cs224d-1

May 11, 2016

In [ ]:
```
x = 3
print x **3
print ' %s %s %d' % ('hello','hi', 12)
s = "hello"
print s.rjust(7) #7 char padding to the right
print s.center(7)
print s.replace('l','leef')
print s.strip()  #strip takes away all the white spaces.
x = [1,2,3]
print x[-1]
print x [-2]
x.pop()  #when you pop an element, it always pops it from the end of the list.
print x

x = range(5)  # [0, 1, 2, 3, 4]
print x[1:3]
print x[2:]  #means from 2 to the end [2, 3, 4]
print x[:2]  # means from the begining till 2  [0, 1]
print x[:]  #means print them all
print x[2:4]  #print the element 2 and 3 - don't print elemnt 4.
print x[:-1]  #print everything except the last element print x[:-1]  [0, 1, 2, 3]
#quick note: so the right side specifies the element that is NOT in the print but that is where
x[:-1] = [-4]   # replace everything before the last elemetn by -4. even though number of eleme
#were larger than one, but still all of them got replaced by -4. so when i printed x again, it
print x  #[-4, 4]

x = ['cat', 'dog', 'moneky']
for index, animal in enumerate(x):
    print '#%d  %s' % (index, animal)

y = [element +'this' for element in x]
print y

d = {'cat': 'cute', 'dog': 'furry'}

print d.get('monkely','NA')
#del d['fish']  #you get an error because there is no fish in there

d = {'person': 2, 'cat': 4, 'spider': 8}
for animal in d:
    #for each key, get the value and then print the set of key and value
    legs = d[animal]
    print (animal,legs)
```

```
#you can write things right away into a dictionary if you want to

lst = range(5)

#look how  x:x **2 created a dictionary element. if it were to be tuple, it should have been li
dic_num_to_square = { x: x **2 for x in lst if x%2 == 0}
print dic_num_to_square


#sets : sets are srounded by {} but the list is srounded by []
animals = {'cat', 'dog'}
print 'cat' in animals  #inf an element exist in a set you will get atrue
#now lets see if the same thing holds for a list ... yeah i guess it holds for both
animalss  = ['cat', 'dog']
print 'cat' in animalss
# if you animalss.add('fish') you will get an error because it is list
# but add totally works for a set. also the remove is only for a set not a list.
animals.add('fish')
print animals  #set(['fish', 'dog', 'cat'])
print len(animals)
print len(animalss)  # the len works for both a list and a set
animals.remove('cat')
print animals  #set(['fish', 'dog'])

# apparentlty the enumerate works for both a list and a set. so you can use it for both
#first let's add another elemetn to the set and then enumerate over all its elements
animals.add('monkey')
for indx, element in enumerate(animals):
    print(indx, element)
#(0, 'fish')
#(1, 'dog')
#(2, 'monkey')

#just like dict and lists, for sets we can also compose them. The set also gets {} but not two
from math import sqrt

num = { int(sqrt(x))  for x in range(30)}
print num  #you can see that numb is a set. the int is there to round the sqrt values otherwise
# set([0, 1, 2, 3, 4, 5])


# Tuples:
#the major differne between tuple and list is that tuple can be used as a key in a dic but not
#here is an example of a tuple being used as a key of a dictionary
d = {(x, x**2): x  for x in range(10)}
print d
#{(6, 36): 6, (0, 0): 0, (7, 49): 7, (4, 16): 4, (5, 25): 5, (3, 9): 3, (9, 81): 9, (2, 4): 2,
#look i have a tuple as the key value
print d[(6,36)]

def sign(x):
    if x > 0:
        return 'pos'
```

```
    elif x < 0:
        return 'neg'
    else:
        return 'zero'
for x in [-1,0,1]:
    print  sign(x),
#so coool, if i just place ',' at the end of a line, i can print everything on the same line

def hello(name, loud = False):
    if loud:
        print "Hello, %s!" %name.upper()
    else:
        print 'hello, %s' %name

hello('bob')
hello('bob', loud = True)


#classes
# name of the class and the object or objects that it receives

class Greeter(object):
    def __init__(self,name):
        self.name = name

    def greet(self):
        print 'hello %s' % self.name

g = Greeter('fred')
g.greet()
```