

1. Describe the differences between lists, strings, and tuples:

2. When would you want to use a tuple?

3. For each the following code snippets, decide if it causes an error. What is the output?

```
my_tuple = "cs", "111", "intro"
my_second_tuple = ("cs", "201", "data structures")
print(my_tuple)
print(my_second_tuple)
```

```
first_tuple = (1,2,5,7)
second_tuple = ("my", "tuple", "is", "best")
print(first_tuple + second_tuple)
```

```
new_tup = (1, 4, 5, 6, 7)
new_tup[4] = 9
print(new_tup)
```

```
nested_tuple = (1, 4, 5, (7, 2, 9), "hi", (2, 3))
print(nested_tuple[0])
print(nested_tuple[3][2])
print(nested_tuple[5][1])
print(nested_tuple[4])
print(nested_tuple[2][0])
```

4. Go through the following program and label each variable as either a local or global variable. Then, write what the output would be

```
a = 9
b = 10

if(a < b):
    c = 5

def mystery(x,y):
    c = 4
    d = 2
    return (x + y + c) / d

result = mystery(a,b)
print(result)
print(a)
print(b)
print(c)
print(d)
```

5. What is the output of the following snippet?

```
my_list = [3, 5, 7, 9]
your_list = my_list
your_list[0] = 1
print(my_list)
print(your_list)
```

6. Fill in the blank:

In the snippet above, *your_list* is an _____ for *my_list*