Week_02_Quiz-JEG2253

September 16, 2025

1 Week 2 Quiz

Due on Tuesday, Sep 16 at 11:59 pm

1.1 Jayat Gonzalez Palomeras - JEG2253

1.1.1 Instructions

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

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-ac5562) - Kernel->Restart & Run All to run all cells in order - use Print Preview, Print-> Save to pdf - post pdf to GradeScope

---

1.2 1. Lists

[2]:
```
# 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'
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
```

1.3 2. Dicts

[3]:
```
# 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
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
```

1.4 3. String Formatting And For Loops

[6]:
```
# Using the len function and f"" string formatting, print the number of␣
 ↪elements in colors defined above.
print(f"the length of colors is {len(colors)}")

# Using the enumerate function, the colors list defined above, and f"" string␣
 ↪formatting
#   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 index, value in enumerate(colors):
    print(f"the value at {index} is {value}")
```

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

1.5 4. List Comprehension

[7]:
```
# Using a list comprehension and the len() function,
#    create a list of the character lengths of each of the strings in colors␣
 ↪(eg. 'blue' -> 4)
# Store the resulting list in variable color_lengths
color_lengths = [len(color) for color in colors]

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

[7]:
```
True
```

1.6 5. Functions and Control Flow

[ ]:
```
# 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␣
 ↪with '_even' added to the end
```

```
#    else if the string has an odd number of characters, return the string with␣
 ↪'_odd' added to the end
# For example: 'blue' should become 'blue_even'
def append_even_odd(string):
    if len(string) == 0:
        return 'empty'
    elif len(string) % 2 == 0:
        return '_even'
    else:
        return '_odd'

print(append_even_odd('test') == 'test_even') #removed the asserts here because␣
 ↪it crashed, switched to print
print(append_even_odd('one') == 'one_odd')  #to be able to see the results for␣
 ↪each
print(append_even_odd('') == 'empty')
```

```
False
False
True
```

1.7 6. Sorting

[17]:
```
# Using sorted(), sort the list color_lengths created above, descending in␣
 ↪value (reverse=True)
# Save as color_lengths_sorted
color_lengths_sorted = sorted(colors, reverse=True)

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

[17]:
```
False
```

1.7.1 For More Practice (not required):

[19]:
```
# Create a list of the key,value pairs in the str_to_int dictionary sorted by␣
 ↪value, decending
str_to_int_sorted = sorted(str_to_int.items(), key=lambda x: x[1], reverse=True)

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

[19]:
```
True
```