1. What is the difference between enclosing a list comprehension in square brackets and parentheses?

The difference between the two kinds of expressions is that the List comprehension is enclosed in square brackets [] while the Generator expression is enclosed in plain parentheses ().

l = [n\*2 for n in range(1000)] # List comprehension

g = (n\*2 for n in range(1000)) # Generator expression

1. What is the relationship between generators and iterators?

In creating a python generator, we use a function. But in creating an iterator in python, we use the iter() and next() functions.

A generator in python makes use of the ‘yield’ keyword. A python iterator doesn’t.

Python generator saves the states of the local variables every time ‘yield’ pauses the loop in python. An iterator does not make use of local variables, all it needs is iterable to iterate on.

A generator may have any number of ‘yield’ statements.

You can implement your own iterator using a python class; a generator does not need a class in python.

To write a python generator, you can either use a Python function or a comprehension. But for an iterator, you must use the iter() and next() functions.

Generator in python let us write fast and compact code. This is an advantage over Python iterators. They are also simpler to code than do custom iterator.

1. What are the signs that a function is a generator function?

If a function contains at least one yield statement (it may contain other yield or return statements), it becomes a generator function. Both yield and return will return some value from a function.

1. What is the purpose of a yield statement?

Yield is a keyword in Python that is used to return from a function without destroying the states of its local variable and when the function is called, the execution starts from the last yield statement. Any function that contains a yield keyword is termed a generator.

1. What is the relationship between map calls and list comprehensions? Make a comparison and contrast between the two.

List comprehension is more concise and easier to read as compared to map.

List comprehension allows filtering. In map, we have no such facility. For example, to print all even numbers in range of 100, we can write [n for n in range(100) if n%2 == 0]. There is no alternate for it in map

List comprehension are used when a list of results is required as map only returns a map object and does not return any list.

List comprehension is faster than map when we need to evaluate expressions that are too long or complicated to express

Map is faster in case of calling an already defined function (as no lambda is required).

Comparing Execution Time