# Lists and Tuples

## Exercises

### Week 6

Would you describe the following Python statement as a **function call**? Or a **method call**?

names.reverse()

*Answer:*

It is described as a method call.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a Python statement that appends a single element to the end of the specified *List* using a **method** call.

prices = [2.65, 7.65, 8.25, 9.56]

*Answer:*

prices = [2.65, 7.65, 8.25, 9.56]

prices.append(1.22)

print(prices)

Write another statement that appends three elements to the end of the specified *List* using a single **method** call.

*Answer:*

prices = [2.65, 7.65, 8.25, 9.56]

prices.extend([4.32, 8.21,1.22])

print(prices)

Now write a for loop that *iterates* over each value in the list and prints it to the screen.

*Answer:*

prices = [2.65, 7.65, 8.25, 9.56, 2.65, 7.65, 8.25, 9.56]

for price in prices:

print(price)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is a method that changes the contents of the associated value referred to as a **mutator**? Or an **accessor**?

*Answer:*

A method that changes the contents of the associated value is referred to as a mutator.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would the contents of the primes list look like after execution of the following statements?

primes = [ 2, 3, 5, 7, 11, 13, 17, 19 ]

primes.pop()

*Answer:*

primes = [ 2, 3, 5, 7, 11, 13, 17 ]

primes.reverse()

*Answer:*

primes = [ 19, 17, 13, 11, 7, 5, 3, 2 ]

primes.remove(7)

*Answer:*

primes=[19, 17, 13, 11, 5, 3, 2]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Provide an example of how the insert() method could be used to add a value of 10 to the beginning of the list shown below.

temps = [ 32, 46, 95, 10, 50 ]

*Answer:*

temps = [ 32, 46, 95, 10, 50 ]

temps.insert(0, 10)

print(temps)

Now write a statement that uses an *accessor* method to find the index of the value 95 within the list.

*Answer:*

temps = [10, 32, 46, 95, 50]

index\_of\_95 = temps.index(95)

print("Index of 95:", index\_of\_95)

Finally write a statement that uses another *accessor* method to count how many times the number 10 appears within the list.

*Answer:*

count = temps.count(10)

print(count)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be stored in the list samples after the following statements were executed?

samples = [ 100.2, 100.6, 99.2, 765.2, 900.2, 400 ]

samples = samples.reverse()

*Answer:*

[400, 900.2, 765.2, 99.2, 100.6, 100.2]

Explain why this is the case.

*Answer:*

The reason is that Python's reverse() method is an in-place procedure. It alters the current list without making a new one by reversing the components of the list in place.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a Python program that uses a **List-Comprehension** to produce the same list as the following code -

values = []

for n in range(100,200):

values.append(x\*x)

*Answer:*

values = [x\*x for x in range(100, 200)]

Now, amend your code so that it only includes even numbers.

*Answer:*

values = [x\*x for x in range(100, 200) if x % 2 == 0]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the *data-type* of the following value?

info = ("Ken", "bae-192", 62)

*Answer:*

Tuple

Is a Tuple **mutable** or **immutable**?

*Answer:*

Immutable

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a statement that creates a Tuple that contains a single element.

*Answer:*

hi\_tuple = (11,)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a single Python statement that **unpacks** the following Tuple into three variables, called x, y and z.

coord = (100, 200, 150)

*Answer:*

x, y, z = coord

Write another statement that uses indexing to access the second element of the Tuple and store it in a variable called ‘height’

*Answer:*

height = coord[1]

Finally write a ‘for’ loop that prints each value within the Tuple.

*Answer:*

for value in coord:

print(value)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

When a Tuple (or any sequence) type value is being passed as an argument to a function, what single character can be used as a prefix to force the sequence to be **unpacked** prior to the call being made?

*Answer:*

The single character used to unpack a sequence (like a tuple) before passing its element as separate arguments to a function is the asterisk(\*).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

When discussing Tuples the phrase **heterogeneous** is sometimes used to describe the type of stored values. What does this mean in practice?

*Answer:*

When a tuple is described as “heterogeneous” it means that it contains elements of different data types.

For example:

(“world”, 2.75, False): This tuple contains a string, a float and a Boolean)

What sister phrase is often used to refer to the type of values stored within a List? And what does this mean?

*Answer:*

The sister phrase often used to refer to the type of values stored within a List is “homogeneous”. This means that all elements within a List must be of the same data type.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**