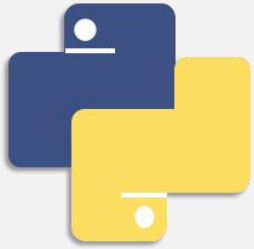


Python Programming - VI

List Operations

Contents

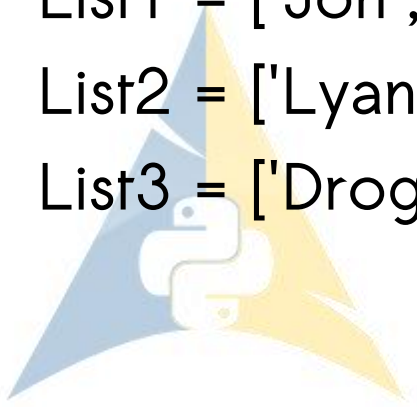
- Basics
- Accessing values
- Updating values
- Deleting values
- Operations
- Indexing, Slicing, Matrices
- Methods

Basics

- Written as a list of comma-separated values (items) between square brackets
- Creating a list is as simple as putting different comma-separated values between square brackets

Examples:

```
List1 = ['Jon', 'Arya', 'Sansa', 'Bran', 'Rickon', 'Rob']
List2 = ['Lyanna', 'Tommen', 'Joffrey']
List3 = ['Drogon', 'Viserion', 'Rhaegal']
```



Accessing Values in List

- Indices can be used for accessing a single or range of values
- Using the list with print as it will, print the entire list

Example:

```
print List3
print List2[0]
print List1[:3]
```

Updating values in List

- You can update single or multiple elements of lists by giving the slice on the left-hand side of the assignment operator, and you can add to elements in a list with the append() method

Ex:

```
List1 = ['Targaryen', 'Lannister', 'Starks']

List1[1] = 'Walkers'  # Assigns new value
```

Deleting Values in List

- To remove a list element, you can use either the del statement if you know exactly which element(s) you are deleting or the remove() method if you do not know.

Example:

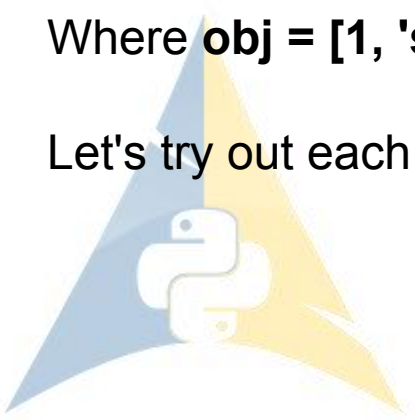
```
List1 = ['Targaryen', 'Lannister', 'Starks']

del List1[1]
```

Operations

function	Operation
len(obj**)**	Finds Length
obj **+** obj	Concatenation
obj ***** no_of_times	Repetition
value **in** obj	Membership (Boolean)
for x **in** obj: print x	Iteration

Where **obj = [1, 'some', 'text']**

Let's try out each



Indexing, Slicing, Matrices

- Because lists are sequences, indexing and slicing work the same way for lists as they do for strings.

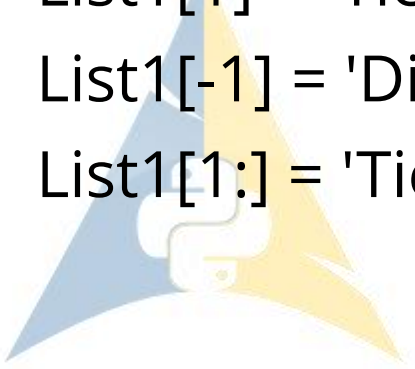
Example:

List1 = ['Win', 'Tie', 'Lose', 'Die']

List1[1] = 'Tie'

List1[-1] = 'Die'

List1[1:] = 'Tie', 'Lose', 'Die'

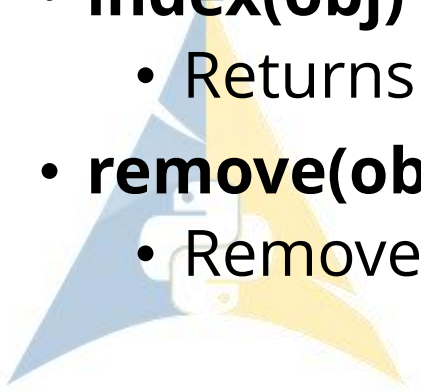


Methods (Built-in)

- **cmp(list1, list2)**
  - Compares both lists
- **len(list)**
  - Gives total length of lists
- **max(list)**
  - Returns the max value from list
- **min(list)**
  - Returns the min value from list
- **list(seq)**
  - Converts a tuple into list

Methods (Built-in)

- **append(obj)**
  - Appends the passed object into Data structure
- **count(obj)**
  - Returns count of occurence of passed object
- **extend(seq)**
  - Appends the contents of seq to list
- **index(obj)**
  - Returns the lowest index in list that obj appears
- **remove(obj)**
  - Removes the passed object from the list



Methods (Built-in)

- **insert(index, obj)**
  - Insert object obj into list at offset index
- **pop(obj_index)**
  - Pops out the object value from sepcified index
- **reverse()**
  - Reverses the values of list
- **sort()**
  - Sorts the list in Ascending order of ASCII

FIN