Day 31

Lambda functions in Python:

What is a Lambda Function?

A lambda function is a small, anonymous function (a function without a name).

- It can take any number of arguments
- It can have only one expression
- The result of that expression is automatically returned

Think of it as a one-line function.

Normal Function vs Lambda Function

Normal function

```
def add(a, b):
    return a + b
```

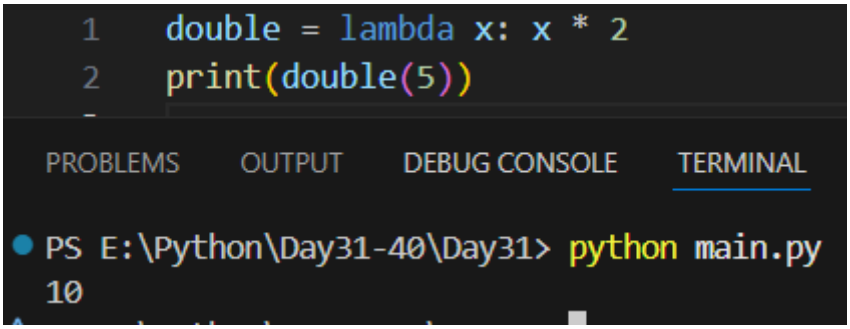
Lambda function

```
add = lambda a, b: a + b
```

Basic Syntax:

```
lambda arguments: expression
```

Example: a basic use case of lambda function.



Example: multiple arguments in lambda.

```
4   sum = lambda a, b, c: a + b + c
5   print(sum(1, 2, 3))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS E:\Python\Day31-40\Day31> python main.py
6
```

Example: Lambda Without Assigning to a Variable

```
7   print((lambda x: x ** 2)(4))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  P

PS E:\Python\Day31-40\Day31> python main.py
16
```

Map, Filter and Reduce in Python:

map() – *Transform each item*

map() applies a function to every element in an iterable (like a list).

Syntax:

map(function, iterable)

Example: without using them making cube of each of the list.

```
9   old = [1,2,3]
10  new = []
11  for i in old:
12      new.append(i*i*i)
13  print(new)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS E:\Python\Day31-40\Day31> python main.py
[1, 8, 27]
```

Example: doing the same as above using map().

```
9   def cube(x):
10      return x*x*x
11  old = [1,2,3,4,5]
12  new = list(map(cube,old))
13  print(new)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS E:\Python\Day31-40\Day31> python main.py
[1, 8, 27, 64, 125]
```

Example: another example of map.

```
15  numbers = [1, 2, 3, 4, 5]
16  def square(x):
17      return x * x
18  result = map(square, numbers)
19  print(list(result))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS E:\Python\Day31-40\Day31> python main.py
[1, 4, 9, 16, 25]
```

Example: map with lambda.

```
21  numbers = [1, 2, 3, 4, 5]
22  result = map(lambda x: x * x, numbers)
23  print(list(result))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PO
PS E:\Python\Day31-40\Day31> python main.py
[1, 4, 9, 16, 25]
```

filter() – *Select items that match a condition*

filter() keeps **only the elements** that return True from a function.

Syntax

filter(function, iterable)

Example:

```
25  numbers = [1, 2, 3, 4, 5, 6]
26  def is_even(x):
27      return x % 2 == 0
28  result = filter(is_even, numbers)
29  print(list(result))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS E:\Python\Day31-40\Day31> python main.py
[2, 4, 6]
```

Example: filter() with lambda

```
31  numbers = [1, 2, 3, 4, 5, 6]
32  result = filter(lambda x: x % 2 == 0, numbers)
33  print(list(result))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Python\Day31-40\Day31> python main.py
[2, 4, 6]
```

reduce() – *Combine all items into one*

reduce() reduces a list to a single value by applying a function cumulatively.

Note: reduce() is not built-in; you must import it.

Syntax

from functools import reduce
reduce(function, iterable)

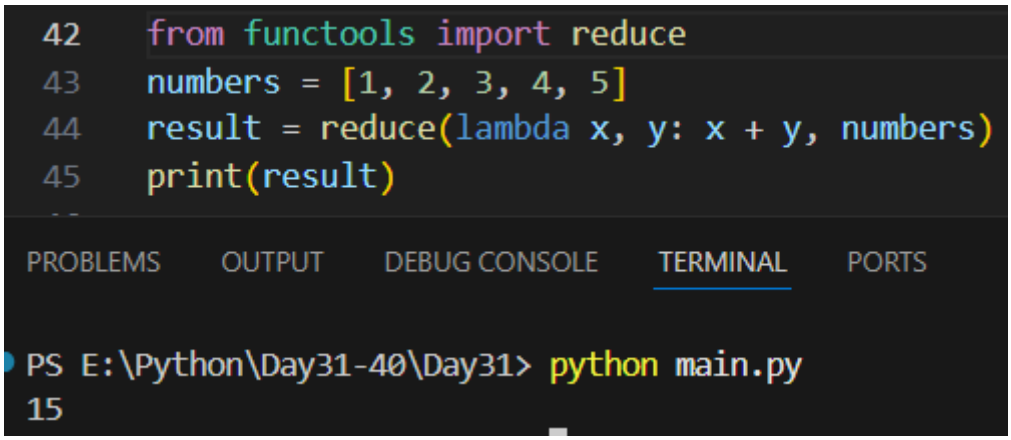
Example:

```
35  from functools import reduce
36  numbers = [1, 2, 3, 4, 5]
37  def add(x, y):
38      return x + y
39  result = reduce(add, numbers)
40  print(result)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS E:\Python\Day31-40\Day31> python main.py
15
```

Example: reduce() and lambda.



Map vs Filter vs Reduce (Simple Comparison)

Function	Purpose	Input	Output
map	Transform items	Iterable	New iterable
filter	Select items	Iterable	Fewer items
reduce	Combine items	Iterable	Single value

--The End--