Intro to python

lecture 3

List and Tuples.

By Mr. Msangi R.I

Python List

- Lists are used to store multiple items in a single variable.
- Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.
- Lists are created using square brackets: Example.

```
FruitsCollections = ["apple", "banana", "cherry"]
print(FruitsCollection)
```

List Items properties

- List items are ordered, changeable, and allow duplicate values.
- `List items are indexed, the first item has index [0], the second item has index [1] etc.`
- Since lists are indexed, lists can have items with the same value

  The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

List

- To determine how many items a list has, use the `len()` function:
  - ```
    thislist = ["apple", "banana", "cherry"]
    print(len(thislist))
    ```
- List items can be of any data type:
  - ```
    list1 = ["apple", "banana", "cherry"]
    list2 = [1, 5, 7, 9, 3]
    list3 = [True, False, False]
    ```
  - ```
    list1 = ["abc", 34, True, 40, "male"]
    ```
- It is also possible to use the `list()` constructor when creating a new list.
  - ```
    thislist = list(("apple", "banana", "cherry")) # note the double round-brackets
    print(thislist)
    ```

Access List Items

- List items are indexed and you can access them by referring to the index number:
- Print the second item of the list:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

- You can specify a range of indexes by specifying where to start and where to end the range.
- When specifying a range, the return value will be a new list with the specified items.

```
thislist =["apple", "banana", "cherry", "orange", "kiwi", "mango"]
print(thislist[2:4])
```

Change Item Value

- To change the value of a specific item, refer to the index number:
- Example if you want to change the second item in the given list,

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
```

Change a Range of Values

- To change the value of items within a specific range, define a list with the new values, and refer to the range of index numbers where you want to insert the new values:

- Example Change the values "banana" and "cherry" with the values "blackcurrant" and "watermelon":

```
thislist =["apple", "banana", "cherry", "orange", "kiwi"]
thislist[1:3] = ["blackcurrant", "watermelon"]
print(thislist)
```

Add List Items

- To add an item to the end of the list, use the append() method:
- Using the append() method to append an item: Example
  - ```
    thislist = ["apple", "banana", "cherry"]
    thislist.append("orange")
    print(thislist)
    ```
- Another method to add into a list is insert().
- The insert() method inserts an item at the specified index:
  - ```
    thislist = ["apple", "banana", "cherry"]
    thislist.insert(1, "orange")
    print(thislist)
    ```

Extend List

- To append elements from another list to the current list, we use the extend() method.

```
thislist = ["apple", "banana", "cherry"]
tropical = ["mango", "pineapple", "papaya"]
thislist.extend(tropical)
print(thislist)
```

- The extend() method does not have to append lists, you can add any iterable object (tuples, sets, dictionaries etc.).

```
thislist = ["apple", "banana", "cherry"]
thistuple = ("kiwi", "orange")
thislist.extend(thistuple)
print(thislist)
```

Remove List Items

- If you want to remove a specific item in the list you use remove() method.
- Example if you want to remove ‘banana’ on the list
  - thislist = ["apple", "banana", "cherry"]
    thislist.remove("banana")
    print(thislist)
- If there are more than one item with the specified value, the remove() method removes the first occurance:

Remove List Items

- The pop() method removes the specified index.

```
thislist = ["apple", "banana", "cherry"]
thislist.pop(1)
print(thislist)
```

- If you do not specify the index, the pop() method removes the last item.
- The del keyword also removes the specified index:

```
del thislist[0]
print(thislist)
```

- The del keyword can also delete the list completely.

```
del thislist
```

- The clear() method empties the list. The list still remains, but it has no content.

```
thislist.clear()
print(thislist)
```

Sort Lists

- List objects have a sort() method that will sort the list alphanumerically, ascending, by default:

```
thislist =["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort()
print(thislist)
```

- To sort descending, use the keyword argument reverse = True:

```
thislist.sort(reverse = True)
print(thislist)
```

- By default the sort() method is case sensitive, resulting in all capital letters being sorted before lower case letters:

Sort Lists

- So if you want a case-insensitive sort function, use str.lower as a key function:

```
thislist = ["banana", "Orange", "Kiwi", "cherry"]
thislist.sort(key = str.lower)
print(thislist)
```

- Also we have The reverse() method,which reverses the current sorting order of the elements.

```
thislist.reverse()
print(thislist)
```

Copy Lists

- You cannot copy a list simply by typing `list2 = list1`, because: `list2` will only be a *reference* to `list1`, and changes made in `list1` will automatically also be made in `list2`.

- There are ways to make a copy, one way is to use the built-in List method `copy()`.

```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()
print(mylist)
```

List concatenate

- There are several ways to join, or concatenate, two or more lists in Python.
- One of the easiest ways are by using the + operator.

```
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]
list3 = list1 + list2
print(list3)
```

- You can also use methods like Append() and Extend() to join two or more list together.

Python Tuples

- Tuples are used to store multiple items in a single variable.
- Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.
- A tuple is a collection which is ordered and **unchangeable**.
- Tuples are written with round brackets.
  - `mytuple = ("apple", "banana", "cherry")`

Tuple With One Item

+To create a tuple with only one item, you have to add a comma after the item, otherwise Python will not recognize it as a tuple.

```
+thistuple = ("apple",)
 print(type(thistuple))

 #NOT a tuple
 thistuple = ("apple")
 print(type(thistuple))
```

- Tuples and lists in Python are similar but with one crucial difference: tuples are immutable, meaning once they're created, you can't modify their contents. This immutability makes tuples useful for situations where you want to ensure that the data remains constant throughout the program execution.
- Just like list You can access tuple items by referring to the index number, inside square brackets:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple[1])
```

Update tuple

- Once a tuple is created, you cannot change its values. Tuples are **unchangeable**, or **immutable** .
- But there is a workaround. You can convert the tuple into a list, change the list, and convert the list back into a tuple.

```
x = ("apple", "banana", “mango")
y = list(x)
y[1] = “orange"
x = tuple(y)
print(x)
```

- Also You are allowed to add tuples to tuples, so if you want to add one item, (or many), create a new tuple with the item(s), and add it to the existing tuple:

```
thistuple = ("apple", "banana", "cherry")
y = ("orange",)
thistuple += y

print(thistuple)
```

Join tuples

- If you want to multiply the content of a tuple a given number of times, you can use the * operator:

```
fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2

print(mytuple)
```

- Also as shown in a previous slide To join two or more tuples you can use the + operator:

Tuple Methods

+Python has two built-in methods that you can use on tuples.

Method	Description
count()	Returns the number of times a specified value occurs in a tuple
index()	Searches the tuple for a specified value and returns the position of where it was found

Q & A