XTECH

PYTHON

Session 5

Speaker. Majid Saqr

TOPICS COVERED

PYTHON BASICS

- Day 1: Introduction to Python and basic syntax, data types, and variables.
- Day 2: Exploring control structures (if statements, loops), functions, and modular programming.

OOP

- Day 3: Introduction to OOP concepts (classes, objects).
- Day 4: Understanding inheritance, polymorphism, and encapsulation.

Speaker. Majid Saqr

TOPICS COVERED (P2)

DATA STRUCTURES AND ALGORITHMS

- Day 5: Introduction to fundamental data structures (lists, tuples, dictionaries).
- Day 6: Delving into algorithms and analysis, focusing on searching algorithms.

APPLICATIONS AND PROJECTS

- Day 7: Engage in a larger-scale programming project incorporating concepts learned throughout the course.
- Day 8: Present and discuss your project, showcasing your understanding and practical application of Python.

WHAT WE WILL TALK ABOUT ?

1) Introduction to fundamental data structures

2) Lists

3) Tuples

4) Dictionaries

Data Structures

A data structure is a specialized format for organizing, processing, retrieving, and storing data.

There are four collection data types in the Python programming language:

Data Structures

List is a collection which is ordered and changeable. Allows duplicate members.

Tuple is a collection which is ordered and unchangeable. Allows duplicate members.

Set is a collection which is unordered, unchangeable, and unindexed. No duplicate members.

Dictionary is a collection which is ordered and changeable. No duplicate members.

Speaker. Majid Saqr

Python Lists

Lists are used to store multiple items in a single variable.

List items can be of any data type:

A list can contain different data types:

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 : thislist = ["apple", "banana", "cherry", 1, True, 2.5]

List Items

When we say that **lists are ordered**, it means that the items have a defined order, and that order will not change.

If you add new items to a list, the new items will be **placed at the end of the list.**

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

Since **lists are indexed**, lists can have items with the same value. ✗

List Length

To determine how many items a list has, use the **len()** function:

Example:

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

Python - Access List Items

List items are indexed and you can access them by referring to the index number:

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon"]
print(thislist[1])
print(thislist[-1])
print(thislist[2:5])
print(thislist[:4])
print(thislist[2:])
print(thislist[-4:-1])
```

Speaker. Majid Saqr

Python - Change List Items

To change the value of a specific item, refer to the index number:

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon"]
thislist[1] = "blackcurrant"
thislist[1:3] = ["blackcurrant", "watermelon"]
thislist[1:2] = ["blackcurrant", "watermelon"]
thislist[1:3] = ["watermelon"]
```

Speaker. Majid Saqr

Insert Items

To insert a new list item, without replacing any of the existing values, we can use the **insert()** method.

The **insert()** method inserts an item at the specified index.

Example:

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(2, "watermelon")
print(thislist)
```

Python - Add List Items append()

To add an item to the end of the list, use the append() method:

Example:

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

Extend List

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

Example: Add the elements of tropical to thislist.

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

Speaker. Majid Saqr

Python - Remove List Items

The **remove()** method removes the specified item.

If there are more than one item with the specified value, the remove() method removes the first occurrence:

Example:

```
thislist = ["apple", "banana", "cherry", "banana", "kiwi"]
thislist.remove("banana")
print(thislist)
```

Remove Specified Index

The pop() method removes the specified index.

If you do not specify the index, the pop() method removes the last item.

The del keyword also removes the specified index:

Example:

```
thislist = ["apple", "banana", "cherry"]
thislist.pop(1)
print(thislist)
thislist.pop()
print(thislist)
del thislist[0]
```

Clear the List

The **clear()** method empties the list.

The list still remains, but it has no content.

Example:

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

Speaker. Majid Saqr

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 ()**.

Python Tuples

Tuple items are ordered, unchangeable, and allow duplicate values.

Tuple items are indexed, the first item has index [0], the second item has index [1] etc.

When we say that **tuples** are ordered, it means that the items have a defined order, and that order will not change.

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

Since **tuples** are indexed, they can have items with the same value:

Speaker. Majid Saqr

Python Tuples

Tuple items can be of any data type:

A tuple with strings, integers and boolean values:

To determine how many items a tuple has, use the len() function:

Example:

```
thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)
```

Speaker. Majid Saqr

Python - Access Tuple Items

You can access tuple items by referring to the index number, inside square brackets:

Example:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple[1])
print(thistuple[-1])
print(thistuple[2:5])
print(thistuple[:4])
print(thistuple[-4:-1])
```

Speaker. Majid Saqr

Python - Update Tuples

Tuples are unchangeable, meaning that you cannot change, add, or remove items once the tuple is created. But there are some workarounds. You can convert the tuple into a list, change the list, and convert the list back into a tuple.

Example:

```
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "kiwi"
x = tuple(y)
```

Python - Loop Tuples - list

You can loop through the tuple items by using a for loop.

Example Tuple:

```
thistuple = ("apple", "banana", "cherry")
for x in thistuple:
  print(x)
```

Example List:

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

CHECKING FOR UNDERSTANDING

At the end of the training course, check the audience's understanding of the topics discussed.

Speaker. Majid Saqr

THANK YOU FOR LISTENING!

Reach out for any questions.

WEBSITE

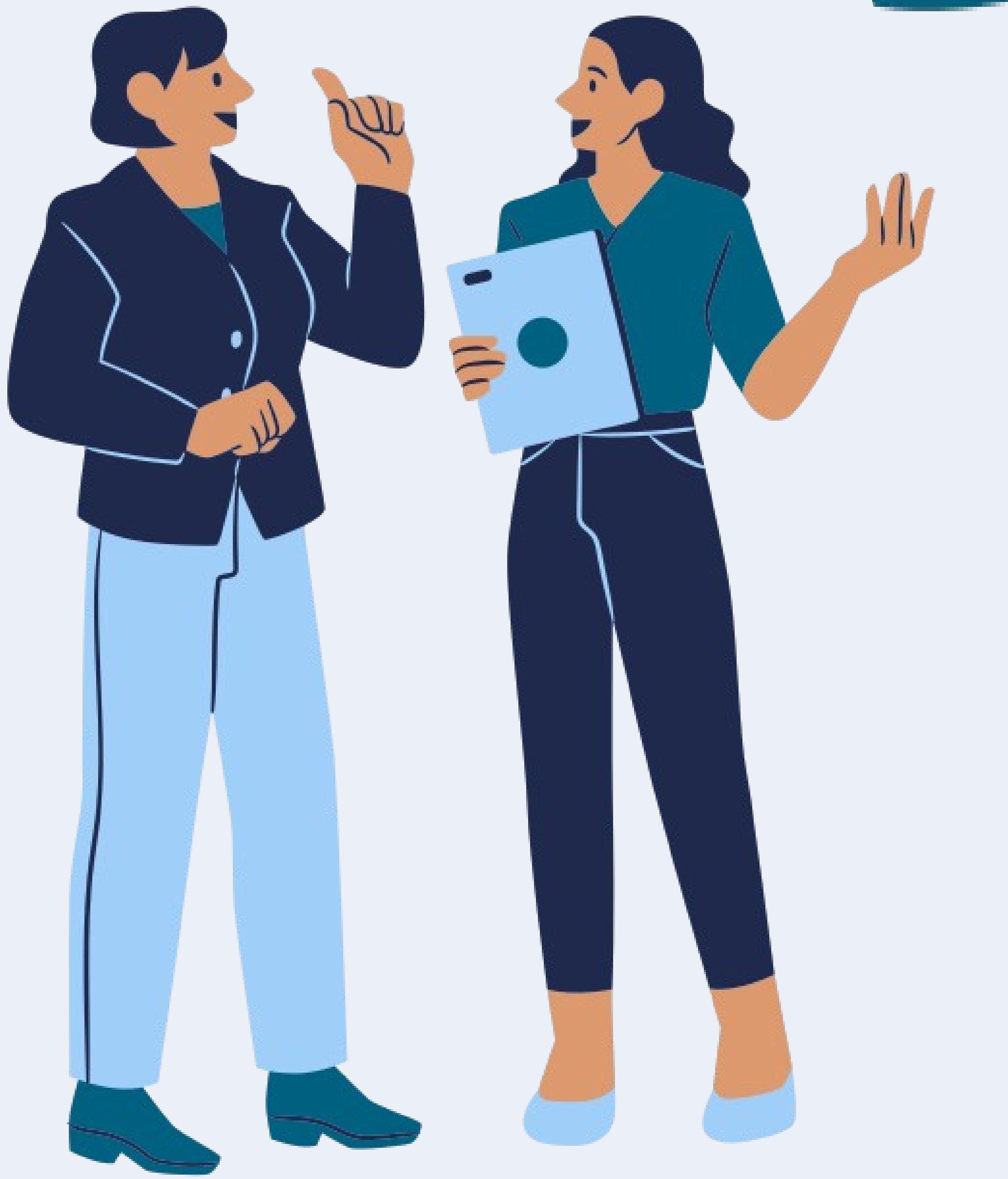
expotech.online

EMAIL ADDRESS

contact@expotech.online

PHONE

01144253820



Speaker. Majid Saqr