In [ ]:
```
##Pallavi Tembare##
#1.Difference between Generator function and normal function
1.Normal function has only one return statement in the loop whereas generator function can use one or more yield
  in the loop
2.Generator function are better i case of memory utilisation and code performance because they allow the functio
  doing all work at a time
3.A function executes when you call it,it returns a value and then its over.No saved state.
  a generator function has yield statement.When it hits that it return a value but saves its internal state.This
  to compute sequence
4.Generator function have lazy execution ( producing items only when asked for )
```

In [ ]:
```
#2.Difference between iterator and generator
1.A generator function returns us a sequence of values to iterate on whereas iterator returns us an iterator obj
  one value at a time
2.In creating a generator, we use a function or comprehension. But in creating an iterator, we use the iter() an
3.Generator saves the states of the local variables every time 'yield' pauses the loop.An iterator does not make
 local variables, all it needs is iterable to iterate on.
4.Every generator is an iterator, not every python iterator is a generator
```

In [8]:
```
#3.Different methods to update dictionary
#1.update() When dictionary is Passed
dict1 = {'Name': 'Sonali', 'Age': 27}
dict2 = {'Sex': 'female' }
dict1.update(dict2)
print('update() When dictionary is Passed:',dict1)

#2.update() When Tuple is Passed
d = {'x': 2}
d.update(y = 3, z = 0)
print('update() When dictionary is Passed:',d)

#3.update when key exists
d = {1: "one", 2: "three"}
d1 = {2: "two"}
d.update(d1)
print('update() when key exists:',d)

#4.Update the key name in dictionary
word_freq = {
    "Hello": 56,
    "at": 23,
    "test": 43,
    "this": 78
}
value = word_freq.pop('at')
word_freq.update({'where': value})
print('Update the key name in dictionary:',word_freq)


#5.Update values of a list of dictionaries using append method
data = [
    {'name': 'Saurabh', 'subjects': ['java', 'python']},
    {'name': 'Tushar', 'subjects': ['c/cpp', 'java']},
    {'name': 'Ajit', 'subjects': ['iot', 'cloud']}
]
print('first student:',data[0])
data[0]['subjects'].append('html')
data[0]['subjects'].pop(1)
print('first student after update value using append method:',data[0])

#6.Update values of a list of dictionaries using insert
```

```
print('third student:',data[2])
data[2]['subjects'].insert(0, 'dbms')
data[2]['subjects'].pop(1)
print('third student after update value using insert method:',data[2])
```

```
update() When dictionary is Passed: {'Name': 'Sonali', 'Age': 27, 'Sex': 'female'}
update() When dictionary is Passed: {'x': 2, 'y': 3, 'z': 0}
update() when key exists: {1: 'one', 2: 'two'}
Update the key name in dictionary: {'Hello': 56, 'test': 43, 'this': 78, 'where': 23}
first student: {'name': 'Saurabh', 'subjects': ['java', 'python']}
first student after update value using append method: {'name': 'Saurabh', 'subjects': ['java', 'html']}
third student: {'name': 'Ajit', 'subjects': ['iot', 'cloud']}
third student after update value using insert method: {'name': 'Ajit', 'subjects': ['dbms', 'cloud']}
```