

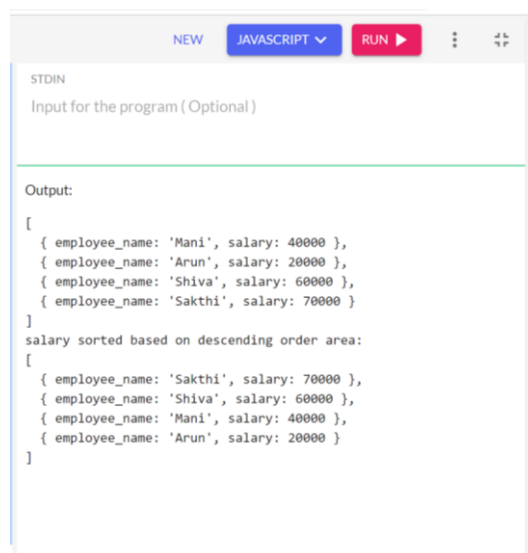
## OUTPUT

1. Write a JavaScript to convert a comma-separated values (CSV) String to a 2D array.  
A new line indicates a new row in the array.

Output:

```
a,2  
c,d
```

2. Write a program to display employee objects based on salary (descending order). [use Objects]



The screenshot shows a JavaScript IDE with a toolbar at the top containing 'NEW', 'JAVASCRIPT', 'RUN', and other icons. Below the toolbar is a section for 'STDIN' with a text input field labeled 'Input for the program (Optional)'. The main area displays the 'Output' of the program, which is a JSON array of employee objects sorted by salary in descending order.

```
Output:  
  
[  
  { employee_name: 'Mani', salary: 40000 },  
  { employee_name: 'Arun', salary: 20000 },  
  { employee_name: 'Shiva', salary: 60000 },  
  { employee_name: 'Sakthi', salary: 70000 }  
]  
salary sorted based on descending order area:  
[  
  { employee_name: 'Sakthi', salary: 70000 },  
  { employee_name: 'Shiva', salary: 60000 },  
  { employee_name: 'Mani', salary: 40000 },  
  { employee_name: 'Arun', salary: 20000 }  
]
```

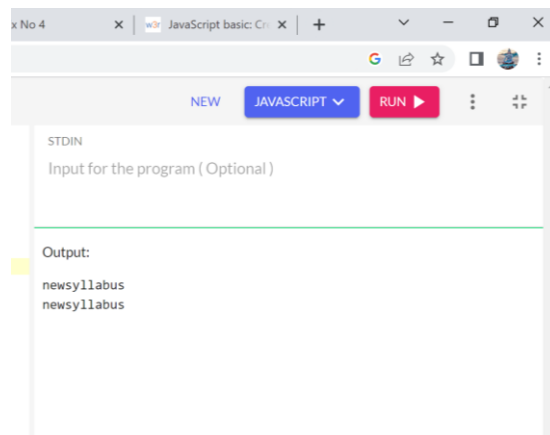
3. Write a JavaScript program to remove all of the numbers from 'A1B2C3D4E5F6G7H8I9J10' [Use Regular expression]



The screenshot shows a web browser window with the following content:

```
ALL NUMBERS FROM THE TEXT
The text is 'A1B2C3D4E5F6G7H8I9J10'
After removing all the numbers:
A,B,C,D,E,F,G,H,I,J
```

4. Write a JavaScript programs to create a new String adding “New!” in front of a given String. If the given String begins with “New!” already then return the original String.



The screenshot shows a JavaScript IDE window with the following content:

```
STDIN
Input for the program ( Optional )

Output:
newsyllabus
newsyllabus
```

