - JavaScript - Visual Studio Code

dit Selection View Go Run Terminal Help

EXPLORER JS sorting.js ×

OPEN EDITORS X JS sorting.js

JAVASCRIPT > node_modules

JS arrays.js JS equality.js JS ifelse.js JS logicaloperators.js JS objects.js () package-lock.json JS sorting.js JS strings.js JS valuevsreference.js JS welcome.js

```

19     name: 'Cadbury 5 Star',
20     price: 10
21   },
22 ]
23
24   function printArray(array){
25     for(let item of array){
26       console.log(item);
27     }
28   }
29
30   printArray(products);

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```
{
  name: 'Cadbury 5 Star', price: 10
}
[
  { 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 }
]
```

erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript\$

7. Use a function, to sort based on price.

Applications sorting.js - JavaScript - ... sorting.js - JavaScript - Visual Studio Code

dit Selection View Go Run Terminal Help

EXPLORER JS sorting.js ×

OPEN EDITORS X JS sorting.js

JAVASCRIPT > node_modules

JS arrays.js JS equality.js JS ifelse.js JS logicaloperators.js JS objects.js () package-lock.json JS sorting.js JS strings.js JS valuevsreference.js JS welcome.js

```

19     name: 'Cadbury 5 Star',
20     price: 10
21   },
22 ]
23
24   function printArray(array){}
25     for(let item of array){
26       console.log(item);
27     }
28   }
29
30   printArray(products);

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```
{
  name: 'Cadbury 5 Star', price: 10
}
[
  { 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 }
]
```

erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript\$

8. The product one is:

Applications sorting.js - JavaScript ... sorting.js - JavaScript - Visual Studio Code

dit Selection View Go Run Terminal Help

EXPLORER JS sorting.js ×

OPEN EDITORS × JS sorting.js

JAVASCRIPT > node_modules

JS arrays.js

JS equality.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

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```
{
  name: 'Samsung LED TV', price: 50000 },
  { name: 'Cadbury 5 Star', price: 10 }
]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node sorting.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 }
```

> OUTLINE

> TIMELINE

9. Pass a function 'sortOnPrice'.

Applications ● sorting.js - JavaScript ... sorting.js - JavaScript - Visual Studio Code

dit Selection View Go Run Terminal Help

EXPLORER JS sorting.js •

OPEN EDITORS 1 UNSAVED ● JS sorting.js

JAVASCRIPT > node_modules

JS arrays.js

JS equality.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

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```

  name: 'Cadbury 5 Star',
  price: 10
},
]

function printArray(array){
  for(let item of array){
    console.log(item);
  }
}

function sortOnPrice([product1, product2]) {
  if (product1.price < product2.price) {
    return product1;
  } else {
    return product2;
  }
}

printArray(products);
}
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node sorting.js
{ name: 'Samsung LED TV', price: 50000 },
{ name: 'Cadbury 5 Star', price: 10 }
]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

> OUTLINE

> TIMELINE

10. Perform sort operation on custom objects.

```

    sorting.js
    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 }
    33
    34 products.sort(sortOnPrice)
    35
    36 printArray(products);

    PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
    { name: 'Samsung LED TV', price: 50000 },
    { name: 'Cadbury 5 Star', price: 10 }
    ]
    erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node sorting.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 }
    erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 
  
```

11. Execute the code.

```

    sorting.js
    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 }
    33
    34 products.sort(sortOnPrice);
    35
    36 printArray(products);

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

12. Update the 'return' statement.

Applications sorting.js - JavaScript - ...

sortOnPrice

```

    sorting.js > sortOnPrice
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 products.sort(sortOnPrice);
36
37 printArray(products);

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Cadbury 5 Star', price: 10 }
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node sorting.js
{ name: 'Cadbury 5 Star', price: 10 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 

```

OUTLINE TIMELINE

13. Re-run the code.

Applications sorting.js - JavaScript - ...

sortOnPrice

```

    sorting.js > sortOnPrice
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 products.sort(sortOnPrice);
36
37 printArray(products);

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Samsung LED TV', price: 50000 }
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node sorting.js
{ name: 'Samsung LED TV', price: 50000 }
{ name: 'Adidas Ultraboost', price: 8000 }
{ name: 'Adidas Alphabounce', price: 5000 }
{ name: 'Sandisk USB', price: 300 }
{ name: 'Cadbury 5 Star', price: 10 }
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 

```

OUTLINE TIMELINE