Title : Python Basic Assignment-24
Name : Bhavikkumar Modi
E-mail : bmodi700@gmail.com
Website : https://keenbm.github.io/

1. What is the relationship between def statements and lambda expressions ?

The def statements are defined with a name as well as are required to pass a value to them. After execution, a return keyword is required to give the result back to the user.

In a lambda function, we do not give a return keyword.Always an expression is returned. It can be used at multiple places. We do not need to assign a variable to it.

2. What is the benefit of lambda?

- Lambda functions reduce the number of lines of code.
- They are generally used when a function is needed temporarily for a short period of time, often to be used inside another function such as filter, map and reduce.
- Using the lambda function, you can define a function and call it immediately at the end of definition. This can't be done with def functions.

3. Compare and contrast map, filter, and reduce.

- The map() function iterates through all items in the given iterable and executes the function we passed as an argument on each of them.
- filter() takes a function object and an iterable and creates a new list.It forms a new list that contains only elements that satisfy a certain condition, i.e. the function we passed returns True.
- reduce() works differently than map() and filter(). It does not return a new list based on the function and iterable we've passed. Instead, it returns a single value.

4. What are function annotations, and how are they used?

Function annotations are arbitrary python expressions that are associated with various parts of functions. These expressions are evaluated at compile time and have no life in python's runtime environment.

```
def fib(n:'int', output:'list'=[])-> 'list':
```

5. What are recursive functions, and how are they used?

Recursive functions are functions that have the capability to call themselves.

```
def factorial(x):

    if x == 1:

        return 1

    else:
        return (x * factorial(x-1))

num = 3

print("The factorial of", num, "is", factorial(num))
```

6. What are some general design guidelines for coding functions?

- Limited use of globals
- Standard headers for different modules
- Naming conventions for local variables, global variables, constants and functions:
- Indentation
- Error return values and exception handling conventions

7. Name three or more ways that functions can communicate results to a caller.

- By using the return keyword at the end of the function.
- By using the print statement.
- By returning a function to the end user.
- By using yield