Programing in Python

Lecture 6 - Lists

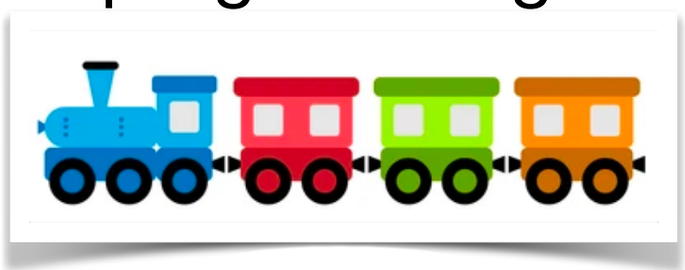
Instructor: Zhandos Yessenbayev

Outline

- Python Collections
- Python Lists
- Indexing and Slicing
- List Methods
- Looping through Lists

Python Collections

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



  - **List** is a collection which is *ordered* and *changeable* (with *duplicates*).

  - **Tuple** is a collection which is *ordered* and *unchangeable* (with *duplicates*).

  - **Set** is a collection which is *unordered*, *unchangeable* (with *no duplicates*).

  - **Dictionary** is a collection which is *ordered* and *changeable* (with *no duplicates*).

Python List

- **List** is a collection which is *ordered* and *changeable*
  (with *duplicates*).

- We can initialize a list with some items

- Items (elements) can be of any type (even another lists)

- *Empty* list can be set with **[ ]** or **list( )**

- *Length* of the list can be found using **len( )** function

```
>>> list1 = [25, 'hello', 'hello', [100, 200] ]
>>> len(list1)
4
>>> list2 = []
>>> len(list2)
0
>>> list3 = list()
>>> list3
[]
```

Indexing and Slicing

```
>>> mylist = ['C', 'H', 'Tom', 10]
>>> len(mylist)
4
>>> mylist[0]
'C'
>>> mylist[1]
'H'
>>> mylist[2]
'Tom'
>>> mylist[-2]
'Tom'
>>> mylist[-1]
10
>>> mylist[0] = 'M'
>>> mylist
['M', 'H', 'Tom', 10]
```

len( ) — returns list's length

Access items by **index**

We can **change** the items

Indexing and Slicing

- To take a **range** of items, we use list **slicing**.
- Syntax: **<list_name>** [start_index : end_index : step]

```
>>> mylist = [0, 1, 2, 3, 4, 5, 6]
>>>
>>> mylist[1:5]
>>>
>>> mylist[3:]
>>>
>>> mylist[:-1]
>>>
>>> mylist[2:-2]
>>>
>>> mylist[1::2]
>>>
>>> mylist[::-1]
```

Find the resulting lists

List Methods

- Python has built-in methods to use on lists.

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Source: https://www.w3schools.com/python/python_lists_methods.asp

List Methods

```
>>> list1 = [0, 1, 2, 3, 4, 5, 6]
>>> dir(list1)
>>>
>>> list1.append(6)
>>>
>>> list2 = list1.copy()
>>>
>>> list3 = list1
>>> list1.clear()
>>>
>>> list2.count(6)
>>>
>>> list2.reverse()
>>> list2.index(6)
>>>
>>> list2.sort()
>>> list2.index(6)
```

What are list1, list2 and list3 after this?

List Methods

```
>>> list1 = [0, 1, 2, 3]
>>> list2 = [4, 5, 6]
>>>
>>> list2.append(7)
>>>
>>> list3 = list1 + list2
>>> list1.extend(list2)
>>>
>>> list1.insert(0, -1)
>>> list1.insert(1, 'ok')
>>>
>>> item = list1.pop(1)
>>> item = list1.pop()
>>>
>>> list1.remove('ok')
>>>
>>> del list1[0]
```

list.append(elmnt)

list.extend(iterable)

list.insert(pos, elmnt)

val = *list*.pop(pos)

list.remove(elmnt)

Looping through Lists

- We use **for** to loop through the list.
- There are 2 use cases when you need:
  - to read the elements
  - to change the elements

```
>>> list1 = [0, 1, 2, 3, 4, 5, 6]
>>>
>>> # Case 1
>>> for item in list1:
...     print(item)
...
>>>
>>> # Case 2
>>> n = len(list1)
>>> for i in range(n):
...     list1[i] = list1[i] * 2
...
```

Remember **range( )** function!

Unique Words

- *Exercise:* **Find all unique words in a file.**

    1. Create a new file *input.txt*
    2. Fill the file with some text
    3. Open the file from a program
    4. Read the file line by line
    5. Find the words in each line
    6. Populate a list of unique words
    7. Print the list of unique words in alphabetic order

Thanks!