

Lists and Tuples

By Mark Luu, for CMPT 141

Exercise 1: is, and equals

```
a = [1, 3, 5, 7, 9]
b = [1, 3, 5, 7, 9]

c = 1
d = 1

print(a == b)
print(a is b)

print(c == d)
print(c is d)
```

Exercise 2: Creating and modifying

```
import copy

srimp2 = ["I", "am", "the", "way,", "the", "truth,", "and", "the", "life.", "No", "one",
          "comes", "to", "the", "father", "except", "through", "-", "$6", "SRIMP", "SPECIAL!"]

# If you're doing it textbook style, you can use srimp3 = srimp2.copy() instead.
srimp3 = copy.deepcopy(srimp2)

for word in srimp2:
    print(word, end = " ")

srimp3[0] = "You"

for word in range(0, len(srimp3)):
    print(srimp3[word], end = " ")

print(srimp2[0:])
```

Exercise 3: References and index/slicing

Create a variable named `even_numbers` with the contents `[2, 4, 6, 8, 10]`.

Create a variable named `even_reference` with the value of `even_numbers`.

Create a variable named `even_deepcopy` using the `copy` library's `deepcopy(data_here)` function or the `.copy()` method to duplicate `even_numbers`.

Change the first value of `even_numbers` to 12.

Change the second value of `even_deepcopy()` to 12.

Use a `for` or `while` loop to print the first three numbers of `even_reference`.

Use slicing to print a list of the first three numbers of `even_deepcopy`.

Exercise 4: Nested Lists

```
milky2 = ["milk", "milk", "milk", "milk", ["milk", "milk", "milk", "milk", None], "milk", "milk",  
         , "milk", "milk"], "milk", "milk", "milk", "milk"]  
print(milky2)  
print(milky2[4])  
print(milky2[4][4][0])
```

Exercise 5: List Comprehension

```
milks = ["milk", "milk", "milk", "milk", ["milk", "milk", "milk", "milk", [None], "milk", "milk",  
      , "milk", "milk"], "milk", "milk", "milk", "milk"]  
  
|  
print([milk for milk in milks if milk == "milk"])  
print([milk for milk in milks])
```

Exercise 6: Common list tools

```
1 milk2 = ["milk", "milk", "milk", "milk",["milk", "milk", "milk", "milk",[None], "milk",  
2     "milk", "milk", "milk"], "milk", "milk", "milk","milk"]  
3  
4  
5 orange_juice = ["orange juice", "orange juice", "orange juice", "orange juice", "orange  
6     juice"]  
7  
8  
9 number_salad = [7, 3, 5]  
10 number_salad.sort()  
11 print(number_salad)  
12 milk2 = milk2 + orange_juice + number_salad  
13 milk2.append("soda")  
14 milk2.remove("milk")  
15  
16  
17 milk2[0] = "sauce"  
18 del milk2[3][4]  
19 print(milk2)
```

Exercise 7: Tuples

```
family = (("brother", 16), ("sister", 16), ("mom", 42),  
          ("dad", 42))  
  
print(family[2][1])
```


Exercise 8: Nested list work

Create a variable named “student” with the value [].

Append the list [“Grug”, ggg666, 99999999] to student.

Remove 99999999 from the Grug list.

Delete the value at index 1 from the Grug list.

Append the list [“Jerry”, jjj111, 11111111] to student.

Use a list comprehension to print every value of student.

Create a variable named “even_numbers” with the tuple (2, 4, 6, 8, 10).

Try to apply the .sort() method on it.

Exercise 9: Mutability

Is a list a mutable object? Is a tuple?