Lists, Tuples, NamedTuples

Introduction to Computer Science!

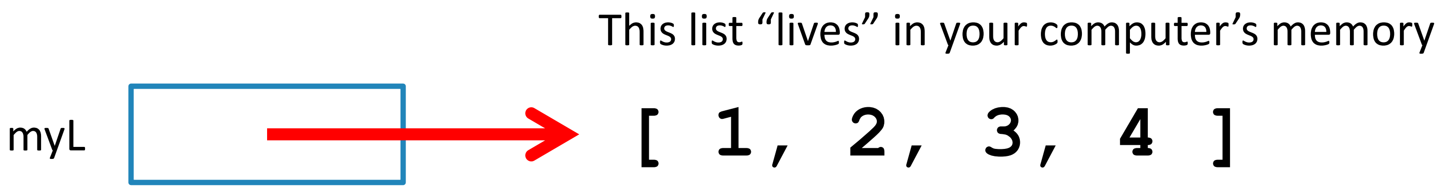
Storing sequences

- Strings – stores sequence of characters
- Lists – stores sequence of any type (including mixed types)
- Tuples – Similar to lists with the difference that they cannot be modified
- NamedTuples – Like tuples but more convenient way of indexing

Lists: Ordered collection of multiple values

- Lists are ordered
- List elements can be accessed by index
- Lists can contain (different) types of objects
- Lists can have duplicate values
- Lists can be nested
- List elements can be modified (mutable)
- Lists are dynamic (they can grow and shrink)

Lists are Mutable, Strings are Not



```
>>> myL = [1, 2, 3, 4]# same as myL = list(range(1,5))
>>> myL[3] = 42   # Indexing MUTATES the list!
                  # It changes the list in place


>>> myS =“Apple”
>>> myS[3]= ‘z’
#Error!
```

Concep Test

```
fruits = ["apple", "banana", "orange"]
fruits[1] = "pear”
print(len(fruits))
```

What is the output of this code?

A. 1

B. 3

C. 4

D. None of the above

List methods

```
Type dir(list) to get all the
methods:
[…'append', 'clear', 'copy', 'count',
'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']

>> help(list.sort)
Help on method_descriptor:

sort(...)
    L.sort(key=None, reverse=False) -> None
-- stable sort *IN PLACE*
```

Concep Test

```
fruits = ["apple", "banana", "orange"]
fruits[-1] = 3
print(fruits.count(3))
```

What is the output of this code?

A. 1

B. 3

C. 4

D. Error

Tuples

- Similar to lists: store a sequence of elements

lst = [10, 20] //ex of a list

tup = (10, 20) //ex of a tuple

- Elements are ordered an can be accessed using the appropriate index

tup[0]

tup[1]

- Different from lists in the following ways
  - Can’t change an element in the tuple
  - Can’t sort the elements in a tuple

Creating empty lists and tuples

- Different ways to create an empty list:

```
lst = []
lst = list()
```

- Different ways to create an empty tuple:

```
tup = ()
tup = tuple()
```

Creating tuples with one element

- Create a tuple with one integer element 10

```
tup = (10) # Incorrect, we’ll discuss why

tup = (10,) # This is correct
```

Concep Test

```
fruits = ("apple", "banana", "orange")
fruits[-1] = 3
print(fruits.count(3))
```

What is the output of this code?

A. 1

B. 3

C. 4

D. Error

Named Tuples

- Used to package data with multiple attributes: e.g. representing a student in your program
- A student’s attributes may be: name, perm number, major etc.
- Named tuples make it easier to access each attribute

Named Tuples

```
from collections import namedtuple

#Design your named tuple object
Student = namedtuple(‘Student’, ‘name perm
major gpa’)

# Create objects of type Student
s1 = Student(“Jack”, 123443, CS, 3.8)
s2 = Student(“Mary”, 8932737, CE, 3.9)

# Access the elements of the objects
print(s1.name, s1.perm)
```

Coding problem

- Write a function **swap** that takes three inputs:

  1) A list: lst

  2) Index1: i1

  3) Index2 : i2

The function should swap the elements at index i1 and i2