

1.

() is a tuple: An immutable collection of values, usually (but not necessarily) of different types. [] is a list: A mutable collection of values, usually (but not necessarily) of the same type.

2.

Iterators: Iterator are objects which uses next() method to get next value of sequence.

Generators: A generator is a function that produces or yields a sequence of values using yield method.

3.

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.

4.

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.

5.

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