Programming Club Meeting 7 Slides



Basics

- Set of values stored in a single variable
 - String is list of chars
- Can have different types in Python (not true in most other languages
- Ordered and changeable, can have duplicates
- Very versatile and multi-purpose in Python
- AKA arrays and slices
- Can loop through each item in a list
 - How for loops work
- Can put lists in lists (2d arrays)

```
Code
 # Basic List
  lst = [1, 2, 3, 4, 2]
  print(lst)
  multi = [1, "hello", 3.8, True]
  print(multi)
  blankList1 = []
  blankList2 = list()
                                      3
                                      4
  print()
                                      2
  # Loop Through List
  for num in 1st:
      print(num)
  print()
  # 2d Array
18 array = [[1, 2], [2, 3], [3, 4]]
```

Output

```
[1, 2, 3, 4, 2]
[1, 'hello', 3.8, True]

1
2
3
4
```

Commands

Indexing notes:

- Like strings
- Starts at 0
- Can be negative
- If you make something from a range, it's called a slice

```
Code
                                   Output
 1 lst = [1, 2, 3, 4, 2]
                                   [1, 7, 3, 4, 2]
  # Indexing
                                   [1, 7, 9, 10]
  print(lst[1]) # Starts at 0
  lst[1] = 7
 6 print(lst)
                                   Length: 4
  lst[2:5] = [9, 10]
                                   True
 8 print(lst)
                                   False
10 print()
                                   10
  # Other Commands
13 print(f"Length: {len(lst)}")
14 print(10 in lst)
15 print(-1 in lst)
16 del blankList1
  print(min(lst))
18 print(max(lst))
```

List Methods (List of List Methods)

```
# Methods
ccounts = [2, 5.3, -89, 200]
accounts.append(9)
print(f"Append 9: {accounts}")
ccounts.insert(1, 9)
print(f"Insert 1,9: {accounts}")
print(f"Index 9: {accounts.index(9)}")
print(f"Pop 2 return: {accounts.pop(2)}")
print(f"Pop 2: {accounts.count(9)}")
print(f"Count 9: {accounts.count(9)}")
```

Output

```
Append 9: [2, 5.3, -89, 200, 9]
Insert 1,9: [2, 9, 5.3, -89, 200, 9]
Index 9: 1
Pop 2 return: 5.3
Pop 2: [2, 9, -89, 200, 9]
Count 9: 2
```

List Methods Cont.

```
Code
                                              Output
 1 # Methods
                                              Accounts2: [7, 9, -89, 200, 9]
 2 | accounts = [2, 9, -89, 200, 9]
 3 if (True): # Copy by reference
                                              Accounts2 Copied: [7, 9, -89, 200, 9]
      accounts2 = accounts
      accounts[0] = 7
      print(f"Accounts2: {accounts2}")
                                              Sorted: [-89, 9, 9, 93, 200]
 7 accounts2 = accounts.copy()
                                              Extend: [-89, 9, 9, 93, 200, 7, 9, -89, 200, 9]
8 print()
                                              Reverse: [9, 200, -89, 9, 7, 200, 93, 9, 9, -89]
9 | accounts[0] = 93
                                              Clear: []
10 print(f"Accounts2 Copied: {accounts2}")
11 print()
12 accounts.sort()
13 print(f"Sorted: {accounts}")
14 accounts.extend(accounts2)
15 print(f"Extend: {accounts}")
16 accounts.reverse()
17 print(f"Reverse: {accounts}")
18 accounts.clear()
19 print(f"Clear: {accounts}")
```

Quick Note

- Most languages have no or next to no methods
- These methods are very helpful and quite optimized
- Use them instead of writing your own

Tuples

- Same as list but can't change anything
- Great way to return values
- Unpack returned values

```
Code
                                 Output
  # Tuple
                                 (3, 5)
  t = (3,5)
                                 x: 8, y: 1
  print(t)
  # t[0] = 8 # causes error
5
  # Returning w/ Tuple
  def getLocation():
8
       return (8,1)
  x, y = getLocation()
10 print(f"x: {x}, y: {y}")
```

Line 4: TypeError: 'tuple' does not support item assignment

Practice

Example Problem: Remove All

- Src:
 - https://pynative.com/python-list-exercise-with-solutions/#h-exercise-10-remove-all-occurrence s-of-a-specific-item-from-a-list
- Goal: Write a Python program that will remove all occurrences of a number from a list.
- Relevant Information:
 - Ex:
 - list1 = [5, 20, 15, 20, 25, 50, 20]
 - To be removed: 20
 - Output: [5, 15, 25, 50]
 - Try with and without 'remove' method

Practice Problem 1: Square Items

- Src:
 - https://pynative.com/python-list-exercise-with-solutions/#h-exercise-3-turn-every-item-of-a-list-into-its-square
- Goal: Write a Python program that will square all of the items in a list.
- Ex:
 - o numbers = [1, 2, 3, 4, 5, 6, 7]
 - Output: [1, 4, 9, 16, 25, 36, 49]

Practice Problem 2: Swap Values

- Goal: Write a Python program that will swap the positions of 2 different values in a list.
- Relevant Information:
 - o Ex:
 - numbers = [1, 2, 3, 4, 5, 6, 7]
 - Swap: 3, 5
 - Output: [1, 2, 5, 4, 3, 6, 7]
 - If there are multiple instances of either value, only look at the first instance
 - If values 1 and 2 are the same, don't do anything

Practice Problem 3: Sum and Average

- Goal: Write a Python program that will find the sum of the numbers in a list as well as their average.
- Relevant Information:
 - Ex:
 - numbers = [-1, 0, 1, 2, 3, 4, 5]
 - Output: Sum 14, Average: 2
 - You can assume that there are only valid numbers in the list and that the length is ≥1

Sum Command

```
1 numbers = [-1, 0, 1, 2, 3, 4, 5]
2 print(sum(numbers))
1 14
```

Next Meeting: List Review + Practice

