Iterating Over Iterables (2)

One of the things you learned about in this chapter is that not all iterables are actual lists.
A couple of examples that we looked at are ranges and the use of the range() function.
In this exercise, we will focus on the range() function.

Code Explanation:

The Python code demonstrates the use of iterators with the range() function.
Here are the steps:
1. Create an iterator object for range(3).
2. Print the next values from the iterator using `next()`.
3. Use a `for` loop to iterate over range(3).
4. Create an iterator object for a large range (10 ** 100), called googol.
5. Print the first 5 values from the googol iterator using `next()`.

Code Implementation:

```
# Create an iterator for range(3): small_value
small_value = iter(range(3))

# Print the values in small_value
print(next(small_value))
print(next(small_value))
print(next(small_value))

# Loop over range(3) and print the values
for num in range(3):
    print(num)

# Create an iterator for range(10 ** 100): googol
googol = iter(range(10 ** 100))

# Print the first 5 values from googol
print(next(googol))
print(next(googol))
print(next(googol))
print(next(googol))
print(next(googol))
```

Execution Output:

```
0
1
2
0
1
2
0
1
2
3
4
```