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

As an expression, lambda returns a value that can optionally be assigned a name. In contrast, the def statement always assigns the new function to the name in the header, instead of returning is as a result. lambda's body is a single expression, not a block of statements.

1. What is the benefit of lambda?

One of the benefits of using lambda expression is the reduced amount of code. See the example below. In this case, we are not passing any parameter in lambda expression because the run() method of the functional interface (Runnable) takes no argument.

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

MAP function to each element of iterator and collects result.

FILTER applies function to each element of iterator and collects those elements for which function returns true. For this function need to return boolean value.

REDUCE applies rolling computation to sequential pair of elements in iterator. Initially it takes two elements from iterator in sequence, applies function, collect result, then take next element in sequence from iterator, applies function and repeat this untill list has single value.

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

Function annotations are arbitrary python expressions that are associated with various part of functions. These expressions are evaluated at compile time and have no life in python's runtime environment. Python does not attach any meaning to these annotations.

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

Recursion is made for solving problems that can be broken down into smaller, repetitive problems. It is especially good for working on things that have many possible branches and are too complex for an iterative approach. One good example of this would be searching through a file system.

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

Safe: It can be used without causing harm.

Secure: It can't be hacked.

Reliable: It functions as it should, every time.

Testable: It can be tested at the code level.

Maintainable: It can be maintained, even as your codebase grows.

Portable: It works the same in every environment.

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

1)Built-in functions, such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal,… You can find an overview with more of these functions.

2)User-Defined Functions (UDFs), which are functions that users create to help them out; And

3)Anonymous functions, which are also called lambda functions because they are not declared with the standard def keyword.