Indexing

Indexing in Python is the process of accessing a single item or a group of items from a collection, such as a list, tuple or string. In Python, the index of an item in a collection starts from `0`, not `1`. For example, the index of the first item in a list is `0`, the index of the second item in a list is `1`, and so on.

In this tutorial, we will learn how to access items from a list, tuple and string using indexing and slicing. We will also learn how to use negative indexing and slicing. Finally, we will learn how to access a range of items from a collection using slicing.

List Indexing

Lists are ordered collections of items. Each item in a list has an index, which is used to access the item.

```
fruits = ["apple", "banana", "cherry"]
print(fruits[0])
```

List values can be altered by using the index of the item to be changed. For example:

```
fruits = ["apple", "banana", "cherry"]
fruits[0] = "kiwi"

print(fruits)
```

Tuple Indexing

Tuples are ordered collections of items. Each item in a tuple has an index, which is used to access the item.

```
numbers = (1, 2, 3)
print(numbers[2])
```

However, tuples are immutable, which means that they cannot be changed.

```
numbers = (1, 2, 3)
numbers[2] = 4
```

String Indexing

Strings are immutable, so you cannot change the value of a string. However, you can access items from a string using indexing and slicing.

```
word = "hello"
print(word[4])
```

Like tuples, strings are immutable. This means that you cannot change the characters inside a string through indexing.

```
word = "hello"
word[4] = "a"
```

Negative Indexing

Negative indexing is a way to access items from the end of a collection. For example, the index of the last item in a list is `-1`, the index of the second last item in a list is `-2`, and so on.

```
numbers = [1, 2, 3, 4, 5]
print(numbers[-1])
```

Slicing

The syntax for slicing is `start_index:end_index:step`. The `start_index` is the index of the first item in the group and the `end_index` is the index of the last item in the group. The `end_index` is not included in the group. The `step` is the number of items to skip between each item in the group. The default value of `step` is `1`.

```
numbers = [1, 2, 3, 4, 5]
print(numbers[1:3])
```

Exercises

1. Given the following list of fruits: `python fruits = ["apple", "banana", "cherry", "orange", "kiwi"]`

    a. Access the second fruit in the list.

b. Access the last fruit in the list using negative indexing.

c. Access the first three fruits in the list using slicing.

d. Change the value of the last fruit in the list to "grape".

2. Given the following string: `python word = "hello, world"`

   a. Access the first character in the string.

   b. Access the last character in the string using negative indexing.

   c. Access the sub-string "hello" in the string.

3. Given the following tuple: `python numbers = (1, 2, 3, 4, 5)`

   a. Access the first number in the tuple.

   b. Access the last number in the tuple using negative indexing.

   c. Access the first three numbers in the tuple using slicing.

4. Given the following list of numbers: `python numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`

   a. Access every other number in the list using slicing.

   b. Access the last five numbers in the list using slicing.

   c. Access the numbers from the middle of the list to the end using slicing.

   d. Change the value of every even number in the list to the string `"even"`.

5. Given the following string: `python sentence = "The quick brown fox jumps over the lazy dog."`

   a. Access the first 10 characters in the string.

   b. Access the last 10 characters in the string using negative indexing.

   c. Access the sub-string "quick brown" in the string.

Further Reading

Check out the following resources to learn more about indexing and slicing in Python: - W3Schools - GeeksforGeeks - Python Documentation

Use search terms like “python indexing” or “python slicing” to find more resources online.