Fundamentals of Computer Programming

Submitted by

Rishav Rauniyar

Section E 8711

BSc (Computing)

Level 4 1st Semester

Submitted to

Krishna Sir

Introduction to Programming

Lab Worksheet

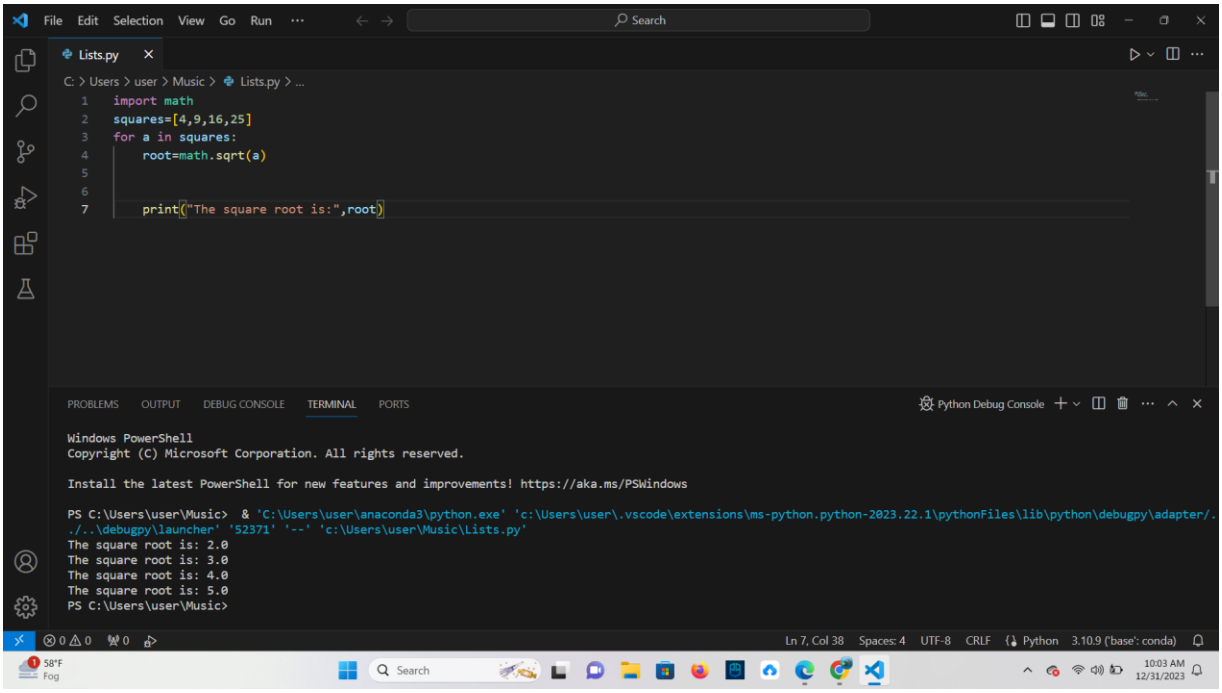
Week 6

Using List Methods

squares = [4, 9, 16, 25]

TASK: Write a for..in loop that iterates over all the elements of the squares list and prints the square root of each to the screen. Hint: you may want to import a function from the math module to help achieve this.

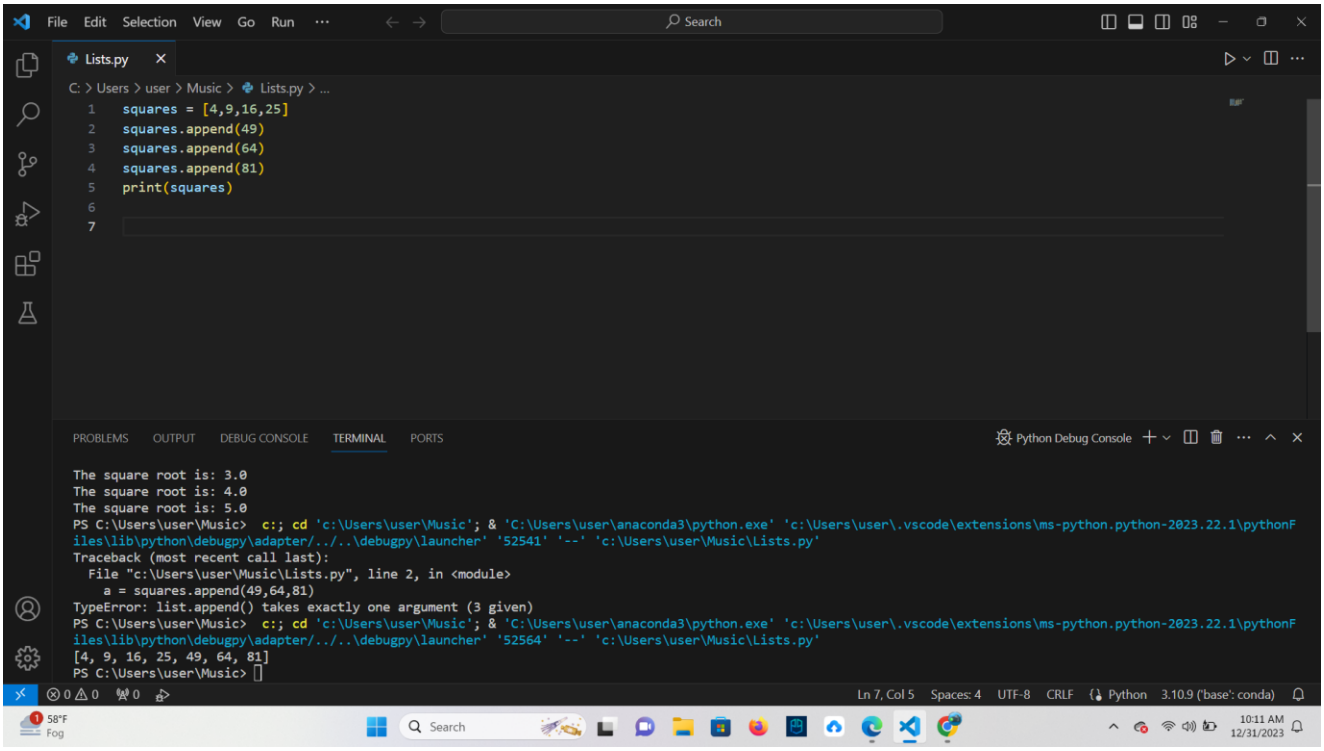
Ans



Introducing Methods

TASK: Write some code that uses the append () method to add the next three-square values (49, 64, 81) to the end of the squares list.

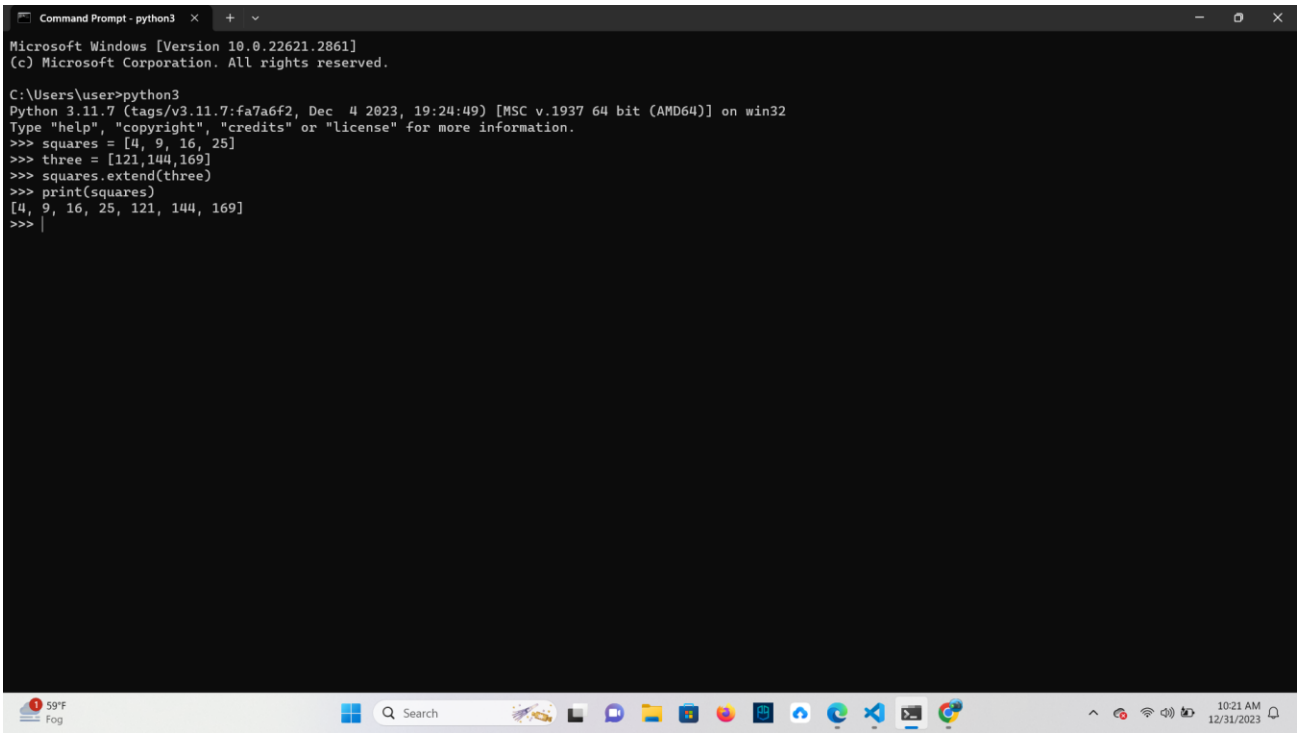
Ans



The extend () method

TASK: Write some code that uses the extend () method to add the next three-square values, starting at 121 (11 x 11), to the end of the squares list.

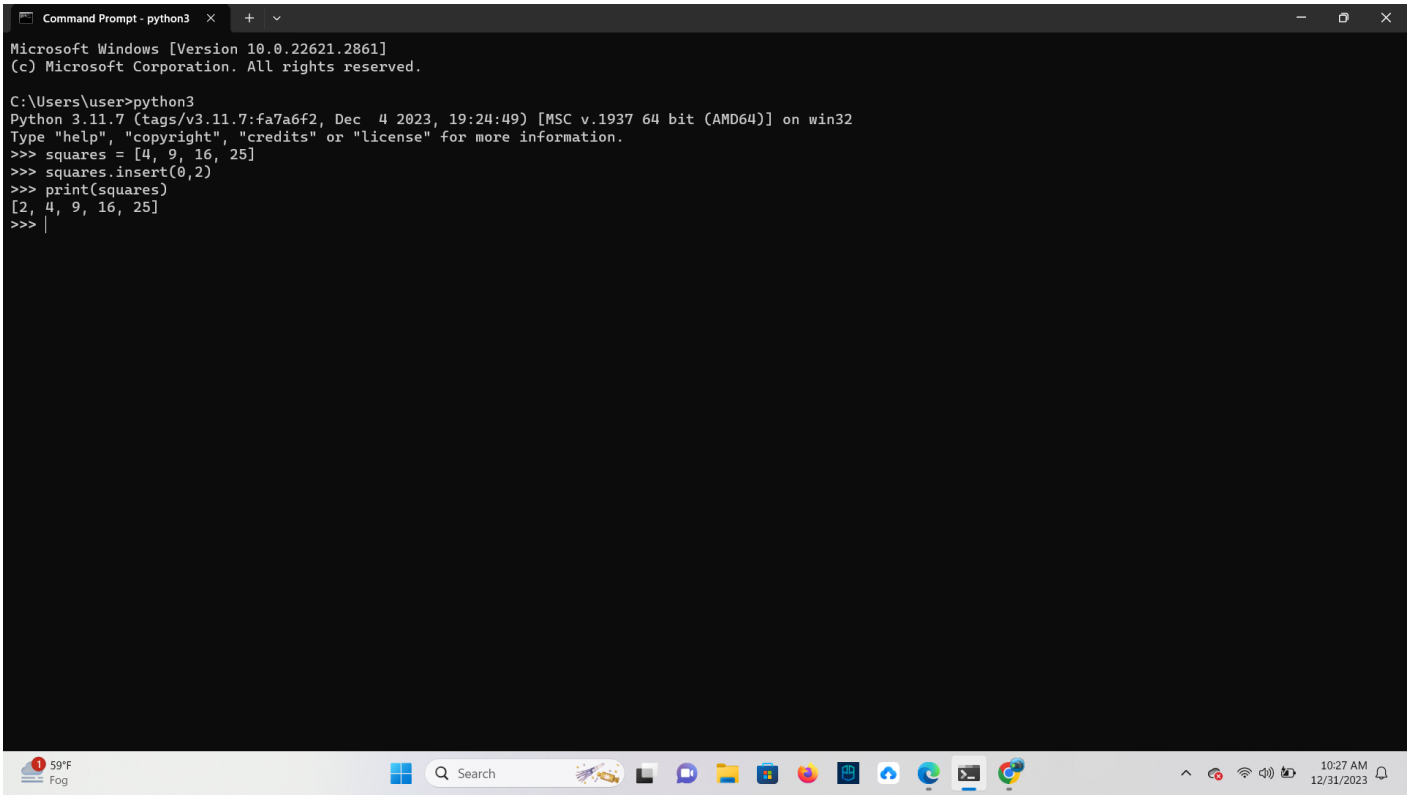
Ans



The insert () method

TASK: Write some code that uses the insert () method to insert the value 2, to the very beginning of the squares list.

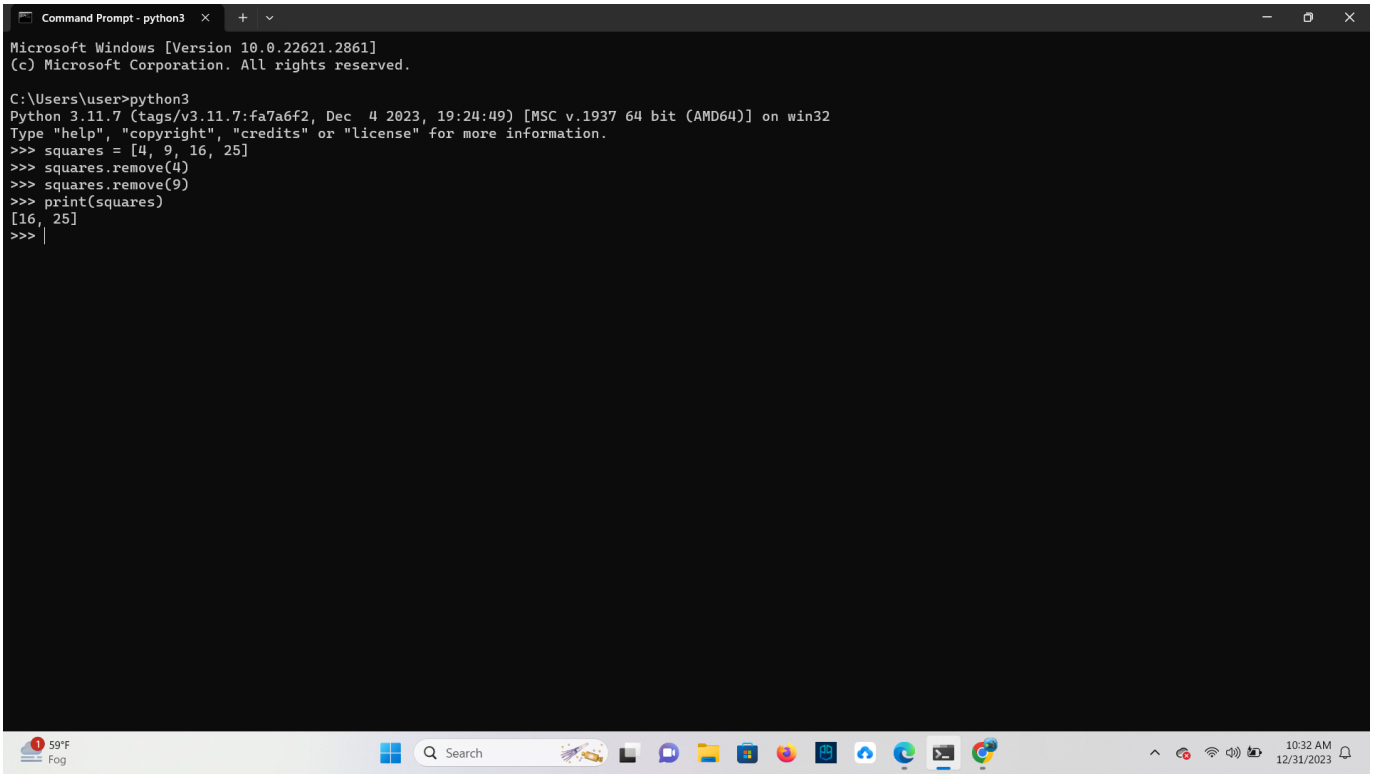
Ans



The remove () method

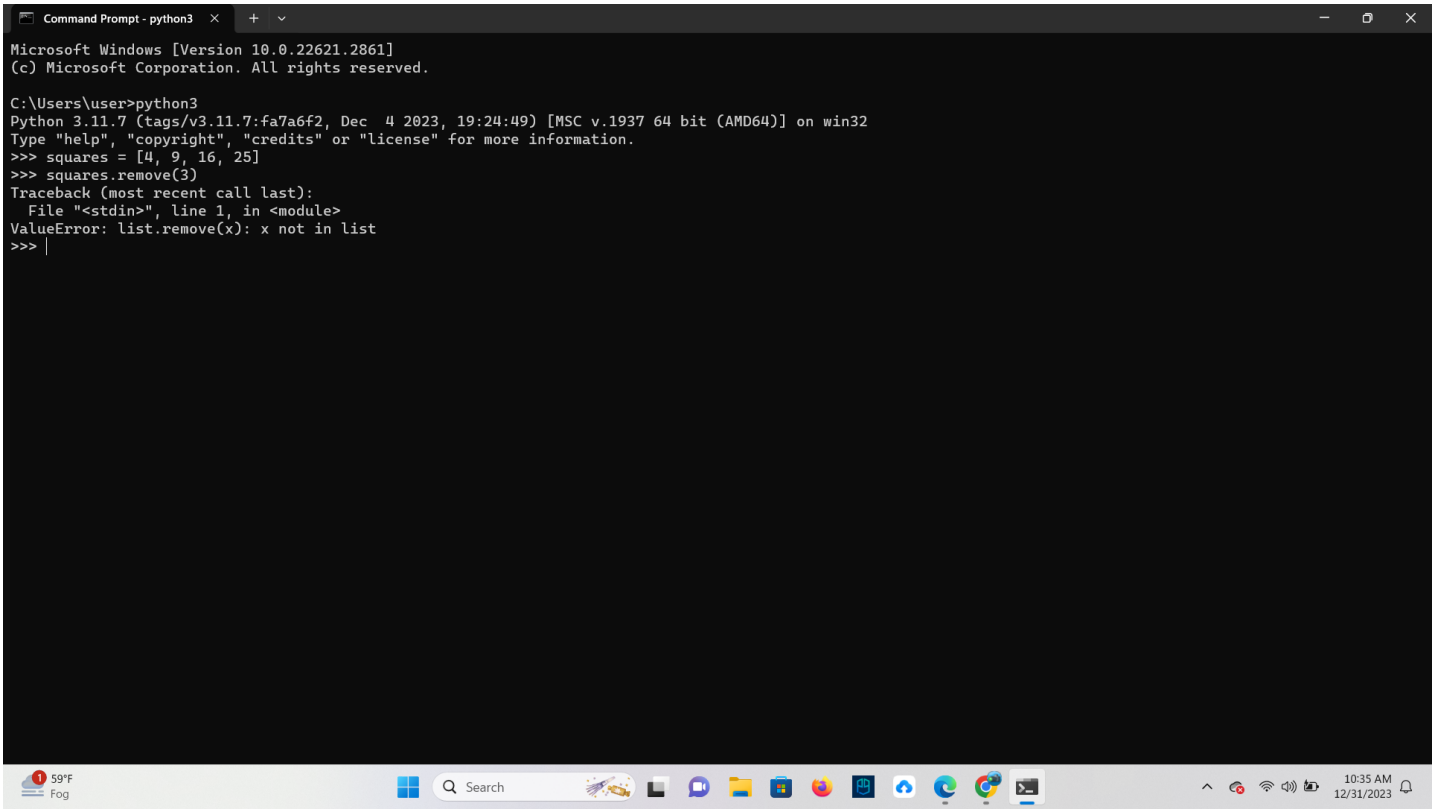
TASK: Write some code that uses the remove () method to remove the value 49 from the squares list. Print the list afterwards to ensure the value has indeed been removed.

Ans



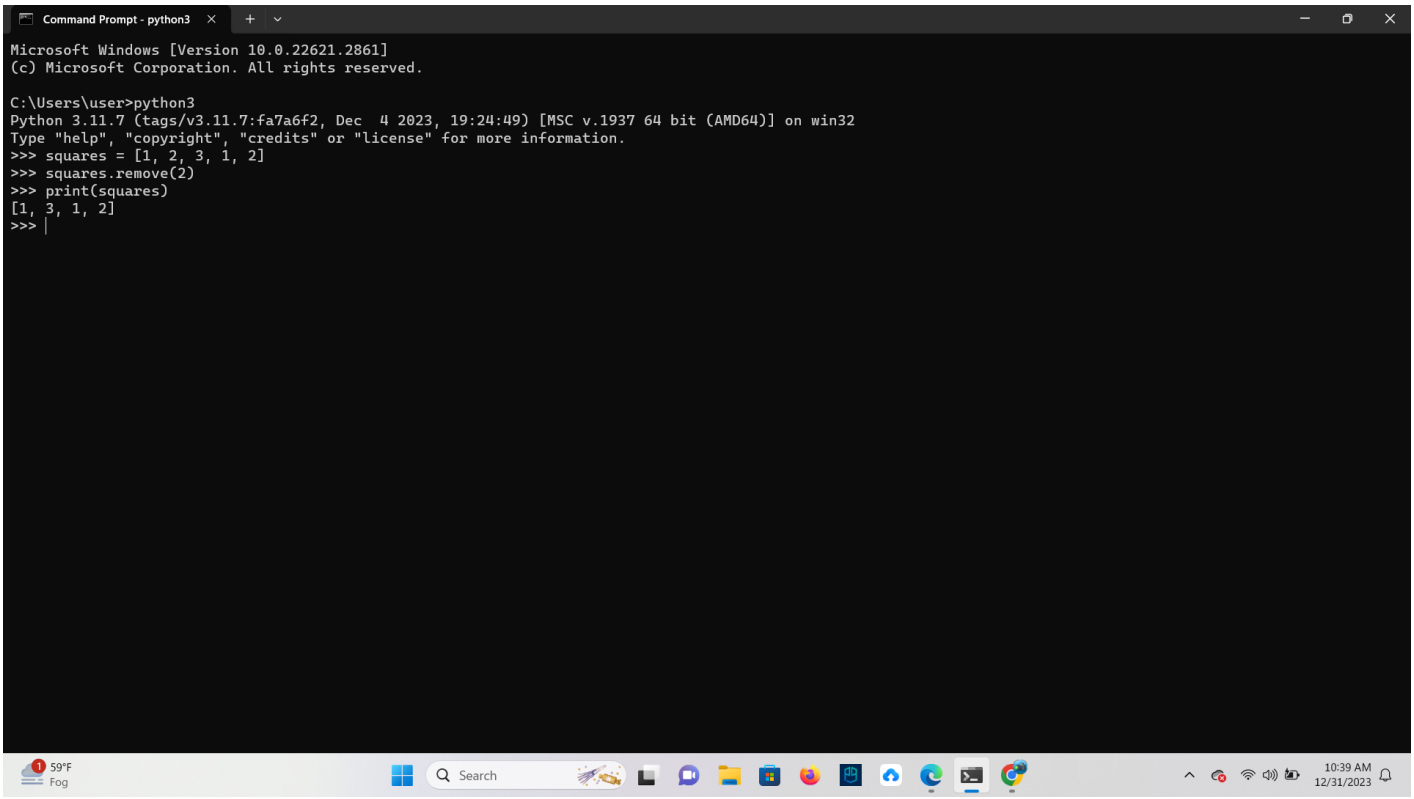
TASK: Write some code that uses the remove () method to remove the value 3 from the squares list. Notice how an error is generated since the given value was not present.

Ans



TASK: Create a simple list that contains the values [1, 2, 3, 1, 2] and then use the remove () method to remove the value 2. Which value is removed?

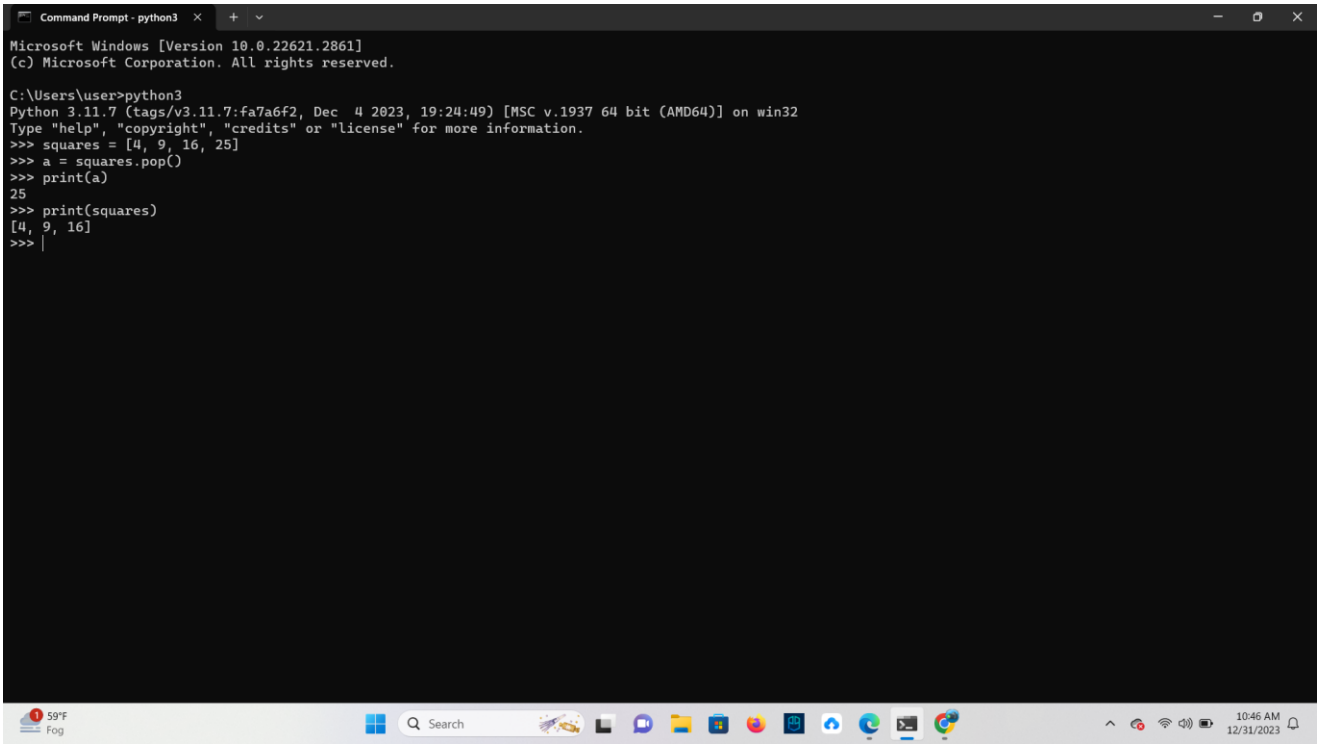
Ans



The pop () method

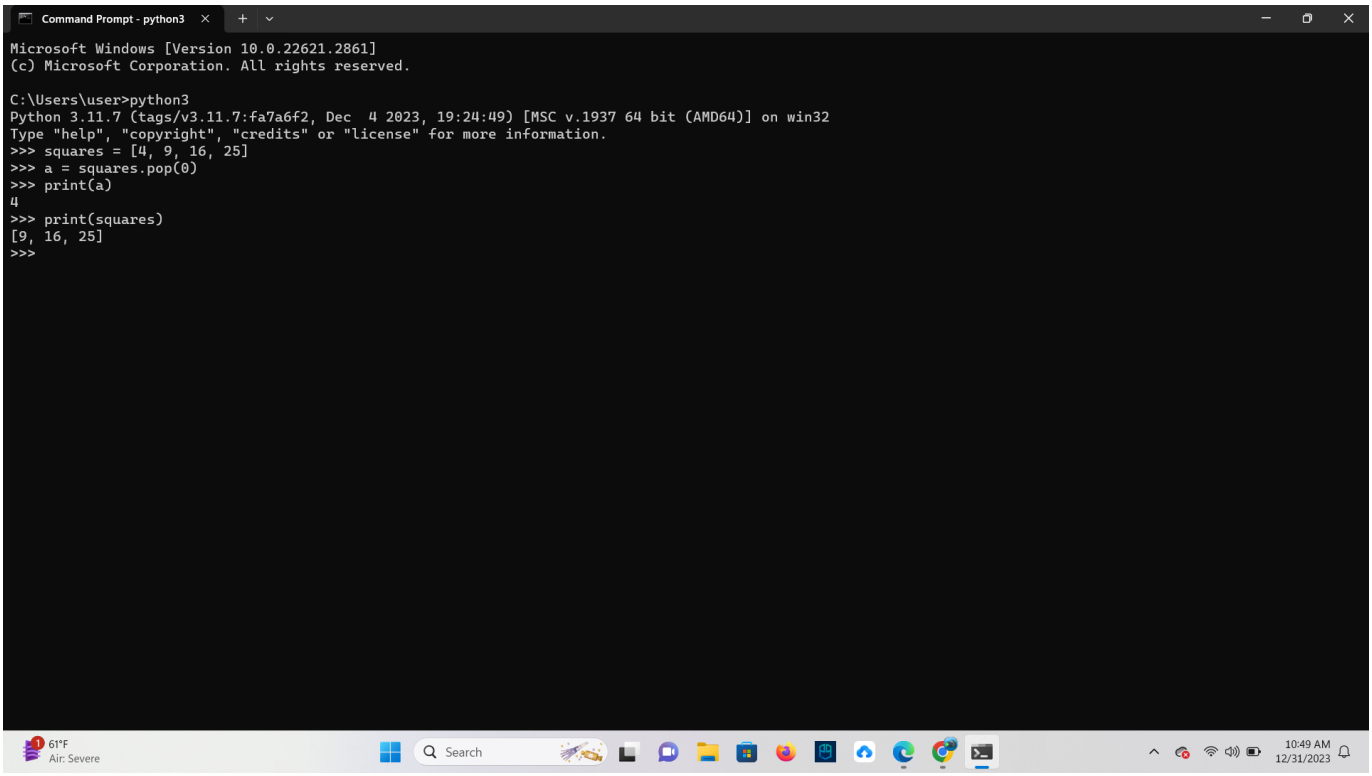
TASK: Write some code that uses the pop () method to remove and display the last value of the squares list. Print the list afterwards to ensure the value displayed has been removed.

Ans



TASK: Write some code that uses the pop () method to remove and display the first value of the squares list. Print the list afterwards to ensure the value has been removed.

Ans



The clear () method

The clear () method mutates a list by removing all elements. After the method call the list still exists, but it is empty.

>>> some_list = [1,2,3,4]

>>> some_list.clear()

>>> print(some_list)

[]

clear () is a very simple method and equivalent to the following slicing assignment -

some_list[:] = []

Again, using the method is probably better practice as the resulting code is easier to understand.

The sort () method

TASK: Write some code that uses the sort () method with no arguments, to alphabetically sort the exact list of names shown below. Display the list after the sort has been called.

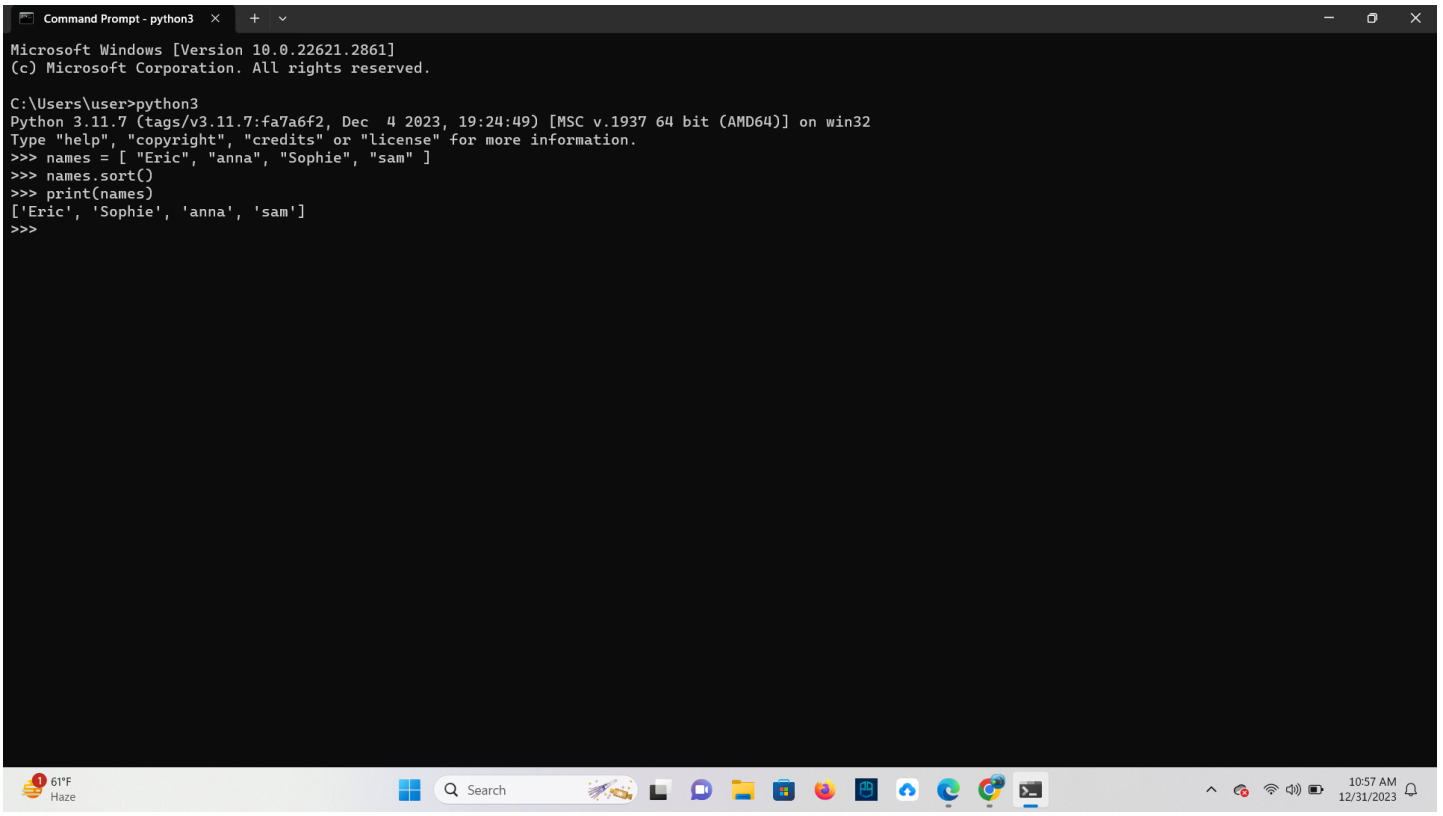
names = ["Eric", "anna", "Sophie", "sam"]

Once you have completed the previous task you should have noticed that the order is not what you may have expected. The result probably looked like this -

['Eric', 'Sophie', 'anna', 'sam']

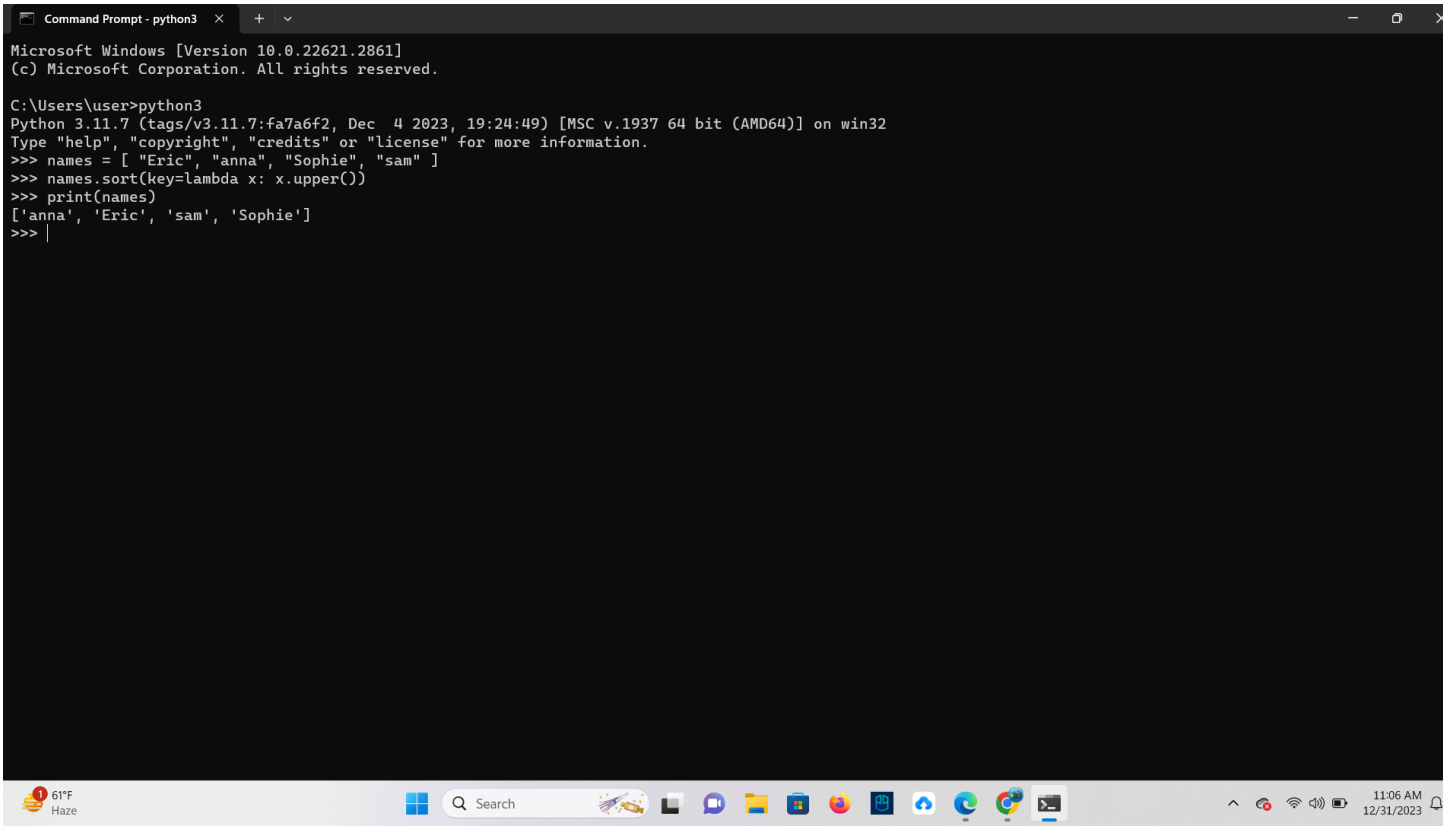
This is because lower-case letters appear later than all upper-case letters when a default sort is applied.

Ans



TASK: Improve your previous solution so that the list is sorted correctly, ignoring the case used to write the names. To achieve this, you will have to include a key argument in the form of a lambda expression that returns each string as uppercase letters only. Hint: you can use the Str. Upper () method to convert a name to uppercase letters.

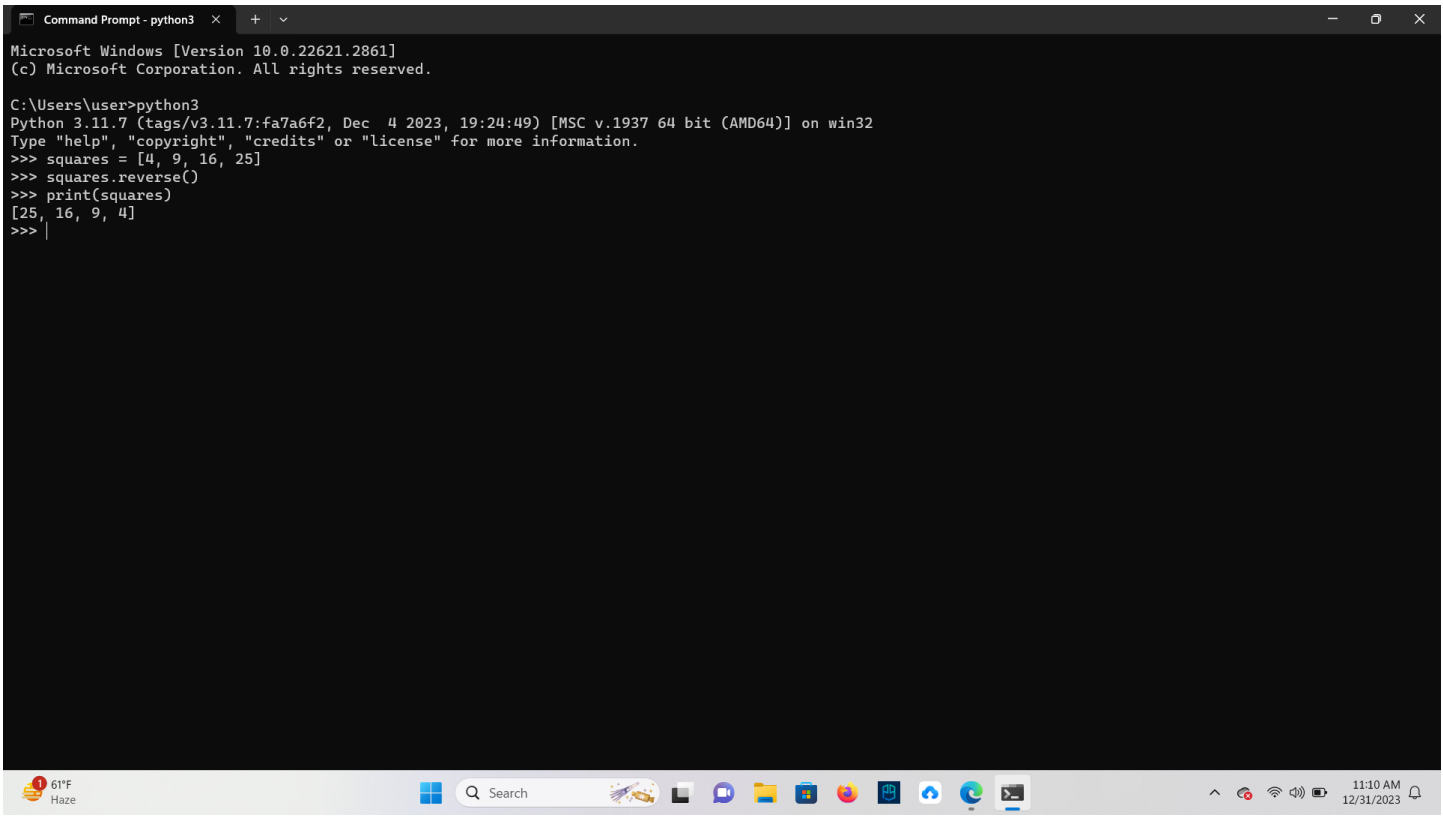
Ans



The reverse () method

TASK: Write some code that uses the reverse () method to reverse the values of the squares list. Print the list afterwards to ensure the values have been reversed.

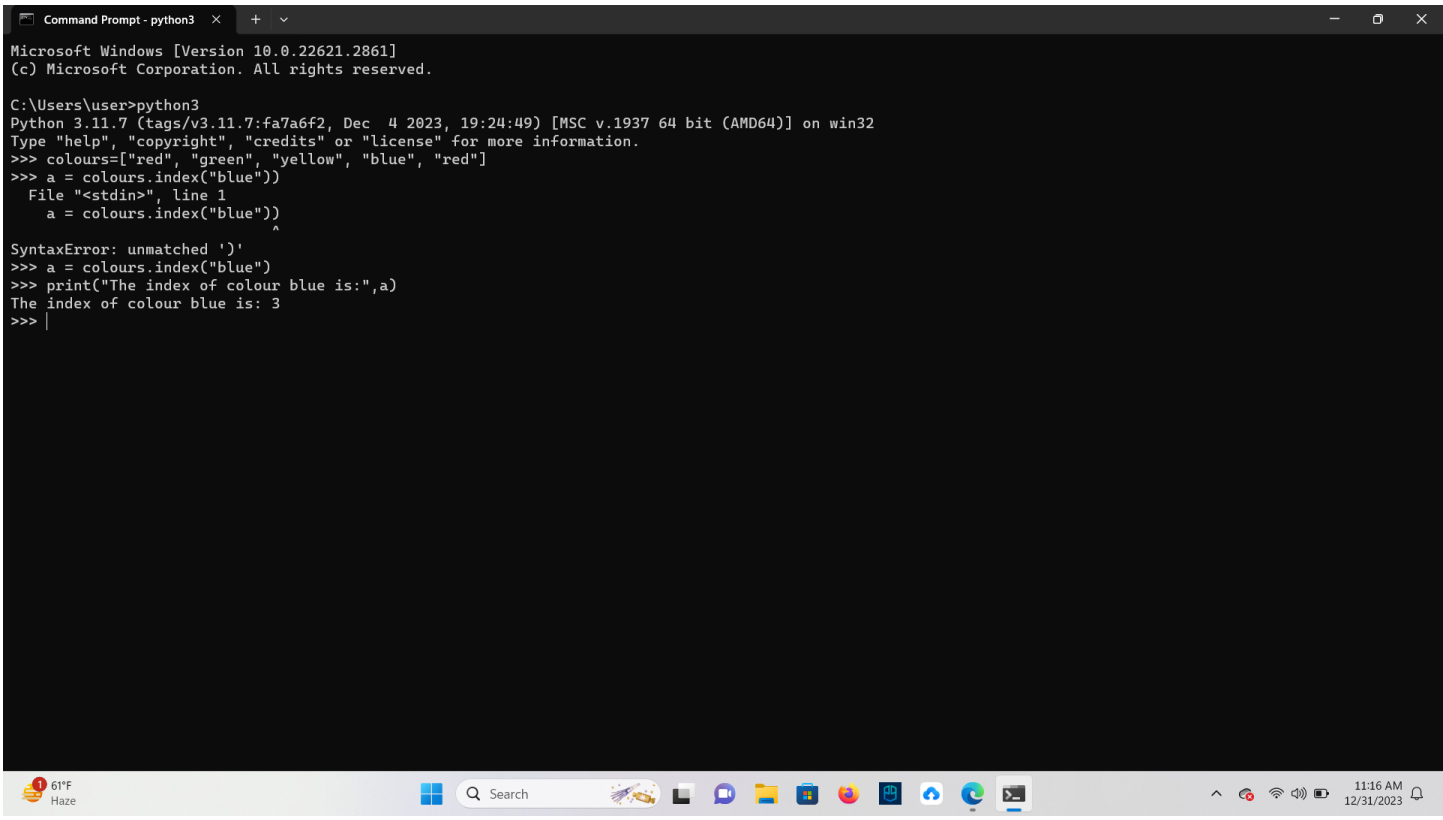
Ans



The index (), count (), and copy (), methods

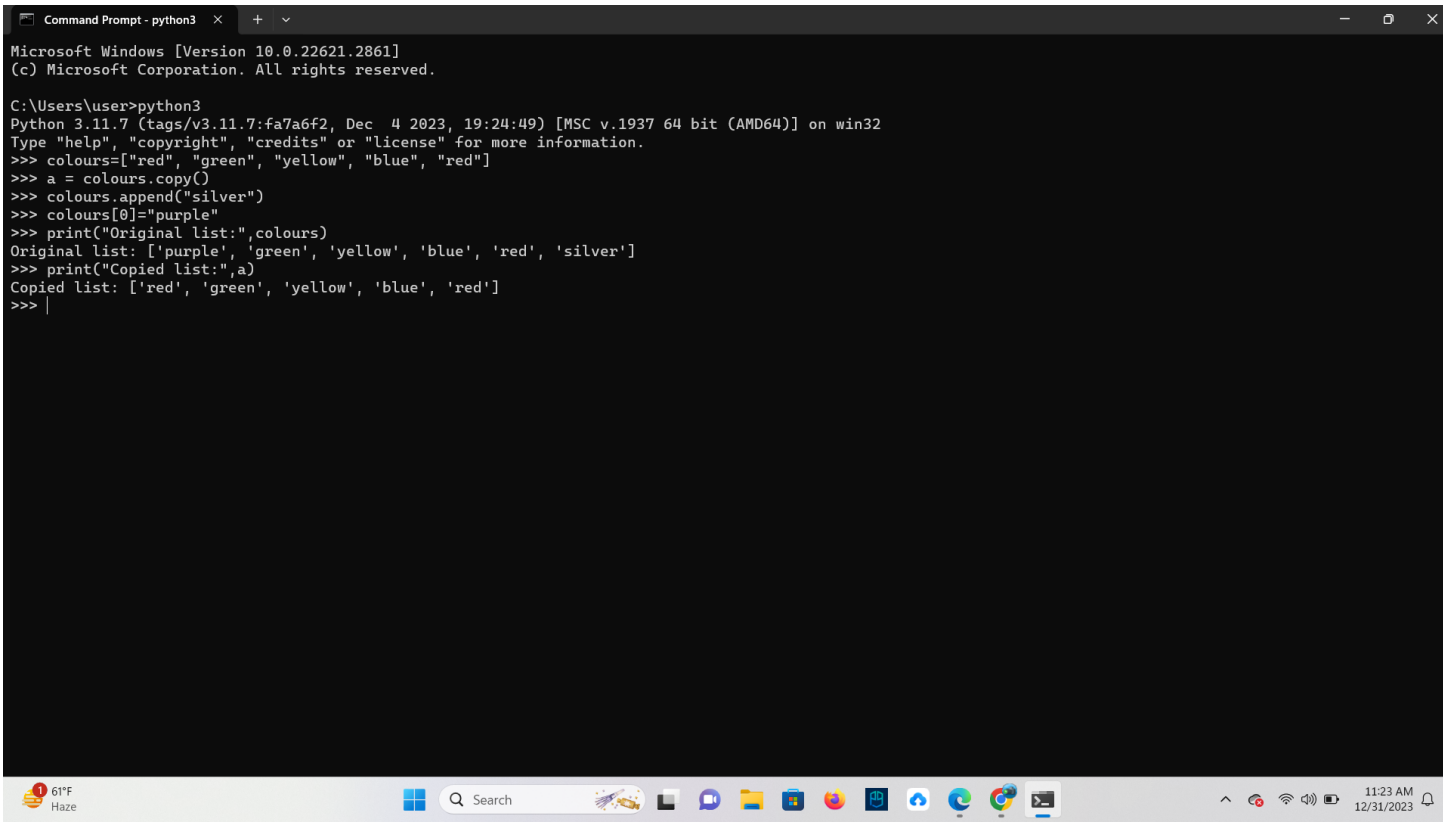
TASK: Write some code that finds the index of the colour blue.

Ans



TASK: Write some code that makes a copy of the colours using the copy () method. Then make some changes to the original list. Print the contents of the copied list to ensure these changes have not affected the copy.

Ans



The del Statement

The del statement is not specific to lists, but a general statement available within the Python language. It can be used to delete values from lists using both indexing and slicing. It can also be used to delete entire variables. Examples of its use are:

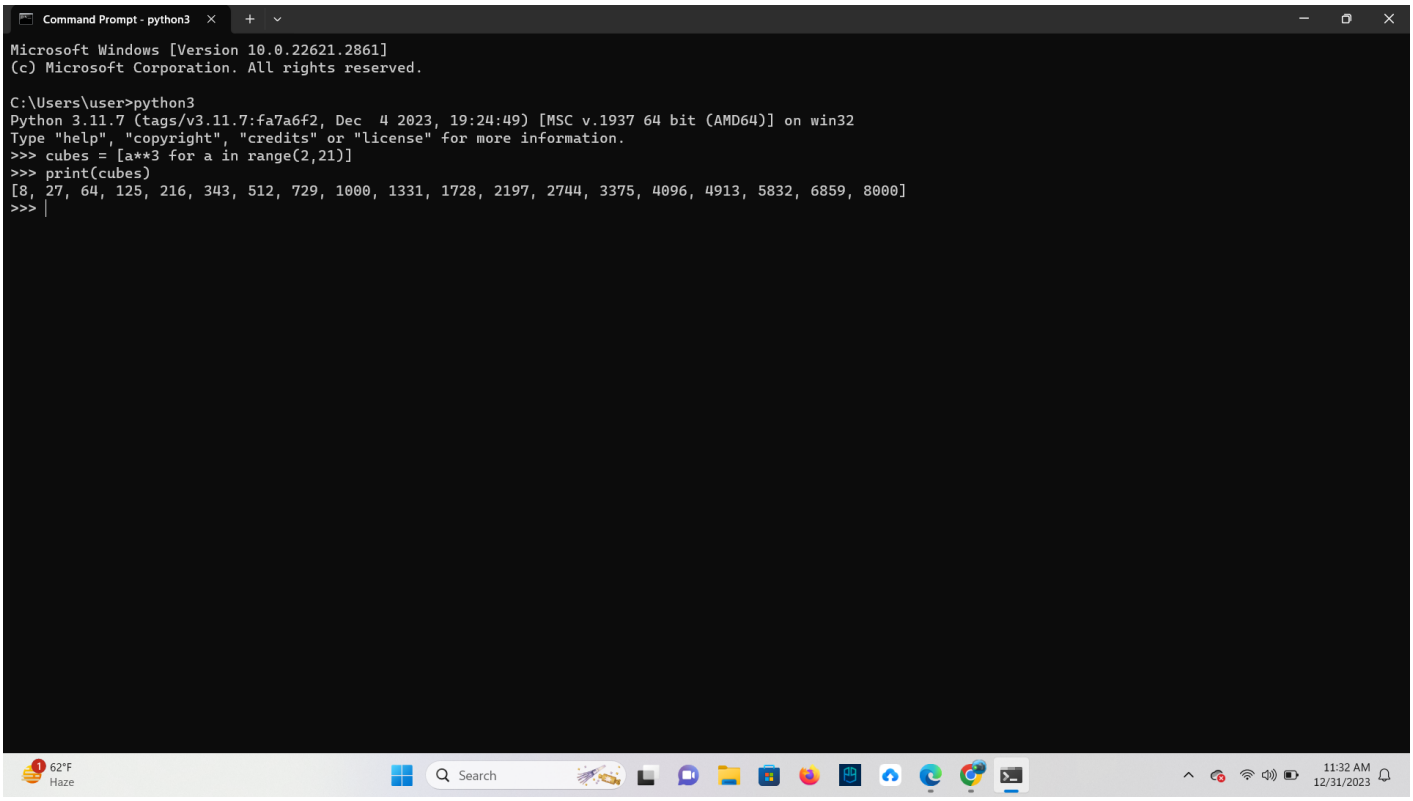
```
>>> del colours[0] # remove first colour
>>> del colours[-1:] # remove last colour
>>> del colours # remove the colour variable
```

The del statement is much more destructive than the mutator methods. Once a variable is deleted it cannot be accessed (unless recreated).

List Comprehensions

TASK: Write some code that uses a list comprehension to create a list called cubes that contains the cubed values (x * x * x) of the numbers from 2 to 20 inclusive.

Ans




```
some_users = [u for u in all_users if len(u) < 8]
```

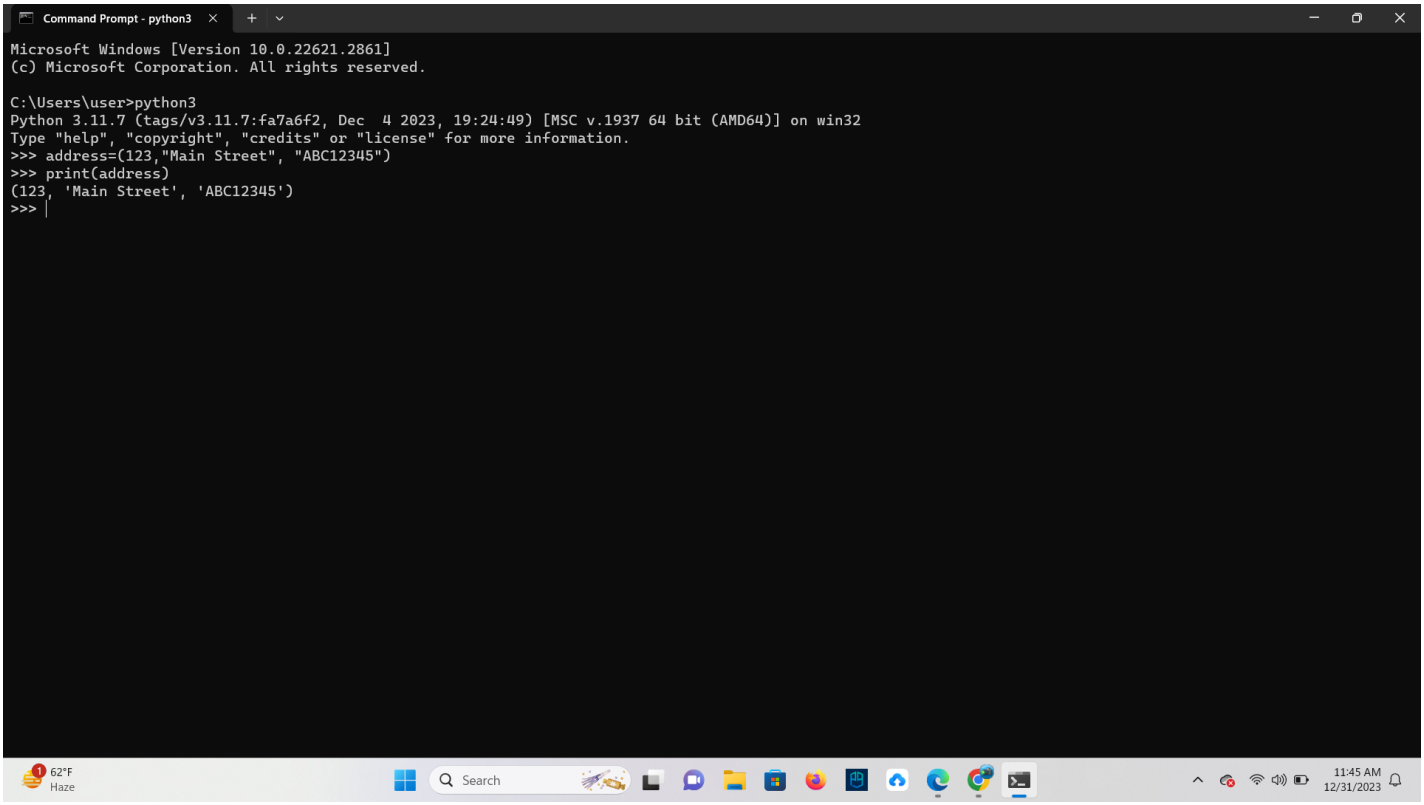
TASK: Examine the above code and work out which user names will be placed in the some_users list. What is the condition that has to be met for inclusion?

Ans So, the condition that has to be met for inclusion is that the length of the user name (**u**) must be less than 8 characters

Introduction to Tuples

TASK: Create a tuple called address that includes your own “house number”, “street” and “postcode” as three different values.

Ans



Empty and Single element Tuples

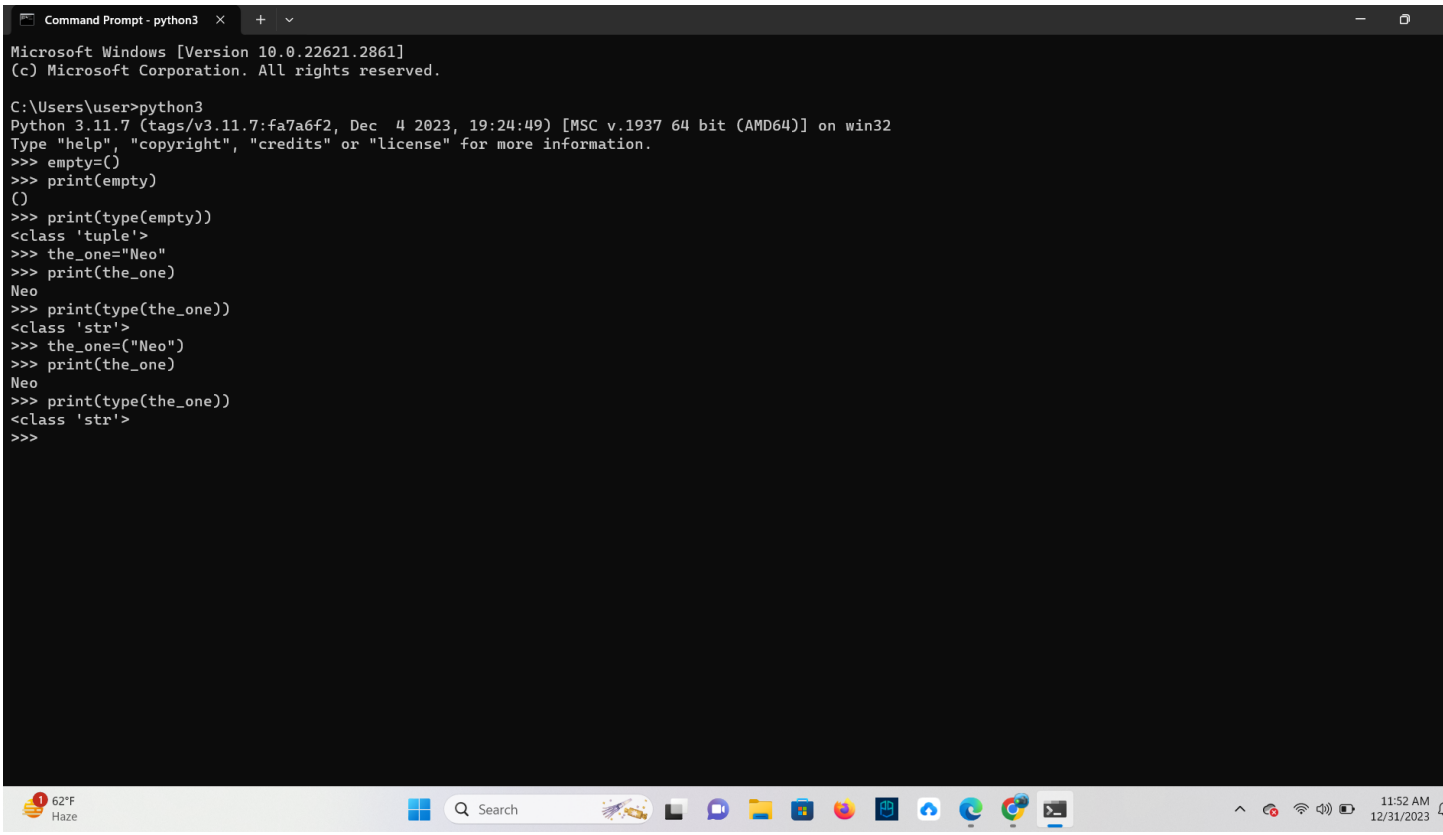
empty = ()

the_one = "Neo",

the_one = ("Neo")

TASK: Try entering the above examples to create single element tuples. Then use the type () function to examine the data-type of the created variables.

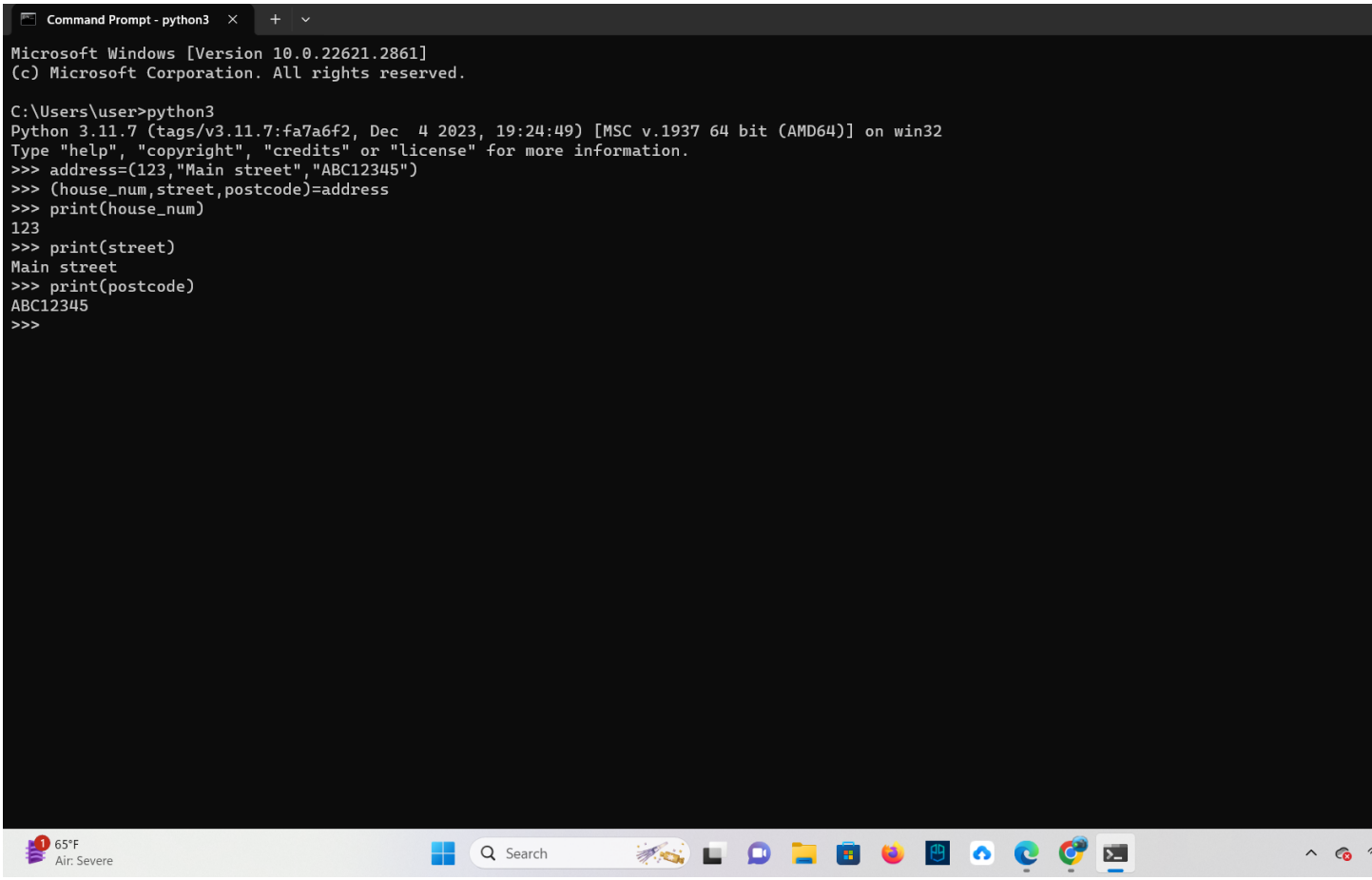
Ans



Sequence Unpacking

TASK: Use sequence unpacking to extract the values you stored within the address tuple earlier. Unpack the tuple into variables named house_num, street and postcode.

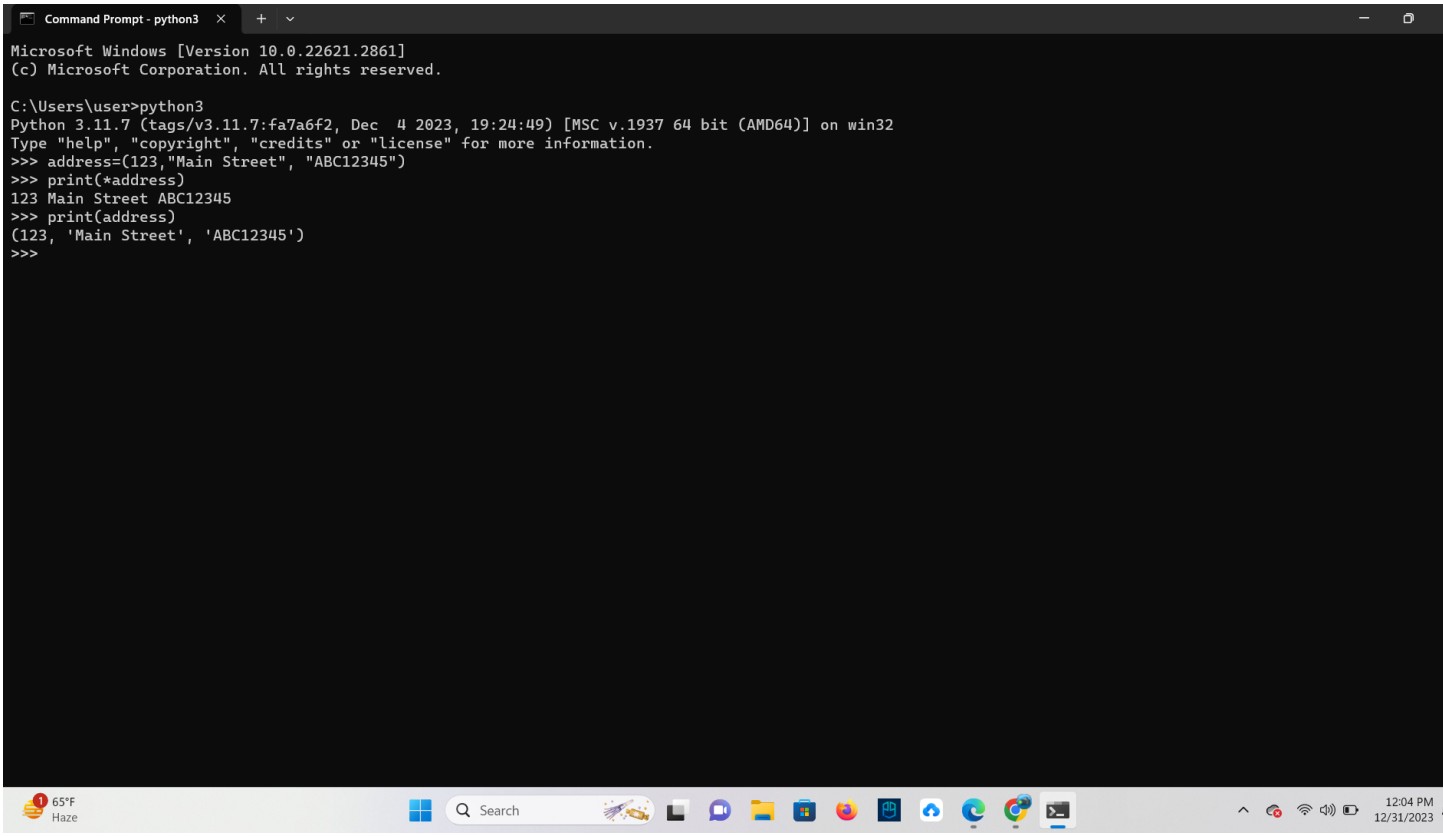
Ans



TASK: Write some code that calls the print () function to output the contents of the address tuple you created earlier. Ensure you use the '*' prefix so that the elements are extracted before being

passed to the function. Compare this with a version of the same code that calls the print () function without using the '*' prefix,

Ans



Tuple Methods

Since tuples are immutable, they have fewer methods than the List type. The reason is simply that methods that change the content are not applicable, so methods such as append (), extend () and insert () do not exist since the tuple content cannot be changed after creation.

In fact, the only methods that are available are count () and index (), which are accessors. These work in exactly the same way as they do with the list type. Also, since tuples are immutable it is not possible to assign to indexes or slices. e.g. the following code is not possible on tuples -

student [0] = "Griffin, Brian

Key Terminology

TASK: Look at each of the phrases below and ensure you understand what each of these means. For any that you do not understand, do a little research to find a definition of each term. This research may involve looking back over these notes, or the associated lecture notes. It may also involve searching for these terms on the Internet.

● Method

Ans A method is a function that “belongs to” an object.

● List comprehension

Ans List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

● Tuple

Ans A tuple is one of 4 built-in data types which is ordered and unchangeable and written with round brackets.

● Tuple Packing

Ans In Python, when we create a tuple, we assign a value to it. This is called "Packing".

● Sequence Unpacking

Ans It refers to the process of extracting individual elements from a sequence (like a tuple or a list).