Math 2300: Discrete Mathematics

Introduction to Python 4 - Functions Textbook Section

Terms, Concepts, and Examples

A function is a set of statements that take inputs, do some specific computation and produces output. The idea is to put some commonly or repeatedly done task together and make a function, so that instead of writing the same code again and again for different inputs, we can call the function. You have already seen some Python defined functions like print( ), len( ) and range( ). You can also define your own functions, called user defined functions.

- A **function** is a block of code which only runs when it is called. You can pass data, known as **parameters**, into a function. The function can also return data as a result.

  The general syntax for a function is

  ```
  def nameIt(argument):
          indentedActionBlock
  ```

  The nameIt is the name of the function. The argument is the input (which could be multiple things separated by commas). The indentedActionBlock is what you want the function to do.

- After defining the function, you can **call the function** with an argument to return a value.

  Example:

  ```
  def evenOdd( x ):
      if (x % 2 == 0):
          print "even"
      else:
          print "odd"
  ```

  Video Example of a Function

Practice Problems

1. Use the following definition of the function front9 to find the output of the program for the given lists.

   ```
   def front9(nums):
     i = 0
     while (i < len(nums) and i < 4):
       if nums[i] == 9:
         return True
       i += 1
     return False
   ```

   (a) front9([1,2,3,9,4])

   (b) front9([1,2,3,4,9,5,0])

(c) front9([5,5])

(d) front9([1,9,9])

2. Write Python code to define each of the following functions. The information under each line tells you what to name your function and the variable input names.

   (a) You and your date are trying to get a table at a restaurant. The parameter "you" is the stylishness of your clothes, in the range 0..10, and "date" is the stylishness of your date's clothes. The result getting the table is encoded as an int value with 0=no, 1=maybe, 2=yes. If either of you is very stylish, 8 or more, then the result is 2 (yes). With the exception that if either of you has style of 2 or less, then the result is 0 (no). Otherwise the result is 1 (maybe).

   Examples:

   dateFashion(5, 10) → 2   dateFashion(5, 2) → 0   dateFashion(5, 5) → 1

```
def dateFashion(you, date):
```

   (b) Given two lists of integers, a and b, each length 3, return a new list of length 2 containing their middle elements.

   Examples:

   middle([1, 2, 3], [4, 5, 6]) → [2, 5]   middle([7, 7, 7], [3, 8, 0]) → [7, 8]
   middle([5, 2, 9], [1, 4, 5]) → [2, 4]

```
def middle(a, b):
```

   (c) Given a list of integers length 2, return True if it contains a 2 or a 3.

   Examples:

   has23([2, 5]) → True   has23([4, 3]) → True   has23([4, 5]) → False

---

```
def has23(nums):
```

   (d) We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return True if we are in trouble.

   Examples:

   parrotTrouble(True, 6) → True   parrotTrouble(True, 7) → False
   parrotTrouble(False, 6) → False

```
def parrotTrouble(talking, hour):
```

(e) We have two monkeys, A and B, and the variables aSmile and bSmile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return True if we are in trouble.

Examples:

monkeyTrouble(True, True) $\rightarrow$ True   monkeyTrouble(False, False) $\rightarrow$ True

monkeyTrouble(True, False) $\rightarrow$ False

```
def monkeyTrouble(aSmile, bSmile):
```

(f) Add up all the numbers in a given list. Then print the total.

Examples:

addUp([2, 5, -3, -8, 4, 0]) $\rightarrow$ 0   addUp([-1, 3, 2, -2, -50]) $\rightarrow$ -48

addUp([-1,0,3,4,2,0,-5]) $\rightarrow$ 3

```
def addUp(nums):
```

(g) Print each element in a list twice.

Examples:

print2list([2, 1, 2, 3, 4]) $\rightarrow$

```
2 2
1 1
2 2
3 3
4 4
```

print2list([0,-1,5)] $\rightarrow$

```
 0 0
 -1 -1
 5 5
```

```
def print2list(nums):
```