CS 133 - Introduction to Computational and Data Science

Instructor: Renzhi Cao
Computer Science Department
Pacific Lutheran University
Spring 2017

Announcement

- Quiz #2
- Form group for projects
- Solutions for eval practice

The built-in function `eval` takes a string and evaluates it using the Python interpreter. For example:

```
>>> eval('1 + 2 * 3')
7
>>> import math
>>> eval('math.sqrt(5)')
2.2360679774997898
>>> eval('type(math.pi)')
<type 'float'>
```

Write a function called `eval_loop` that iteratively prompts the user, takes the resulting input and evaluates it using `eval`, and prints the result.

It should continue until the user enters `'done'`, and then return the value of the last expression it evaluated.

Introduction to Python

- In the previous class, you have learned for and while loops.
- Today we are going to learn some more fancy features about python.

Sorting

Sort from smallest to largest

```
x = [4,1,2,3]
y = sorted(x)
x.sort()
```

Sort from largest to smallest (absolute value)

```
x= sorted([-1, -4, 2, 4], key = abs, reverse = True)
```

OR

```
x= sorted(x, key = abs, reverse = True)
```

List comprehensions

Create a list based on certain rules or from another list.

```
l1 = [x for x in range(5)] # [0, 1, 2, 3, 4]
l2 = [x for x in range(5) if x%2 == 0] # [0, 2, 4]
l3 = [ x * x for x in range(5)] # [0, 1, 4, 9, 16]
l4 = [x * x for x in l2] # [0 , 4, 16]
```

Modules

- When a Python program starts it only has access to a basic functions and classes.

  ("int", "dict", "len", "sum", "range", ...)
- "Modules" contain additional functionality.
- Use "import" to tell Python to load a module.

```
>>> import math
>>> import nltk
```

Import the math module

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.cos(0)
1.0
>>> math.cos(math.pi)
-1.0
>>> dir(math)
['__doc__', '__file__', '__name__', '__package__', 'acos', 'acosh',
'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',
'cosh', 'degrees', 'e', 'exp', 'fabs', 'factorial', 'floor', 'fmod',
'frexp', 'fsum', 'hypot', 'isinf', 'isnan', 'ldexp', 'log', 'log10',
'log1p', 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'trunc']
>>> help(math)
>>> help(math.cos)
```

Modules: Imports

import mymodule	Brings all elements of mymodule in, but must refer to as mymodule.<elem>
from mymodule import x	Imports x from mymodule right into this namespace
from mymodule import *	Imports all elements of mymodule into this namespace

Import the math module

```
>>> import math
math.cos
>>> from math import cos, pi
cos
>>> from math import *
```

Randomness

Sometimes you need to generate random data in your experiment

```
import random
random.random() # uniform value between 0 and 1
random.seed(x) #why would we want to use a seed?
random.randrange(10) # from 0 to 9
random.randrange(3,6) # from 3 to 5
list = range(60)        # generate a list
random.shuffle(list) # reorders elements in a list
random.sample(list,4) # get 4 elements from the list with no repetitions
random.choice(list) # get 1 element from the list
```

Files: Input

inflobj = open(‘data’, ‘r’)	Open the file ‘data’ for input
S = inflobj.read()	Read whole file into one String
S = inflobj.read(N)	Reads N bytes (N >= 1)
L = inflobj.readlines()	Returns a list of line strings

Reading files example

```
#Using dictionaries to count occurences:
name_count = dict()
>>>for line in open('names.txt'):
...         name = line.strip()
...         name_count[name] = name_count.get(name,0)+ 1
...
>>> for (name, count) in name_count.items():
...         print name, count
...
```

Files: Output

outflobj = open(‘data’, ‘w’)	Open the file ‘data’ for writing
outflobj.write(S)	Writes the string S to file
outflobj.writelines(L)	Writes each of the strings in list L to file
outflobj.close()	Closes the file

File output example

```
>>>input_file = open(“in.txt")
output_file = open(“out.txt", "w")
for line in input_file:
        output_file.write(line)
```

“w” = “write mode”
“a” = “append mode”
“wb” = “write in binary”
“r” = “read mode” (default)
“rb” = “read in binary”
“U” = “read files with Unix or Windows line endings”

Exercise in groups

1. Create a list with 500 random numbers. # ran_list = [random.random() for _ in range(500)]
2. Print the list
3. Save these numbers to a file - Random.txt
4. Read the file: Random.txt and randomly pickup 30 number from the file
5. Sort these numbers from largest to smallest
6. Display the elements between 10th and 20th position.

Group project 1

Check the website. **Due date is March 23.**