Python Generators

In Python, a generator is a function that returns an iterator that produces a sequence of values when iterated over.

Generators are useful when we want to produce a large sequence of values, but we don't want to store all of them in memory at once.

Create Python Generator

In Python, similar to defining a normal function, we can define a generator function using the `def` keyword, but instead of the `return` statement we use the `yield` statement.

```
def generator_name(arg):
    # statements
    yield something
```

Here, the `yield` keyword is used to produce a value from the generator.

When the generator function is called, it does not execute the function body immediately. Instead, it returns a generator object that can be iterated over to produce the values.

Example: Python Generator

```
def my_generator(n):

    # initialize counter
    value = 0

    # loop until counter is less than n
    while value < n:

        # produce the current value of the counter
        yield value

        # increment the counter
        value += 1

# iterate over the generator object produced by my_generator
for value in my_generator(3):

    # print each value produced by generator
    print(value)
```

```
0
1
2
```

In the above example, the `my_generator()` generator function takes an integer `n` as an argument and produces a sequence of numbers from **0** to `n-1`.

The `yield` keyword is used to produce a value from the generator and pause the generator function's execution until the next value is requested.

The `for` loop iterates over the generator object produced by `my_generator()`, and the print statement prints each value produced by the generator.

We can also create a generator object from the generator function by calling the function like we would any other function as,

```
generator = my_range(3)
print(next(generator))  # 0
print(next(generator))  # 1
print(next(generator))  # 2
```

Python Generator Expression

In Python, a generator expression is a concise way to create a generator object.

Generator Expression Syntax

A generator expression has the following syntax,

```
(expression for item in iterable)
```

Example 2: Python Generator Expression

```
# create the generator object
squares_generator = (i * i for i in range(5))

# iterate over the generator and print the values
for i in squares_generator:
    print(i)
```