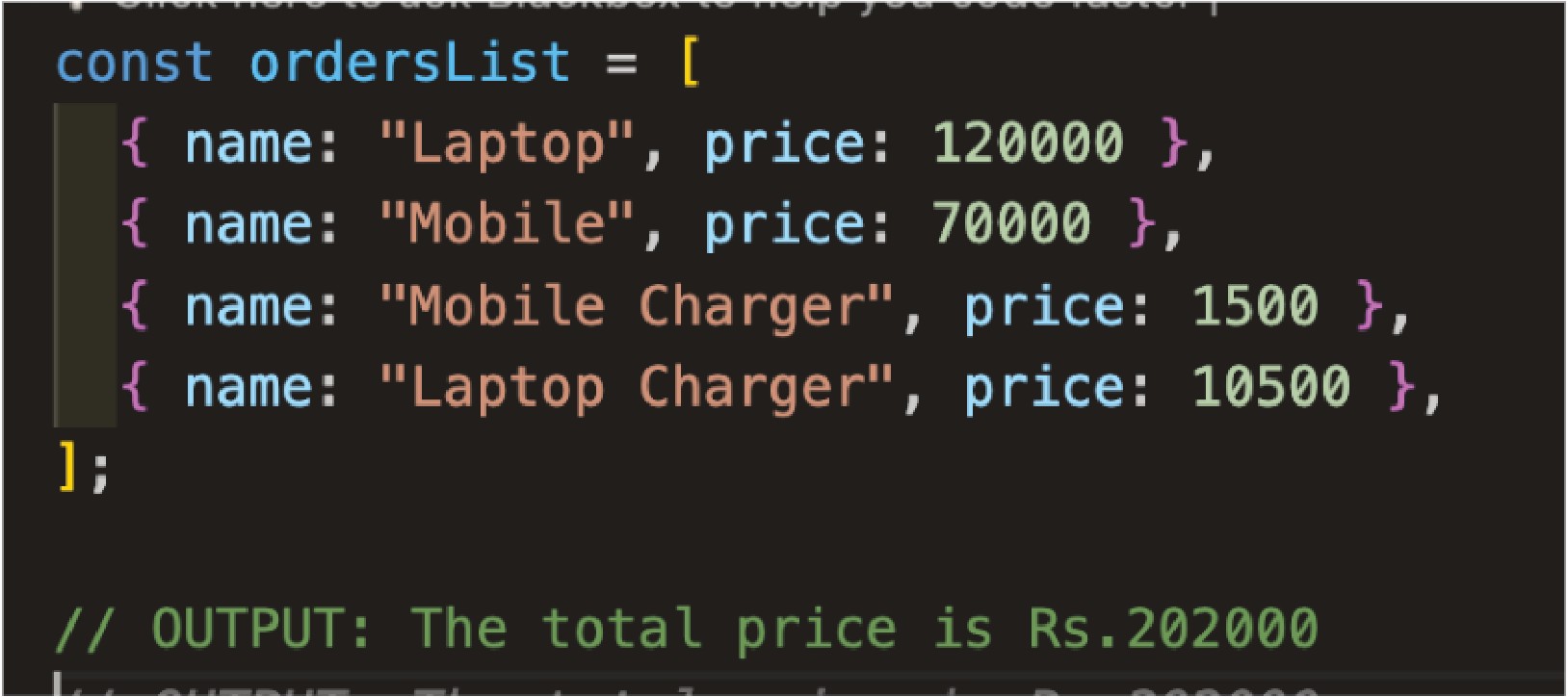
1. You are building an e-commerce website. Write a function that calculates the total price of a customer's order. You're given an array of items, each with a price property. Use the forEach method to iterate through the array and sum up the prices to get the total order amount



**Solution:** ..,.

const ord e rs L ist = [

{ name : " La ptop " , p r ice : 120000 } ,

{ name : "Mobile " , p r ice : 70000 },

{ name : "Mobile Cha r ge r " , p r ice : 1500 },

{ name : " Laptop Cha r ge r " , p r ice : 10500 }.

];

let sum = 0 ;

o rdersList . forEa ch ( ( data ) => ( sum += da ta . p r ice ) ) ; console . log ( 'The tota l p r ice is Rs . ${sum} ' ) ;

1. Inthis challenge, your task is to create a function that generates a random number and prints it to the console every 2 seconds. The program should keep printing new random numbers indefinitely, with a 2- second delay between each number.

**Solution:**

funct ion ra nd omNu m be r Gene ra t o r ( ) {

const ra n dom N u m ber = Mat h .floor ( Ma t h . ra ndom ( ) \* 100) ; console . log ( r andom Numbe r )

function p rin tWit h in t erva l( ) { ra ndom Numbe rGene r a to r ( ) ;

setint erva l( r andom Numbe rGene r ator , 2000 ) ;



1. You are given an array of expense objects representing monthly expenses. Each object has properties, amount and category. Use the map method to create a new array that includes the calculated tax for each expense. Assume a tax rate of 10%.

Solution:

let expenses *=* [

{ amou nt : 100 , catego ry : " Utilities " },

{ amou nt : 200 , catego r y : "G roce r ies " },

{ amou nt : 50 , ca tego ry : " Ente r t ainment " },

l ;

let expensesWith Ta x = expenses . ma p ( ( expense ) => { let t ax = expense . amou nt \* 0 . 1 ;

ret u rn { . . . expense , t ax : tax } ;

}) ;

console . log ( " Expe nses wit h Tax : " , expen sesWit hTax ) ;

output:

Expenses with Tax: [

{ amount: 100, category : 'Utilities ', tax: 10 },

{ amount: 200, category: 'Groceries', tax : 20 },

{ amount: 50, category: 'Entertainment ', tax: 5 }

1. Using the same array of expense objects,use the filter method to create a new array that includes only the expenses related to the category "Groceries.''

Solution:

JavaScript

### let expen ses =

{ amou nt : 100 , c at ego ry : " Utilit ies " },

{ amou n t : 200 , c at ego ry : "G roce r ies " },

{ amou n t : 50 , cat ego ry : " E nte r t ainmen t " },

l ;

let g r ace r iesE xpe n ses = ex pen ses . f ilte r (

( expe n se) = > expe n se . cat ego ry = == " G roce r ies "

) ;

con sole . log ( " G roce r ies Expe n ses : " , g r ocer ies Expe nses ) ;

### Ou tput :

G roce r ies Expenses: [ { amou nt : 200 , catego ry: ' G roce r ies ' }

1. Using the same array of expense objects, use the reduce method to calculate the total amount of all expenses.

**Solutior** JavaScript .,.

let expenses =

{ amount: 1 00, category: "Utilities" },

{ amount: 200, category: "Groceries" },

{ amount: 50, category: "Entertainment" },

];

let totalExpenses = expenses. reduce ((acc, expense) => ace + expense. amount, 0); console. log ("Total Expenses: " , totalExpenses) ;

//Output :

Total Expenses: 350

1. You have a list of expenses, each with an amount and a category. Now, create a function named categorizeExpense that, based on the expense amount, returns either "High Expense" if it's more than 100, or "Low Expense" otherwise. Afterward, use this function along with the map method to generate a new array called categorizedExpenses, where each element represents the category for the corresponding expense in the original list. Finally, print out the categorizedExpenses array.

let expenses = [

{ amount: 100, category: "Utilities " },

{ amount: 200, category: "Groceries" },

{ amount: 50, category: "Entertainment " },

l ;

function categorizeExpense (expense) { if (expense.amount > 100) {

return "Hig h Expense" ; else {

return "Low Expense" ;

}

}

let categorized Expenses = expenses. map ((expense) =>

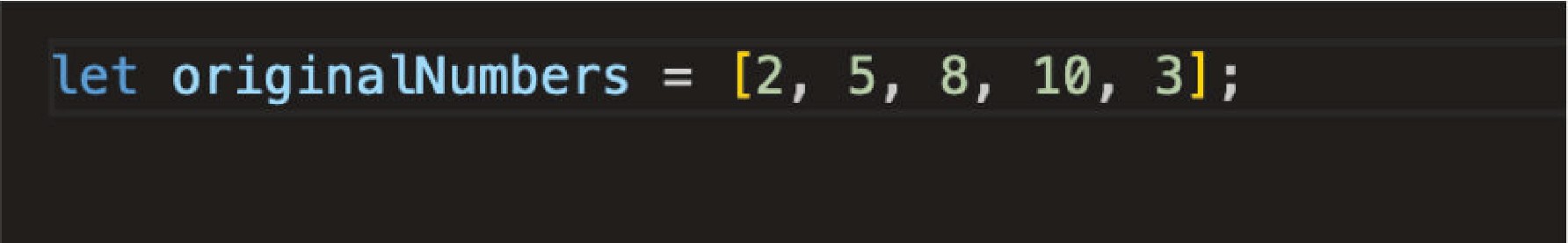
categorizeExpense (expense));

console .log ("Categorized Expenses: " , categorizedExpenses) ;

//Output :

Categorized Expenses: [ 'Low Expense ' , 'High Expense ', 'Low Expense' ]

1. Consider an array of numbers named originalNumbers with the values [2, 5, 8, 10, 3]. Your task is to use the forEach method to iterate through each element in the array. During the iteration, double the value of each number. After completing the iteration,display the modified array.



**Solution**

#### let originalNumbers = [2, 5, 8, 10, 3 ];

originalNumbers. forEach ((number, index, a rray) => {

array[indexJ = number \* 2;

} ) ;

#### console.log ("Doubled Numbers: " , originalNumbers) ;

Output: Doubled Numbers : [ 4, 10, 16, 20, 6 I

8. Using the same array of numbers, use the forEach method to collect and store only the even numbersin a new array.

Solution:

let evenNumbe rs = [ ] ;

o riginalNumbers \_ forEach (number => {

if ( numbe r % 2 === 0 ) { evenNumbers . push ( number ) ;

}

} ) ;

console.log ( "Even Numbers :" , ev enNumber s ) ;

Output: Even Numbers: I 2, 8, 10 ]