Additional Tuple Examples:

CS 1112 | Professor Basit | Supplemental material to topic on Tuples

-- Adding items to a Tuple?

We're aware we cannot modify a tuple. There is no built-in append() method available. So, what can we do?

- **Convert the tuple into a list**, add the item, and convert it back!

```
thistuple = ("apple", "banana", "cherry")
y = list(thistuple)   # type cast the tuple into a LIST
y.append("orange")    # add the new item to the LIST
thistuple = tuple(y)  # type cast back into a tuple
```

- You can **add a tuple to a tuple** (we've seen a similar example)

```
thistuple = ("apple", "banana", "cherry")
y = ("orange",)  # Create a NEW tuple with the item
thistuple += y   # Add it to the existing tuple
```

(Examples thanks to w3schools.com)

-- Returning More Than One Value from a Function?

Functions can only return a single value. However, we can use a tuple and put as many values as we need inside it, then return the tuple object as the function's return value.

```
def sum_and_avg(x, y, z):
    s = x + y + z
    a = s/3
    return (s, a) # return two values inside of a tuple

(S, A) = sum_and_avg(3, 8, 5)  # Call the function
# Receive the two values in two variables called S and A
print('Sum =', S)
print('Avg =', A)
```

The result/output would be Sum = 16 and Avg = 5.333…

(Example thanks to dataquest.io)

-- How to use the **all()** function in conjunction with tuples?

*The Python **all()** function returns true if all the elements of a given iterable (List, Dictionary, Tuple, set, etc.) are True otherwise it returns False. It also returns True if the iterable object is empty. It can also be used to test a condition on all elements of a tuple.*

- Checking if a **tuple is empty**

```
# Empty tuple and list
tup = ()
print(all(tup))  # Output will be True

lst = []
print(all(lst))  # Output will be True
```

- Checking **if all elements meet certain criteria**: greater than zero

```
# all() with condition - if all elements are greater than 0
tup2 = (1, -3, 0, 2, 4)
print(all(ele > 0 for ele in tup2))  # Output: False
```

- Checking **if all elements meet certain criteria**: even

```
# all() with condition - if all elements are even
tup3 = (2, 4, 6, 8, 10)
print(all(ele % 2 == 0 for ele in tup3))  # Output: True
```

Note: think of the structure of the code as…
*For **every element** in the tuple, **if** each element is **even** then return **True** **otherwise** return **False**.*
That means, if even **one** element is **odd**, then the statement will return False.

(Examples thanks to geeksforgeeks.org)