

main.py

Share

Run

```
1
2 #EX-1
3 l=[1,2,3,4,5]
4 def square_all(numbers):
5     return numbers**2
6
7 res=map(square_all,l)
8
9 print(list(res))
10
11 #using lambda function
12
13 res=map(lambda numbers:numbers**2,l)
14
15 print(list(res))
16
17
18
19
20
```

Output

```
[1, 4, 9, 16, 25]
[1, 4, 9, 16, 25]

=== Code Execution Successful ===
```

main.py

Share

Run

```
1
2 #EX-2
3 l=[1,2,3,0,-1,-2,-3]
4 def filter_positive(numbers):
5     return numbers>=0
6
7 res=filter(filter_positive,l)
8
9 print(list(res))
10
11 # using lambda function
12
13 res=filter(lambda numbers:numbers>=0,l)
14 print(list(res))
15
```

Output

[1, 2, 3, 0]

[1, 2, 3, 0]

=== Code Execution Successful ===

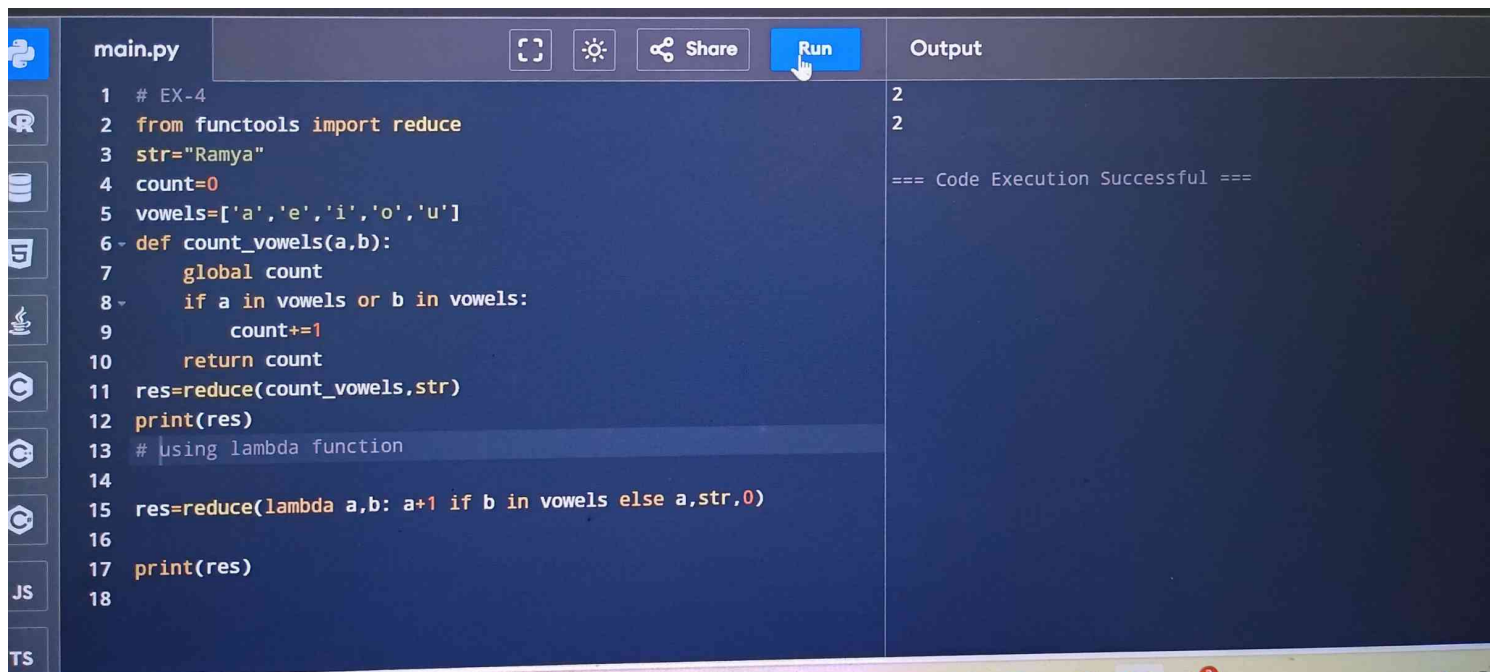
The image shows a Jupyter Notebook interface with a dark theme. The left sidebar contains icons for various data science tools. The main area is divided into two panels: a code editor on the left and an output panel on the right.

Code Editor (main.py):

```
1 # EX-4
2 from functools import reduce
3 n=int(input("enter num:"))
4 l=range(n,0,-1)
5
6 def calculate_factorial(a,b):
7     return a*b
8
9 res=reduce(calculate_factorial,l)
10 print(res)
11
12 # using lambda
13
14 res=reduce(lambda a,b:a*b,l)
15 print(res)
16
17
18
19
```

Output Panel:

```
enter num:4
24
24
=== Code Execution Successful ===
```



```
main.py  Run  Output
1 # EX-4
2 from functools import reduce
3 str="Ramya"
4 count=0
5 vowels=['a','e','i','o','u']
6 def count_vowels(a,b):
7     global count
8     if a in vowels or b in vowels:
9         count+=1
10    return count
11 res=reduce(count_vowels,str)
12 print(res)
13 # using lambda function
14
15 res=reduce(lambda a,b: a+1 if b in vowels else a,str,0)
16
17 print(res)
18
```

2
2
=== Code Execution Successful ===