Day 15 Lambda Funtion

A lambda function is a small anonymous function.

Lambda function can take any number of arguments, but only have one expression.

Syntax

Lambda arguments = expression

E.X

```
data = lambda a, b : a+b.

result = data(30, 20)

Print (result)
  ↳ 50.
```

Conditional class in Lambda

```
result = lambda num : num % 2 == 0

Data = (result (17))
Print (Data)
```

```
If data == True:
    print ('The number is even')

else:
    print ('The number is odd').
```

↳ Output False
The number is odd

Use def funtion

```
def myfun(n):
    return lambda a : a * n

val = myfun(3)
print(val(10))
```

Output ↳ 30.

map function

· map() functions returns a result after applying the given function to each item of a given iterable.

Syntax :

```
map (fun, iter)
```

E.X get the square values

```
def square (n)
    print ('n = {}'.format(n))
    return (n**2)

for i in map (square [1, 2, 3, 4, 5])

    print (i)
```

↳

Output

```
n = 1
1
n = 2
4
n = 3
9
n = 4
16
n = 5
25
```

Get Started | Day15_Practice.py

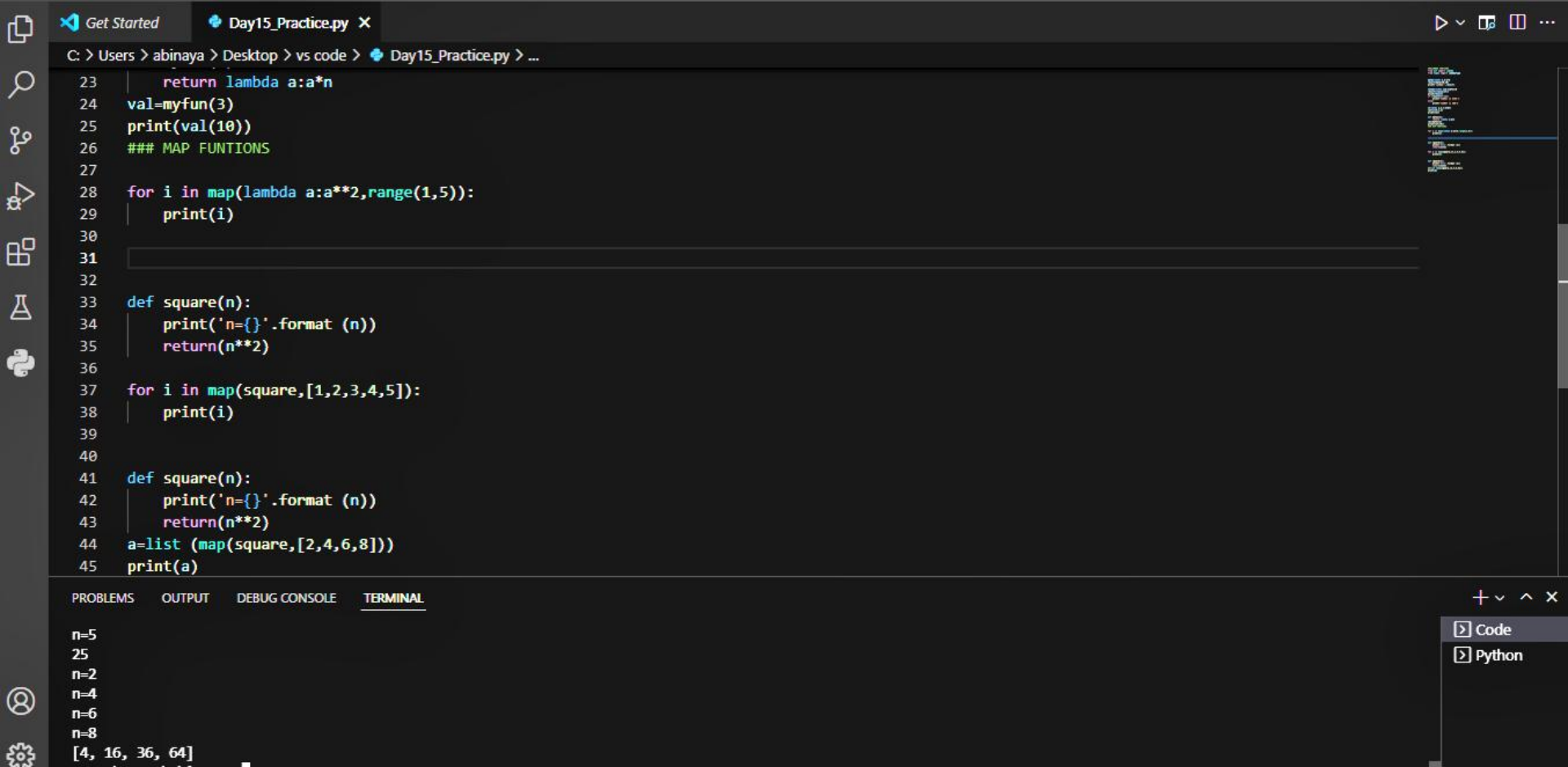
C: > Users > abinaya > Desktop > vs code > Day15_Practice.py > ...

```python
###LAMBDA FUNTIONS
from ast import Lambda
from types import LambdaType


data=lambda a,b:a+b
result=data(30,20)
print('Lambda:',result)

value=lambda num:num%2==0
result=(value(17))
print(result)
if (result==True):
    print('number is even')
else:
    print('number is odd')

x=lambda a,b,c:a*b*c
val=x(10,2,3)
print(val)

def myfun(n):
    return lambda a:a*n
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
False
number is odd
60
30
1
4
9
16
```

Code
Python


Get Started
Day15_Practice.py
C: > Users > abinaya > Desktop > vs code > Day15_Practice.py > ...
return lambda a:a*n
val=myfun(3)
print(val(10))
MAP FUNTIONS
for i in map(lambda a:a**2,range(1,5)):
print(i)
def square(n):
print('n={}'.format (n))
return(n**2)
for i in map(square,[1,2,3,4,5]):
print(i)
def square(n):
print('n={}'.format (n))
return(n**2)
a=list (map(square,[2,4,6,8]))
print(a)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
n=5
25
n=2
n=4
n=6
n=8
[4, 16, 36, 64]
Code
Python