

Discussion April 13: Python Review

Basics

Commenting and Printing

```
#A comment is preceded by the # symbol

#Block commenting: Can use triple quotes
"""
This is a haiku
That interpreter won't run
Because triple quotes
"""

#Print function
print("This is something that will be printed!")

#Printing multiple variables
a = 5; b = 6; c = "wombat"
print(a,b,c) #5 6 wombat

#Printing variables together with strings
a = 42
print("The answer to the universe is: " + str(42))
```

If/Else Statements

```
#Typical flow sequence
a = 5
if a == 0: sign = 'zero'
elif a > 0: sign = 'positive'
else: sign = 'negative'
print("The number is " + sign)

#With boolean operators
happy = True; rainy = False
if happy and not rainy: print("Time to go out!")

#If/else in one line
```

```
b = 3
c = "even" if b % 2 == 0 else "odd"
print(c) #odd
```

For Loops

```
#Print the numbers 0 through 4 line-by-line
n = 5
for i in range(n):
    print(str(i))
```

```
#Print every string in the list called basket
basket = ['apple', 'banana', 'orange']
for fruit in basket:
    print(fruit)
```

Functions

```
#Define a simple function and use it.
def addNumbers(a,b):
    return a+b
result = addNumbers(4,3)
print(result) #7
```

Data Structures

Lists

```
#Define a list
oddnums = [1,3,5,7,9]
```

```
#Get length of list
length = len(oddnums)
print(length) #5
```

```
#Get an element from the list
e = oddnums[2]
print(e) #Result is 5
```

```

#Get subsets of the list
sub1 = oddnums[1:4]; print(sub1) #[3,5,7]
sub2 = oddnums[2:]; print(sub2) #[5,7,9]
sub3 = oddnums[:2]; print(sub3) #[1,3]
sub4 = oddnums[:-1]; print(sub4) #[1,3,5,7]

#Append an element to the list
oddnums.append(11)
print(oddnums) #[1,3,5,7,9,11]

#Merge two lists together
oddnums.extend([13,15])
print(oddnums) #[1,3,5,7,9,11,13,15]

#Check if an item is in the list
if 6 in oddnums: print("yes")
else: print("no") #Will print no since 6 isn't in the list

```

Dictionaries

```

#Create a dictionary
food = {
    "sushi": "Japanese",
    "pizza": "Italian",
    "burrito": "Mexican",
    "ramen": "Japanese",
    "lasagna": "Italian",
    "bulgogi": "Korean"
}

#Add a new key to the dictionary
food["bibimbap"] = "Korean"

#Iterate through all keys and values
for key,value in food.items():
    print(key + " is " + value + " food")

#Search for a value by its key
type = food["ramen"]
print(type) #Japanese

```

```

#Check if a key is in the dictionary
if "burrito" in food: yesBurrito = True
else: yesBurrito = False
print(yesBurrito) #True

#Check how many key/values pairs in the dictionary
length = len(food)
print(length) #7

```

Strings

```

#Replace one character with another
oldstring = "Life is good?"
newstring = oldstring.replace('?', '!')
print(newstring) #Life is good!

#Remove a character from a string
oldstring = "My name is John-Jacob"
newstring = oldstring.replace("-", "")
print(newstring) #My name is JohnJacob

#Split a string by a specific character
string = "Grocery Outlet Bargain Market"
new = string.split(" ")
print(new) #['Grocery', 'Outlet', 'Bargain', 'Market']

```

Reading in Files

```

#Read text file by line into a list
filepath = "C:/Users/jstwa/Desktop/tags.txt"
with open(filepath, 'r') as f:
    data = f.readlines()
data = [x.rstrip() for x in data]

#Read csv file by line into a list
import csv
filepath = "C:/Users/jstwa/Desktop/tags.csv"
f = open(filepath, 'r')
reader = csv.reader(f)
data = []

```

```

for row in reader:
    data.append(row)

#Store each column into an individual list
ids = []; names = []
for item in data:
    curr_id,curr_name = item.split(' ',1) #Split only on 1st occurrence
    ids.append(int(curr_id)); names.append(curr_name)

```

Numpy

```

#You first need to import the package numpy
import numpy as np

```

```

#Convert a list into a numpy array
mylist = [1,3,5,7,9]
myarray = np.array(mylist)
print(myarray) #[1 3 5 7 9]

```

```

#Convert a list of lists into a 2D numpy array
my2Dlist = [[1,3],[2,4],[7,8]]
my2Darray = np.array(my2Dlist)
print(my2Darray)

```

```

[[1 3]
 [2 4]
 [7 8]]

```

```

#Various simple operations on 2D numpy arrays
n,p = np.shape(my2Darray) #Get the dimensions of 2D array
my1Darray = np.ravel(my2Darray) #Unravel 2D array into 1D array
myrsharray = np.reshape(my2Darray,(2,3)) #Reshape the array into 2 x 3
mytsparray = my2Darray.T #Transpose the array

```

```

#Standardize 2D array
mystdzarray = (my2Darray - np.mean(my2Darray))/np.std(my2Darray)

```

```

#Matrix multiply 2D numpy arrays
newarray = np.dot(my2Darray,mytsparray)

```

```

#Defining a new array

```

```

zeros = np.zeros((3,3)) #3x 3 matrix of all zeros
ones = np.ones((5,4)) #5 x 4 matrix of all ones
sevens = np.repeat(7,6) #1 x 6 matrix (1D array) of all sevens
iden = np.diag(np.repeat(1,5)) #5 x 5 identity matrix
unif = np.random.rand(3,3) #3 x 3 matrix of random values over Unif(0,1)

#Random sampling without replacement
np.seed(1234) #So the results stay fixed
data = [1,3,5,7,9,11,13,15,17,19,21]
sample = np.random.choice(data,size=5,replace=False)
print(sample) #[17  5 19 21  3]

#Apply a function over the columns of 2D array
#For rows, change axis argument to 1
my2Darray = np.array([[1,3],[2,4],[7,8]])
col_sums = np.apply_along_axis(np.sum,axis=0,arr=my2Darray)
print(col_sums) # [10 15]

```

Pickle

```

#Import the pickle package first
import pickle

#Save any Python object to a file to be used for later
savelist = [1,3,5,7,9]
pickle.dump(savelist,open("file.p",'wb'))

#Load a pickled object from file
loadedsavelist = pickle.load(open("file.p",'r'))
print(loadedsavelist) #[1, 3, 5, 7, 9]

```