CSCI 1913

01/23/17

what we learn?

- Lists - mutable objects
- How does Python work internally? – pointers
- Dictionaries
- How to use Dictionaries to simulate to switch statement?

Lists

- Lists is mutable version of a tuple.
  - [e0, e1, e2, …en] : list of n elements.
  - [e] : list of 1 element
  - [] : list of 0 elements
- we can do anything to a list that we can with a tuple.

  For example, t =[‘a’,’b’,’c’], t[1] => ‘b’, len(t)=>3 ‘a’ in t => true , for E in t: print (E)

- Change a list (when t =[‘a’,’b’,’x’])
  - t[2] = ‘x’
  - print (t) # [‘a’,’b’,’x’]
- Delete a list
  - Del t[2]
  - Print (t) # [‘a’,’b’]

Another example l=[‘a’,’b’,’c’,’d’] (0,1,2,3)

- del l[1]
- print (l) # [‘a’,’c’,’d’] (0,1,2)

How does Python work internally? – pointers

Python variables are actually pointers.

Pointer = memory address

- pointers are small.
- Assume that pointers are all the same size.
- Just moves pointers

L=[‘a’,’b’,’c’]

L looks like this:

List	string	‘a’ (index = 0)	String	‘b’(index = 1)	string	‘c’(index=2)

Pointer aliasing (; I’ve got two names): two different variables are pointing one list.

M = L

M = [1] = ‘x’

Print (L) # [‘a’,’x’,’c’]

- Pointers never need to worry about this with immutable objects.

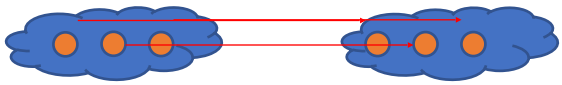
Empty list

L=[]

Make a list and initialize the same list

- K*[e]=[e,e,e,…e(k)]

Dictionary



Keys (object; cloud)

Values

Immutable any objects

Maybe different type different type

{K0:V0,K1:V1,…,Kn-1:Vn-1}

{K:V} 1 pair

{} 0 pair

01/26/17

What do we learn?

- Dictionaries
  - How to use a dictioanries to simulate and switch
- OOP in Python

Dictionaries?

- Let's say there is a language
  D={'zilcho'=0,'two'=1,'duo'=2}
  D['zilcho']=0
  D['duo']=2