Intor to Python Lesson 3: Data Structures

List

- A collection of objects
 - Object can have different types
- O Items are accessible through their index
 - O Index starts from 0
- O Items are seperated by comma ","
- O Negative index: -1, -2, ...
- Slice: [startIndex : stopIndex]
 - o myList[1:3]
 - o myList[:3]
 - o myList[1:]
- Check membership with "in"/"not in"
 - 1 in myList
 - ,Alice" not in myList

```
myList = [1, 2.0, "Ali", True]
print(myList[0]) #list index starts from 0 and ends at N-1 (N: number of elements)
print(myList[1])
print(myList[2])
print(myList[3]) #generates an Error!
```

Mutability & Order

- Lists & Strings are both ordered becuased their eleemnts can be accessed by their position
- o myList[0]
- o myString= "Ali"
 - \circ myString[0] \rightarrow "A"

- Lists are mutable
 - myList[0] = ",one"
- O Strings are immutable
 - \circ myString[0] = "a" \rightarrow Error

List Methods

- len(myList) → Number of elements
- o max(), min()
- o sorted()

- myList.append(): add to end
- myList.insert(pos, element)
- + or extend()
- remove(element)
 - What happens with duplicate elements?

XGram 1.0

- You are going to develop the next generation of Instagram called "XGram,...
- For this challenge write 5 lists
 - Users: Alice, Bob
 - postsAlice :Wow! Iam finally in Karlsruhe!,
 Great music @ AKK
 - opostsBob: Great time at python class!
 - imgsAlice: shorturl.at/chDUX, shorturl.at/aorP1
 - o imgsBob: shorturl.at/wxFLR

- Add a new message to postBob
 - I just learned about lists!
- add oldPostsAlice to the beginning of postsAlice
 - oldPostsAlice=[,,Got the ticket", ,,On my way to Karlsruhe"]
- Print last two posts of Alice

Tuples

- Immutable sequance of elements
- Packing & Unpacking data

```
location = (13.4125, 103.866667)
print("Latitude:", location[0])
print("Longitude:", location[1])
```

```
dimensions = 52, 40, 100
length, width, height = dimensions
print("The dimensions are {} x {} x {}".format(length, width, height))
```

Sets

- Mutable unordered collection of data
- Items are unique
- Create a grocery list out of
 - o set([,,milk", ,,apple", ,,orange", ,,milk"])
- o set([1,2,3]) set([2,3])

- o intersection()
- o issubset()
- o union()

Dictionaries

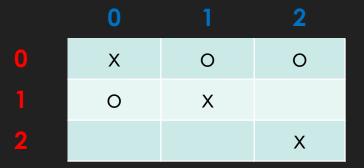
- A dictionary is a mutable data type that stores mappings of unique keys to values
- myDict = {key1:value1, key2:value2, ...}

```
myDict = {"Ali":"0176xxxx", "Farshid":"0164xxx"}
myDict["Ali"] # "0176xxxx"
myDict["Ali"] = "0175xxx"

phoneNumber = myDict["Shahin"]
phoneNumber = myDict.get("Shahin", "Is not on your contact list")
```

Compound Data Structures

```
doozTable = [['x', 'o', 'o'],['o', 'x', ' '],[' ', ' ', 'x']]
print(doozTable[0])
print(doozTable[1])
print(doozTable[2])
print(doozTable[1][0])
```



Dooz 1.0

- Implement dooz for only a single round
- Use ,x' for user1 and ,o' for user2
- User 1 starts the game and enters the row and then enters the column
- User 2 does the same
- Print the dooz table
 - Empty cells with " "

- Note the type of data returend by input()!
- Note that index starts with zero
- Initial the dooztable with single space stings

XGram 1.1

- Create a compound data structure that stores all data from the 1st version
- Use tuples to pack image urls with the relevant posts
 - Check out the urls to find the relevant posts!
 - For posts that do not have an image set the url to empty string

 Read the name of the user from the input and print the last post including the url (if available)