Week 2 Quiz

Qi Meng - qm2162

Instructions

Replace the Name and UNI in cell above and the notebook name

Replace all '__' below using the instructions provided.

When completed,

- make sure you've replaced Name and UNI in the first cell and filename (eg: Week_02_Quiz-brg2130)
- Kernel->Restart & Run All to run all cells in order
- use Print Preview, Print-> Save to pdf
- and post pdf to GradeScope

---

1. Lists

In [1]:
```
# Create a list containing the strings 'blue','red','green'
colors = ['blue','red','green']

# Assert that value at index 0 of the list colors is equal to 'blue'
assert colors[0] == 'blue'

# Using list indexing, print out the value of colors at index 1
# You should see the output "red" without quotes
print(colors[1])
```

```
red
```

2. Dicts

In [2]:
```
# Create a dictionary which maps the string keys 'zero','one','two'
#    to the int values 0,1,2
str_to_int = {'zero': 0, 'one': 1, 'two': 2}

# Assert that the value returned for key 'two' equals 2 in str_to_int
assert str_to_int['two'] == 2

# Using str_to_int, print out the value for the key 'one'
# You should see the output 1
print(str_to_int['one'])
```

```
1
```

3. String Formatting And For Loops

In [3]:
```
# Using the len function and f"" string formatting, print the number of eleme
print(f"the length of colors is {len(colors)}")

# Using the enumerate function, the colors list defined above, and f"" string
#   for every index,value pair from enumerate(colors)
#   print "the value at index {index} is {value}"
#   Ex:
#     the value at index 0 is blue
#     the value at index 1 is red
#     the value at index 2 is green
for idx, clr in enumerate(colors):
    print(f"the value at index {idx} is {clr}")
```

```
the length of colors is 3
the value at index 0 is blue
the value at index 1 is red
the value at index 2 is green
```

4. List Comprehension

In [4]:
```
# Using a list comprehension and the len() function,
#    create a list corresponding to the lengths of each of the strings in col
# Store the resulting list in variable color_lengths
color_lengths = [len(clr) for clr in colors]

# Assert that the first value in color_lengths is 4 (the length of 'blue')
assert color_lengths[0] == 4
```

5. Functions and Control Flow

In [5]:
```
# Define a function called append_even_odd
# It should expect to take in a string
#    if the string is empty (has length of 0), return 'empty'
#    else if the string has an even number of characters, return the string w
#    else if the string has an odd number of characters, return the string wi
# For example: 'blue' should become 'blue_even'
def append_even_odd(s):
    if len(s) == 0:
        return 'empty'
    elif len(s) % 2 == 0:
        return s + '_even'
    else:
        return s + '_odd'


assert append_even_odd('test') == 'test_even'
assert append_even_odd('one') == 'one_odd'
assert append_even_odd('') == 'empty'
```

6. Sorting

In [6]:
```
# Using sorted(), sort the list color_lengths created above, descending in va
# Save as color_lengths_sorted
color_lengths_sorted = sorted(color_lengths, reverse=True)

# Assert that the last element of color_lengths_sorted is 3
assert color_lengths_sorted[-1] == 3
```

For More Practice (not required):

In [7]:
```
# Create a list of the key,value pairs in the str_to_int dictionary sorted by
str_to_int_sorted = sorted([(k, v) for k, v in str_to_int.items()], key = lam

# assert that the first element of str_to_int_sorted is ('two',2)
assert str_to_int_sorted[0] == ('two', 2)
```

In [ ]:
```

```