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

Lambda expressions are one-line methods without a name. Lambdas differ from normal Python methods because they can have only one expression,

can't contain any statements and their return type is a function object.

Keyword def statements commence the start of the function header followed by a function name to uniquely identify the function.

Function naming follows the same rules of writing identifiers in python.

2. What is the benefit of lambda?

Benefits of lamda function are :

Lesser Lines of Code.

Lambda functions are inline functions and so executes comparatively faster.

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

Map applies as a transformation to an element.

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 accumulates only elements matching a condition.

filter() forms a new list that contains only elements that satisfy a certain condition, i.e. the function we passed returns True.

Reduce accumulates all elements to a single value, by using immutable values.

reduce() works by calling the function we passed for the first two items in the sequence. The result returned by the function is used in another call to function alongside with the next (third in this case), element.

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

Function annotation is the standard way to access the metadata with the arguments and the return value of the function.

These are nothing but some random and optional Python expressions that get allied to different parts of the function.

They get evaluated only during the compile-time and have no significance during the run-time of the code.

They do not have any significance or meaning associated with them until accessed by some third-party libraries.

They are used to type check the functions by declaring the type of the parameters and the return value for the functions.

The string-based annotations help us to improve the help messages.

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

A recursive function is a function that calls itself during its execution. This means that the function will continue to call itself and repeat its behaviour until some condition is met to return a result.

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

Naming Conventions.

Name your classes and functions consistently. While naming of function of methods always use self for the first argument.

Use of regular and updated comments are valuable to both the coders and users.

Use 4-space indentation and no tabs.

Don’t use non-ASCII characters in identifiers.

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

Functions can return multiple values, tuple.

Functions can return class object.

Functions can return list, dictionary.

Functions can return function object.

Function can return single value.