Lists and Tuples

Exercises

Week 6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

names.reverse()

Answer: 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.append(69.12)

Write another statement that appends three elements to the end of the specified List using a

single method call.

Answer: prices.extend([11.4, 22.65, 103.16])

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

Answer:

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 called 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:

Index = temps.index(95)

print(Index)

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:

Python's reverse() method reverses a list in-place. This means that the

method modifies the original list instead of creating a new one. As a result,

the original list is modified, which is why the second statement has the expected

output.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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(100,200,2)]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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:

apple=("a",)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 item in coord:

print(item)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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:

You can pass the values from the tuple to the function using the asterisk (\*) prefix.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

When discussing Tuples the phrase heterogeneous is sometimes used to describe the type

of stored values. What does this mean in practice?

Answer:

The phrase heterogeneous in tuples means you can mix different types of data within the same tuple—integers,

strings, floats, etc.

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 for lists is homogeneous means that all the elements within the list are of the same data type.