1. Square brackets comprehension return a list whereas () brackets comprehension return a generator object which can be iterated through next().

2. Iterators are the objects that use the next() method to get the next value of the sequence. A generator is a function that produces or yields a sequence of values using a yield statement. Both have a common relationship that they use next() to traverse through.

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. The yield statement returns a generator object to the one who calls the function which contains yield, instead of simply returning a value.

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).