
```
# Generators
tup = (num ** 2 for num in range(1, 6))
print ("Generator Object ->", tup)

print ("-----------------------------------------------")

try:
  print ("Value from Tuple Comprehension ->", tup.__next__())
  print ("Value from Tuple Comprehension ->", tup.__next__())
  print ("Value from Tuple Comprehension ->", tup.__next__())
  print ("Value from Tuple Comprehension ->", tup.__next__())
  print ("Value from Tuple Comprehension ->", tup.__next__())
  print ("Value from Tuple Comprehension ->", tup.__next__())
except Exception as e:
  print ("No Values present in the Generator Object")

print ("-----------------------------------------------")

# yield => Creates the generator object, adds the value to the object and
return it to the caller

def crtLoc():
  yield "Bengaluru"
  yield "Chennai"
  yield "Coimbatore"
  yield "Noida"
  yield "Mumbai"

all_loc = crtLoc()
print ("All Locations ->", all_loc)
try:
  while(True):
    print ("Location ->", all_loc.__next__())
except:
  print ("No Locations present in the Generator Object")

print ("-----------------------------------------------")

def yrange(min_num, max_num, incr_number=1):
  count = min_num

  while (count <= max_num):
    yield count
    count += incr_number

all_nos = yrange(2, 10, 2)
try:
  while(True):
    print ("Number ->", all_nos.__next__())
```

```
except:
  print ("No Numbers present in the Generator Object")

print ("-----------------------------------------------")
```