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

Both of them use for creating a function but lambda create an anonymous function.

1. What is the benefit of lambda?

Lambda let us to create single line function which will use less space and code and also we can use it when we need a function as an input for example in map

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

* All three expect a function object as first argument, and iterable(s)
* Map will go through iterable and run function for all of them but filter go through iterable and it return new list of them which pass the condition and reduce get the iterable but during running use the value which return by previous iter and use it for calculation in new iter

- Reduce is not a build-in function any more but map and filter are built-in functions

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

Function annotations, available in 3.X , are syntactic embellishments of a function’s arguments and result, which are collected into a dictionary assigned to the function’s \_\_annotations\_\_ attribute.

Python places no semantic meaning on these annotations, but simply packages them for potential use by other tools.

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

It is a function that calls itself until reach to desired result and it work similar to loop

def add(a) :

if a > 1 :

return a + add(a-1)

else:

return a

print(add(5))

this function return

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

- Functions should generally be small and as self-contained as possible,

- have a single unified purpose,

- communicate with other components through input arguments and return values.

- They may use mutable arguments to communicate results too if changes are expected,

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

Return, yield, print inside function, passing mutable arguments the variable outside of the function, using global variables .