**Assignment\_25**

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

A) Another way brackets and parentheses differ is that square brackets can describe a list comprehension, e.g. [x for x in y] Whereas the corresponding parenthetic syntax specifies a tuple generator: (x for x in y) See: Why is there no tuple comprehension in Python? The first is a list, the second is a tuple. Lists are mutable, tuples are not.

2) What is the relationship between generators and iterators?

A) An iterator is typically something that has a next method to get the next element from a stream. A generator is an iterator that is tied to a function. This has the advantage that you don't need to store infinite numbers in memory to iterate over them. for i in genCountingNumbers (): print i if i > 20: break

3) What are the signs that a function is a generator function?

A) A generator function in Python is defined like a normal function, but whenever it needs to generate a value, it does so with the yield keyword rather than return. If the body of a def contains yield, the function automatically becomes a Python generator function.

4) What is the purpose of a yield statement?

A) yield keyword is used to create a generator function. A type of function that is memory efficient and can be used like an iterator object. In layman terms, the yield keyword will turn any expression that is given with it into a generator object and return it to the caller.

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

A) As you can see, both the list comprehension and the map function produce the same output. However, the map function returns a map object, which is similar to a generator. This makes it more memory efficient compared to a list comprehension. To retrieve the values, we need to convert the map object to a list.