**Python Practice**

Week 2, January 30, 2018

**#1** Function Interface:

**def squared\_nums(num\_list):**

**"""**

**Squares numbers in num\_list**

**num\_list: list of numbers**

**Returns: list of these numbers squared**

**"""**

What the Function Does:  
 Each number in the list will be squared to create a new list of squared numbers

Test Cases:  
 [1, 2, 3], [], [-1, -2, -3], [1, 0]

Expected Results:  
 [1, 4, 9], [], [1, 4, 9], [1, 0]

**#2** Function Interface:

**def check\_title(title\_list):**

**"""**

**Removes strings in title\_list that have numbers and aren't title case**

**title\_list: list of strings**

**Returns: list of strings that are titles**

**"""**

What the Function Does:  
 Each string in the list that has numbers and isn’t in title case is removed to create a new string  
Test Cases:  
 ['Hello World', 'Hello\_world', 'Title'],   
 ['Hello\_World', 'A great 3 Days', 'hello World'], [‘10, 11, 12’]

Expected Results:  
 ['Hello World', 'Hello\_world', 'Title'], [], []

**#3** Function Interface:  
 **def restock\_inventory(inventory):**

**"""**

**Increases inventory of each item in dictionary by 10**

**inventory: a dictionary with:**

**key: string that is the name of the inventory item**

**value: integer that equals the number of that item currently on hand**

**Returns: updated dictionary where each inventory item is restocked**

**"""**

What the Function Does:  
Updates a dictionary that keeps track of an inventory, in order to increase the amount of items on record

Test Cases:  
{'pencil':10, 'pen':8, 'paper':7}, {'pencil':0, 'pen':-3, 'paper':-11},   
{'pencil':0.5, 'pen':-3.2, 'paper':11.0}

Expected Results:  
{'pencil':20, 'pen':18, 'paper':17},   
{'pencil':10, 'pen':7, 'paper':-1},   
{'pencil':10.5, 'pen':6.8, 'paper':21.0}

Actual Results:  
{'pen':18, 'paper':17, 'pencil':20},   
{'pen':7, 'paper':-1, 'pencil':10},   
{'pen':6.8, 'paper':21.0, 'pencil':10.5}

**#4** Function Interface:  
**def filter\_0\_items(inventory):**

**"""**

**Removes items that have a value of 0 from a dictionary of inventories**

**inventory: dictionary with:**

**key: tring that is the name of the inventory item**

**value: nteger that equals the number of that item currently on hand**

**Returns: the same inventory\_dict with any item that had 0 quantity removed**

**"""**

What the Function Does:  
Updates the dictionary so that all items that their are currently none of are removed from the dictionary

Test Cases:  
{'pencil':10, 'pen':8, 'paper':7}, {'pencil':0, 'pen':-3, 'paper':-11},   
{'pencil':0.5, 'pen':-3.2, 'paper':0.0}

Expected Results:  
{'pen':8, 'paper':7, 'pencil':10}, {'pen':-3, 'paper':-11},   
{'pen':-3.2, 'pencil':0.5}

**#6** Function Interface:  
 **def average\_grades(grades):**

**"""**

**Takes grade values from a dictionary and averages them into a final grade**

**grades: a dictionary of grades with:**

**key: string of students name**

**value: list of integer grades received in class**

**Returns: dictionary that averages out the grades of each student**

**"""**

What the Function Does:  
Changes the dictionary so each student has one average grade associated with them rather than a list of grades

Test Cases:  
{'Michael' :[100, 78, 88, 900/10], 'Daniel':[80, 95, 77, 64.0], 'Josh':[99, 89, 94, 66]},   
{'Michael' :[5\*20, 188 \* .5, 88], 'Daniel':[80.5, .15, 66, 64.0] 'Josh':[99 + 1 \* -2, 40/.5]},   
{'Michael' :[78], 'Daniel':[90], 'Josh':[900/-9]}

Expected Results:  
{'Josh' : 87.0, 'Daniel': 79.0, 'Michael': 89.0},  
{'Josh' : 88.5, 'Daniel': 52.6625, 'Michael': 94.0},  
{'Josh' : -100.0, 'Daniel': 90.0, 'Michael': 78.0}