**Python Slicing Examples**

Here’s a detailed explanation of Python slicing and the `range()` function, along with examples to help you understand how they work.  
  
---  
  
## \*\*Slicing in Python\*\*  
Slicing is used to extract a portion of a sequence (like a list, string, or tuple) by specifying a start, stop, and step.  
  
### \*\*Syntax\*\*  
```python  
sequence[start:stop:step]  
```  
- \*\*start\*\*: The index where the slice starts (inclusive).  
- \*\*stop\*\*: The index where the slice ends (exclusive).  
- \*\*step\*\*: The interval between elements in the slice (default is 1).  
  
### \*\*Examples\*\*  
```python  
# Example 1: Slicing a list  
my\_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print(my\_list[2:7]) # Output: [2, 3, 4, 5, 6]  
print(my\_list[:5]) # Output: [0, 1, 2, 3, 4] (start defaults to 0)  
print(my\_list[5:]) # Output: [5, 6, 7, 8, 9] (stop defaults to end of list)  
print(my\_list[::2]) # Output: [0, 2, 4, 6, 8] (every second element)  
print(my\_list[::-1]) # Output: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0] (reversed list)  
  
# Example 2: Slicing a string  
my\_string = "Hello, World!"  
print(my\_string[7:12]) # Output: 'World'  
print(my\_string[::-1]) # Output: '!dlroW ,olleH'  
  
# Example 3: Negative indices  
print(my\_list[-5:-2]) # Output: [5, 6, 7] (negative indices count from the end)  
```  
  
---  
  
## \*\*Range in Python\*\*  
The `range()` function generates a sequence of numbers. It is commonly used in loops or to create lists.  
  
### \*\*Syntax\*\*  
```python  
range(start, stop, step)  
```  
- \*\*start\*\*: The starting number of the sequence (default is 0).  
- \*\*stop\*\*: The end of the sequence (exclusive).  
- \*\*step\*\*: The difference between each number in the sequence (default is 1).  
  
### \*\*Examples\*\*  
```python  
# Example 1: Basic range  
print(list(range(5))) # Output: [0, 1, 2, 3, 4] (start defaults to 0)  
print(list(range(2, 8))) # Output: [2, 3, 4, 5, 6, 7]  
print(list(range(1, 10, 2))) # Output: [1, 3, 5, 7, 9] (step is 2)  
  
# Example 2: Negative step  
print(list(range(10, 0, -2))) # Output: [10, 8, 6, 4, 2]  
  
# Example 3: Using range in a loop  
for i in range(3):  
 print(i) # Output: 0, 1, 2 (one per line)  
```  
  
---  
  
### \*\*Combining Slicing and Range\*\*  
You can use `range()` to generate indices and then slice a sequence:  
```python  
my\_list = [10, 20, 30, 40, 50, 60, 70, 80, 90]  
indices = range(1, 8, 2) # Indices: [1, 3, 5, 7]  
sliced\_list = [my\_list[i] for i in indices]  
print(sliced\_list) # Output: [20, 40, 60, 80]  
```