Python List Slicing is a technique that allows you to extract a portion (or a subsequence) of a list based on specified indices. It provides a flexible and concise way to manipulate lists by creating new lists that contain a subset of the original elements.

The general syntax for list slicing is:

```
new_list = original_list[start:end:step]
```

where:

- `start` is the index at which the slicing begins (inclusive).
- `end` is the index at which the slicing ends (exclusive).
- `step` is an optional parameter that specifies the increment between indices (default is 1).

Here's an explanation of the different components of list slicing:

1. Start Index: The slicing starts from the element at the `start` index, which is included in the sliced portion. If the `start` index is not specified, it defaults to 0, indicating the beginning of the list.

2. End Index: The slicing ends at the element just before the `end` index. The element at the `end` index is not included in the sliced portion. If the `end` index is not specified, it defaults to the length of the list, indicating the end of the list.

3. Step Size: The `step` parameter determines the increment between the indices. A positive `step` value moves forward in the list, while a negative `step` value moves backward. The default `step` value is 1, meaning adjacent elements are included in the sliced portion. Using a `step` value of 2 would include every other element, for example.

Now, let's see some examples of list slicing in Python:

Program:

```
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

slice1 = numbers[2:6]     # [2, 3, 4, 5]

slice2 = numbers[:5]      # [0, 1, 2, 3, 4]

slice3 = numbers[5:]      # [5, 6, 7, 8, 9]

slice4 = numbers[1:9:2]   # [1, 3, 5, 7]

slice5 = numbers[9:1:-1]  # [9, 8, 7, 6, 5, 4, 3, 2]

print(slice1)
print(slice2)
print(slice3)
print(slice4)
print(slice5)
```

Output:

```
[2, 3, 4, 5]
[0, 1, 2, 3, 4]
[5, 6, 7, 8, 9]
[1, 3, 5, 7]
[9, 8, 7, 6, 5, 4, 3, 2]
```

In the above code, we have a list called `numbers` containing integers from 0 to 9.

We then demonstrate various slicing examples:

- `slice1` extracts elements from index 2 to 5 (`[2, 3, 4, 5]`).

- `slice2` extracts elements from the beginning of the list up to index 4 (`[0, 1, 2, 3, 4]`).

- `slice3` extracts elements from index 5 to the end of the list (`[5, 6, 7, 8, 9]`).

- `slice4` extracts elements from index 1 to 8 with a step of 2 (`[1, 3, 5, 7]`).

- `slice5` extracts elements from index 9 to 1 (in reverse order) with a step of -1 (`[9, 8, 7, 6, 5, 4, 3, 2]`).

These examples demonstrate the versatility of list slicing and how it can be used to extract specific subsequences from a list based on the specified indices and step size.