

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

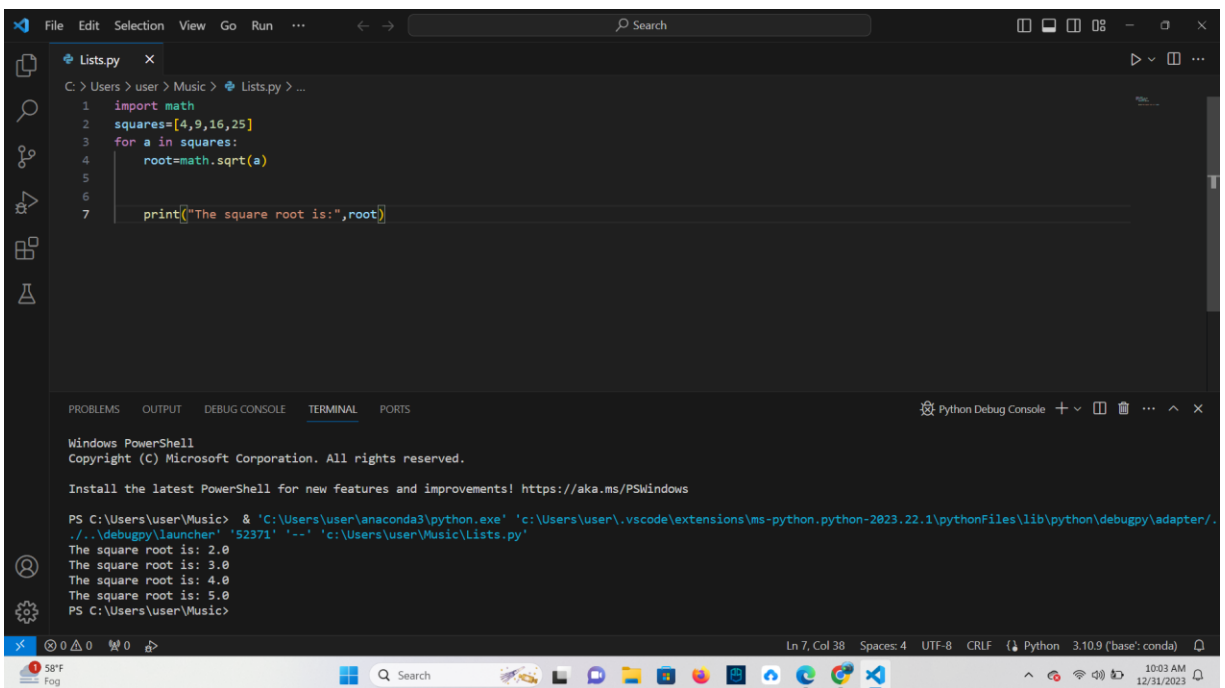
Week 6

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



The screenshot shows a Visual Studio Code editor window with a file named `Lists.py` open. The code in the editor is as follows:

```
1 import math
2 squares=[4,9,16,25]
3 for a in squares:
4     root=math.sqrt(a)
5
6
7 print("The square root is:",root)
```

Below the editor, the `TERMINAL` panel is active, showing the output of running the script. The output is:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

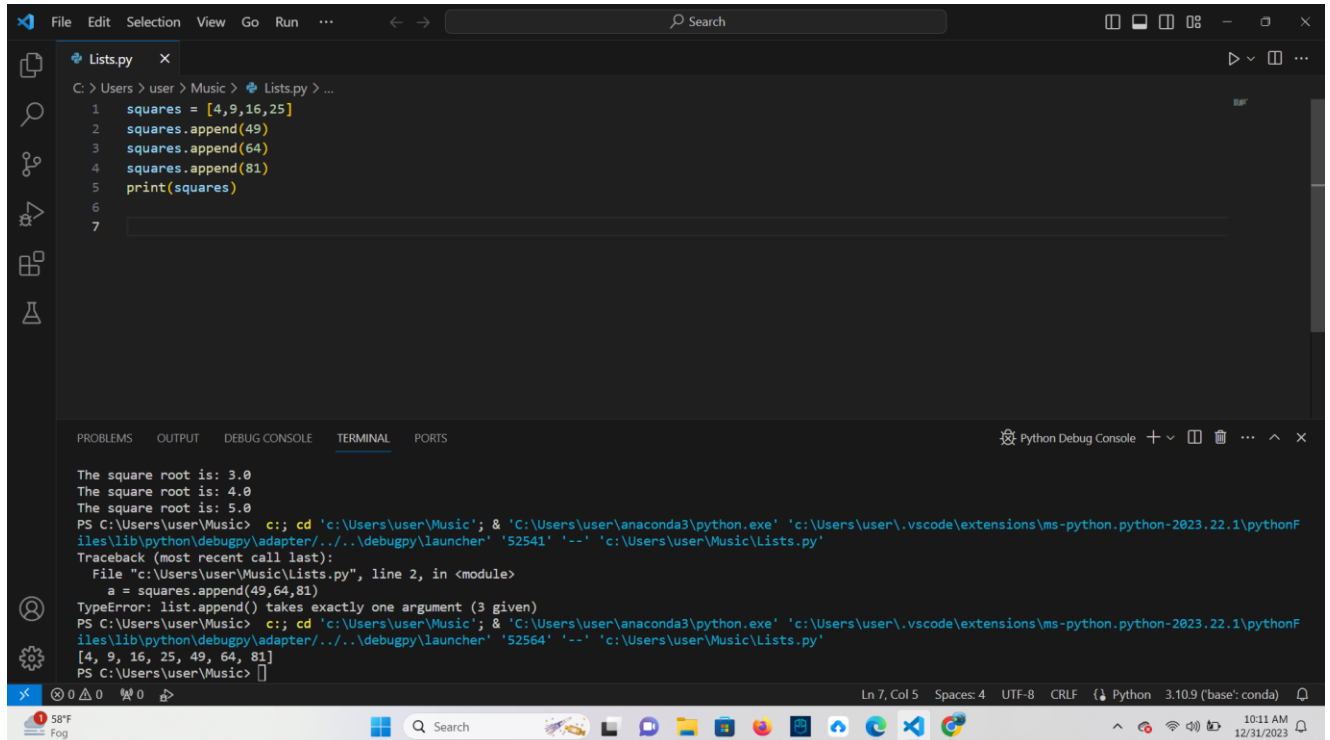
PS C:\Users\User\Music> & 'C:\Users\User\anaconda3\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.python-2023.22.1\pythonFiles\lib\python\debugpy\adapter\
..\debugpy\launcher' '52371' '--' 'c:\Users\User\Music\Lists.py'
The square root is: 2.0
The square root is: 3.0
The square root is: 4.0
The square root is: 5.0
PS C:\Users\User\Music>
```

The status bar at the bottom of the editor indicates the current position is `Ln 7, Col 38`, with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the Python interpreter is set to `Python 3.10.9 (base: conda)`. The system tray at the bottom shows the date and time as `10:03 AM 12/31/2023`.

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 screenshot shows a VS Code editor with a file named `Lists.py` open. The code in the editor is as follows:

```
1 squares = [4,9,16,25]
2 squares.append(49)
3 squares.append(64)
4 squares.append(81)
5 print(squares)
6
7
```

The terminal output shows the execution of the script, which prints the list `[4, 9, 16, 25, 49, 64, 81]`. The terminal also shows a traceback for a `TypeError` that occurred when the `append()` method was called with multiple arguments. The error message is:

```
TypeError: list.append() takes exactly one argument (3 given)
PS C:\Users\user\Music> c:; cd 'c:\Users\user\Music'; & 'C:\Users\user\anaconda3\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2023.22.1\pythonF
iles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '52541' '---' 'c:\Users\user\Music\Lists.py'
Traceback (most recent call last):
  File "c:\Users\user\Music\Lists.py", line 2, in <module>
    a = squares.append(49,64,81)
TypeError: list.append() takes exactly one argument (3 given)
PS C:\Users\user\Music> c:; cd 'c:\Users\user\Music'; & 'C:\Users\user\anaconda3\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2023.22.1\pythonF
iles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '52564' '---' 'c:\Users\user\Music\Lists.py'
[4, 9, 16, 25, 49, 64, 81]
PS C:\Users\user\Music>
```

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

```
Command Prompt - python3 x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> squares = [4, 9, 16, 25]
>>> three = [121, 144, 169]
>>> squares.extend(three)
>>> print(squares)
[4, 9, 16, 25, 121, 144, 169]
>>> |
```

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

```
Command Prompt - python3 x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> squares = [4, 9, 16, 25]
>>> squares.insert(0,2)
>>> print(squares)
[2, 4, 9, 16, 25]
>>> |
```

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

```
Command Prompt - python3 x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> squares = [4, 9, 16, 25]
>>> squares.remove(4)
>>> squares.remove(9)
>>> print(squares)
[16, 25]
>>> |
```

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

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> squares = [4, 9, 16, 25]
>>> squares.remove(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: list.remove(x): x not in list
>>>
```

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

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> squares = [1, 2, 3, 1, 2]
>>> squares.remove(2)
>>> print(squares)
[1, 3, 1, 2]
>>> |
```

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


```
Command Prompt - python3 x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> squares = [4, 9, 16, 25]
>>> a = squares.pop()
>>> print(a)
25
>>> print(squares)
[4, 9, 16]
>>> |
```

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

```
Command Prompt - python3 x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> squares = [4, 9, 16, 25]
>>> a = squares.pop(0)
>>> print(a)
4
>>> print(squares)
[9, 16, 25]
>>>
```

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.

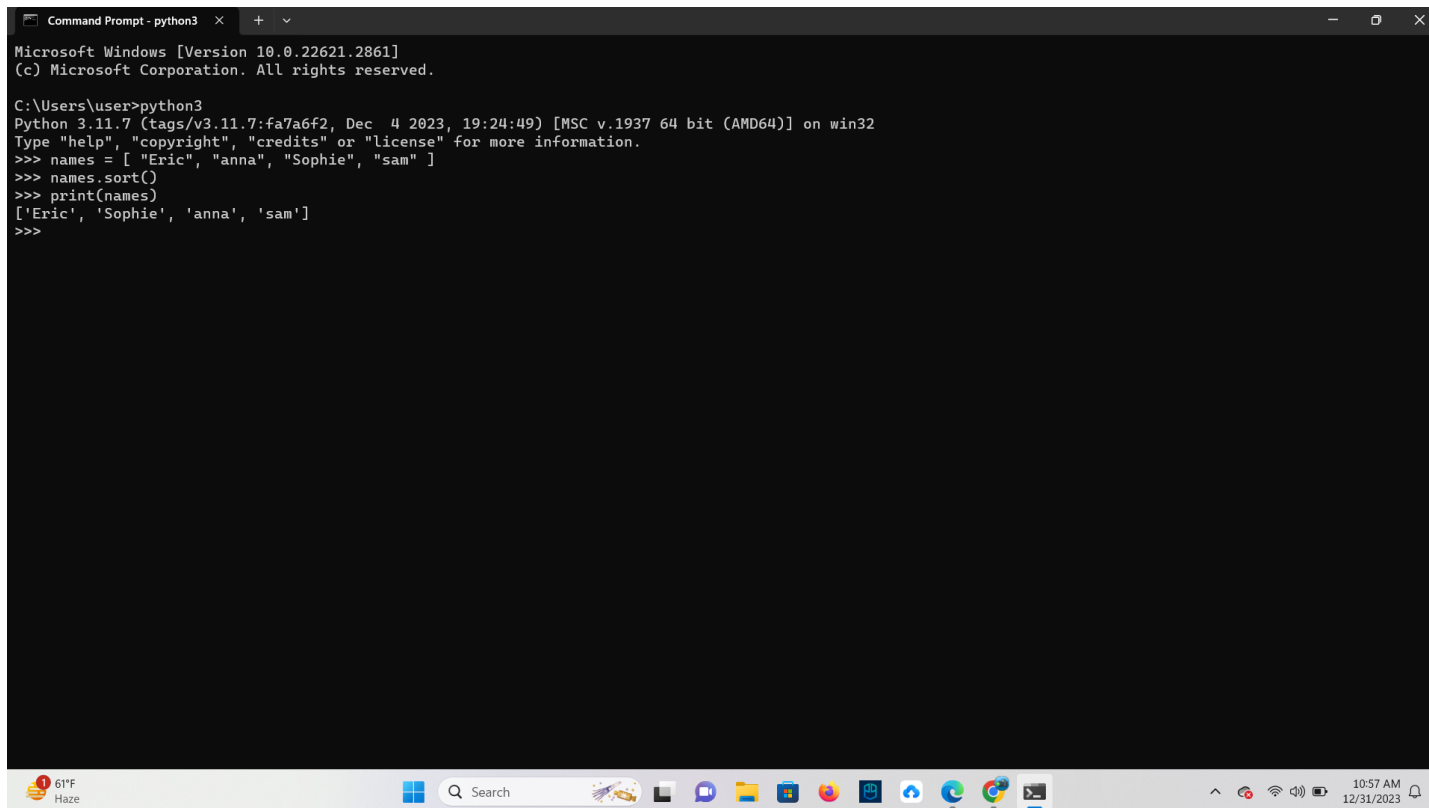
```
names = [ "Eric", "anna", "Sophie", "sam"]
```

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

A screenshot of a Windows Command Prompt window titled "Command Prompt - python3". The window shows the execution of a Python script. The output of the script is displayed, showing the sorted list of names. The taskbar at the bottom shows the system clock as 10:57 AM on 12/31/2023.

```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> names = [ "Eric", "anna", "Sophie", "sam" ]
>>> names.sort()
>>> print(names)
['Eric', 'Sophie', 'anna', 'sam']
>>>
```

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

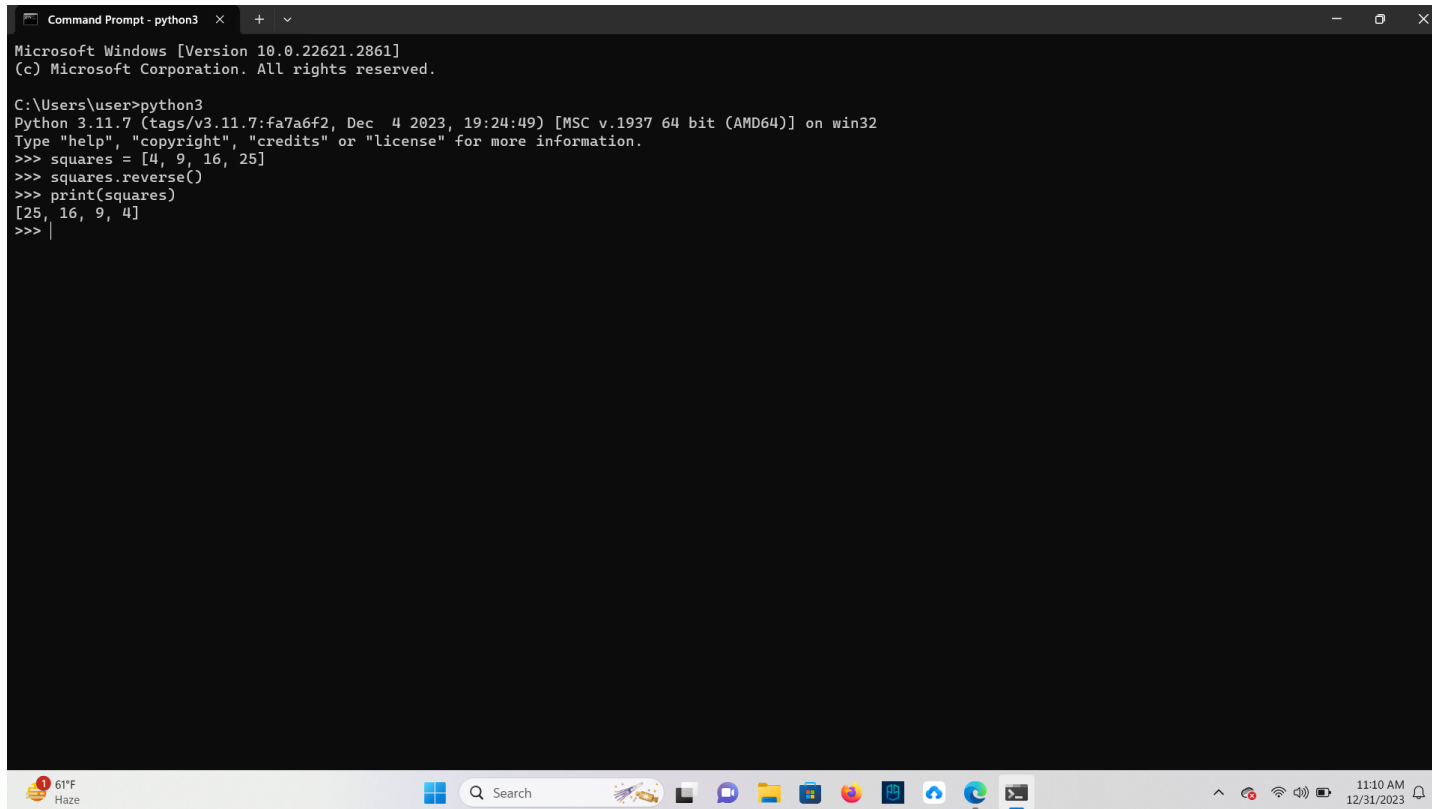
```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> names = [ "Eric", "anna", "Sophie", "sam" ]
>>> names.sort(key=lambda x: x.upper())
>>> print(names)
['anna', 'Eric', 'sam', 'Sophie']
>>> |
```

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

A screenshot of a Windows Command Prompt window titled "Command Prompt - python3". The window shows the execution of a Python script. The output includes the Windows version (10.0.22621.2861), the Python version (3.11.7), and the execution of the code: squares = [4, 9, 16, 25], squares.reverse(), and print(squares), resulting in the output [25, 16, 9, 4]. The taskbar at the bottom shows the date and time as 11:10 AM on 12/31/2023.

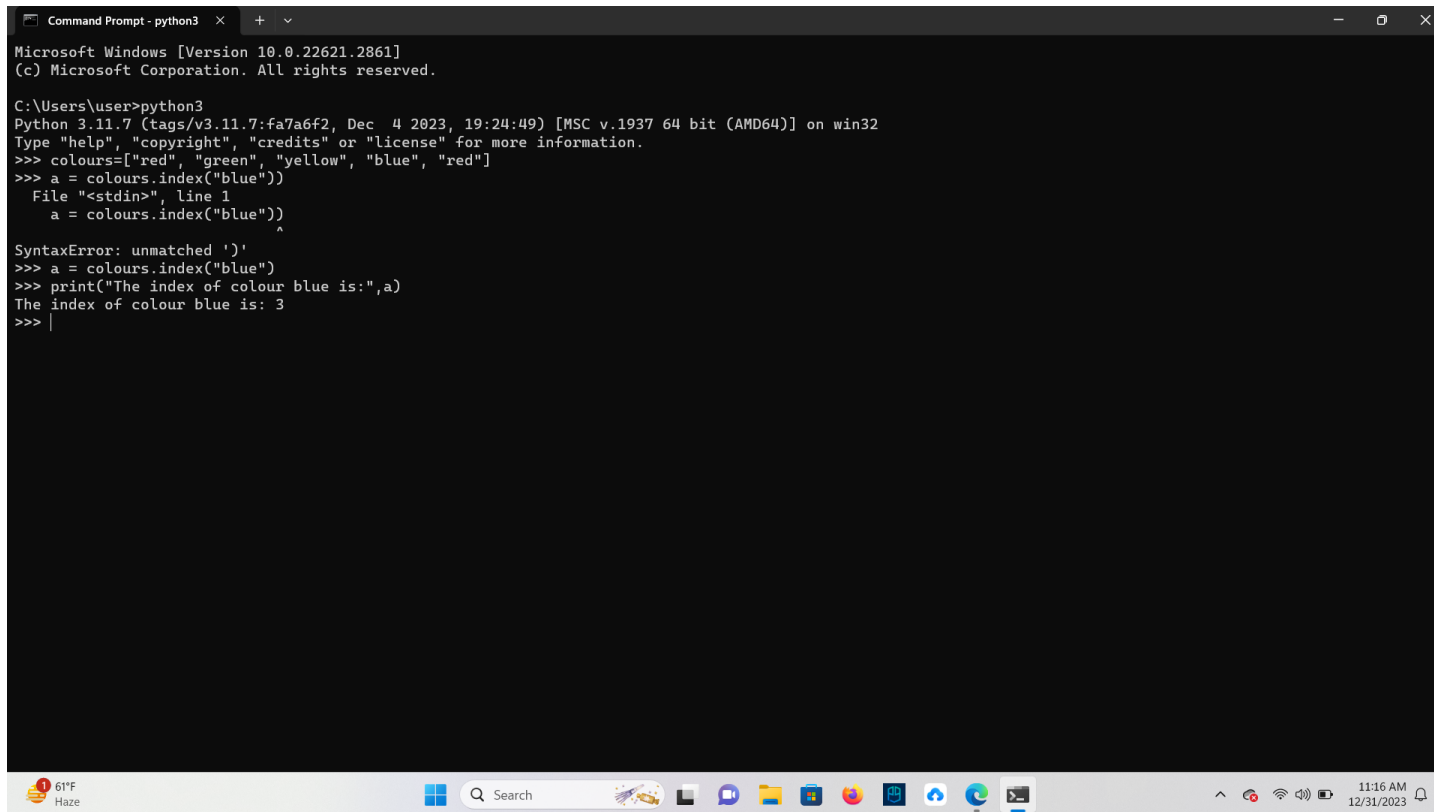
```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> squares = [4, 9, 16, 25]
>>> squares.reverse()
>>> print(squares)
[25, 16, 9, 4]
>>> |
```

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

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

Ans

A screenshot of a Windows Command Prompt window titled "Command Prompt - python3". The window shows the execution of a Python script. The script defines a list of colors: ["red", "green", "yellow", "blue", "red"] and attempts to find the index of the color "blue". The output shows the index is 3. The window also displays the Windows taskbar at the bottom with various icons and the system clock showing 11:16 AM on 12/31/2023.

```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> colours=["red", "green", "yellow", "blue", "red"]
>>> a = colours.index("blue")
>>> File "<stdin>", line 1
      a = colours.index("blue")
                                ^
SyntaxError: unmatched ')
>>> a = colours.index("blue")
>>> print("The index of colour blue is:",a)
The index of colour blue is: 3
>>> |
```

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

```
Command Prompt - python3 x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> colours=["red", "green", "yellow", "blue", "red"]
>>> a = colours.copy()
>>> colours.append("silver")
>>> colours[0]="purple"
>>> print("Original list:",colours)
Original list: ['purple', 'green', 'yellow', 'blue', 'red', 'silver']
>>> print("Copied list:",a)
Copied list: ['red', 'green', 'yellow', 'blue', 'red']
>>> |
```

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

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> cubes = [a**3 for a in range(2,21)]
>>> print(cubes)
[8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728, 2197, 2744, 3375, 4096, 4913, 5832, 6859, 8000]
>>>
```

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

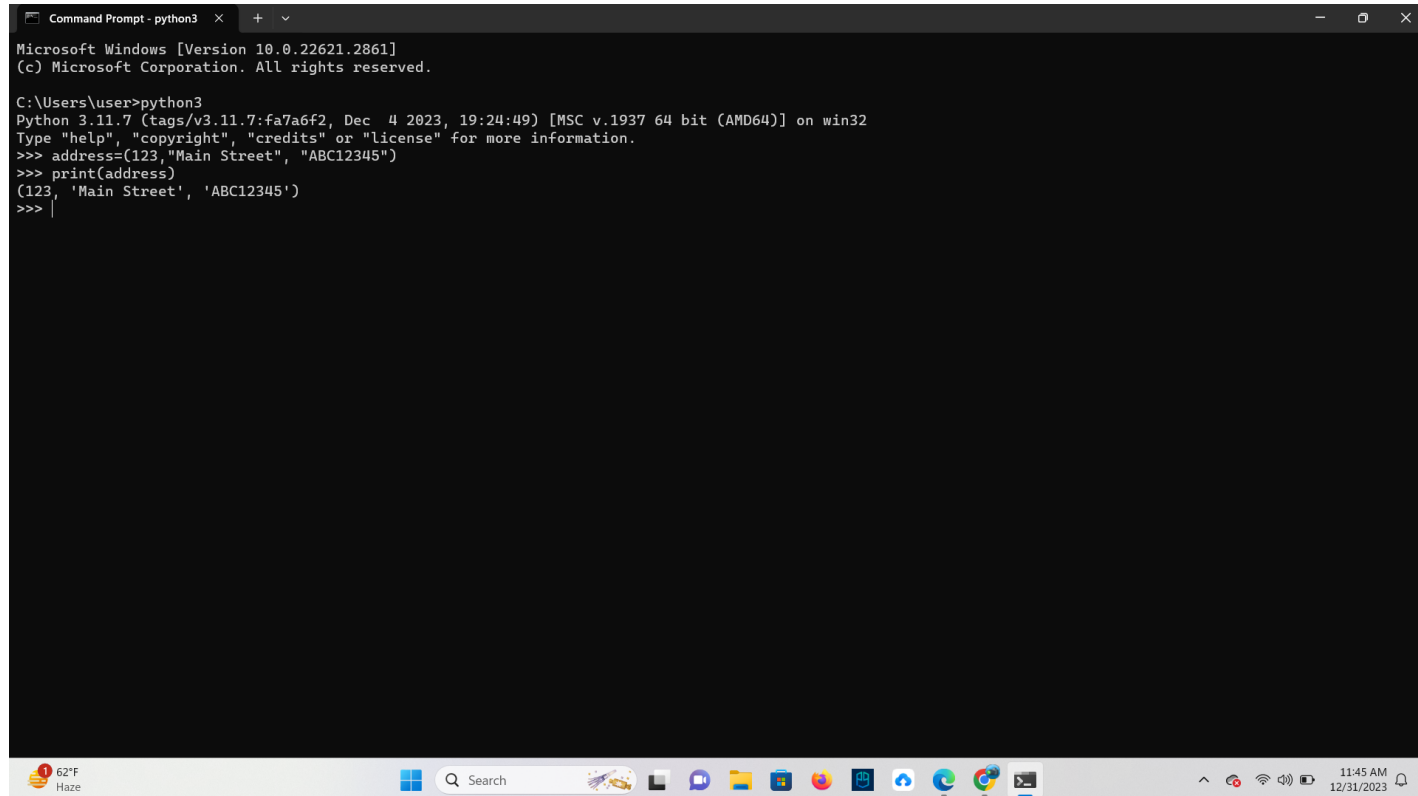
TASK: Examine the above code and work out which user names will be placed in the some_users list. What is the condition that has to be met for inclusion?

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

A screenshot of a Windows Command Prompt window titled "Command Prompt - python3". The window shows the following text:

```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> address=(123,"Main Street", "ABC12345")
>>> print(address)
(123, 'Main Street', 'ABC12345')
>>> |
```

The window has a taskbar at the bottom showing the Windows logo, a search bar, and various application icons. The system tray on the right shows the date and time as 11:45 AM on 12/31/2023.

Empty and Single element Tuples

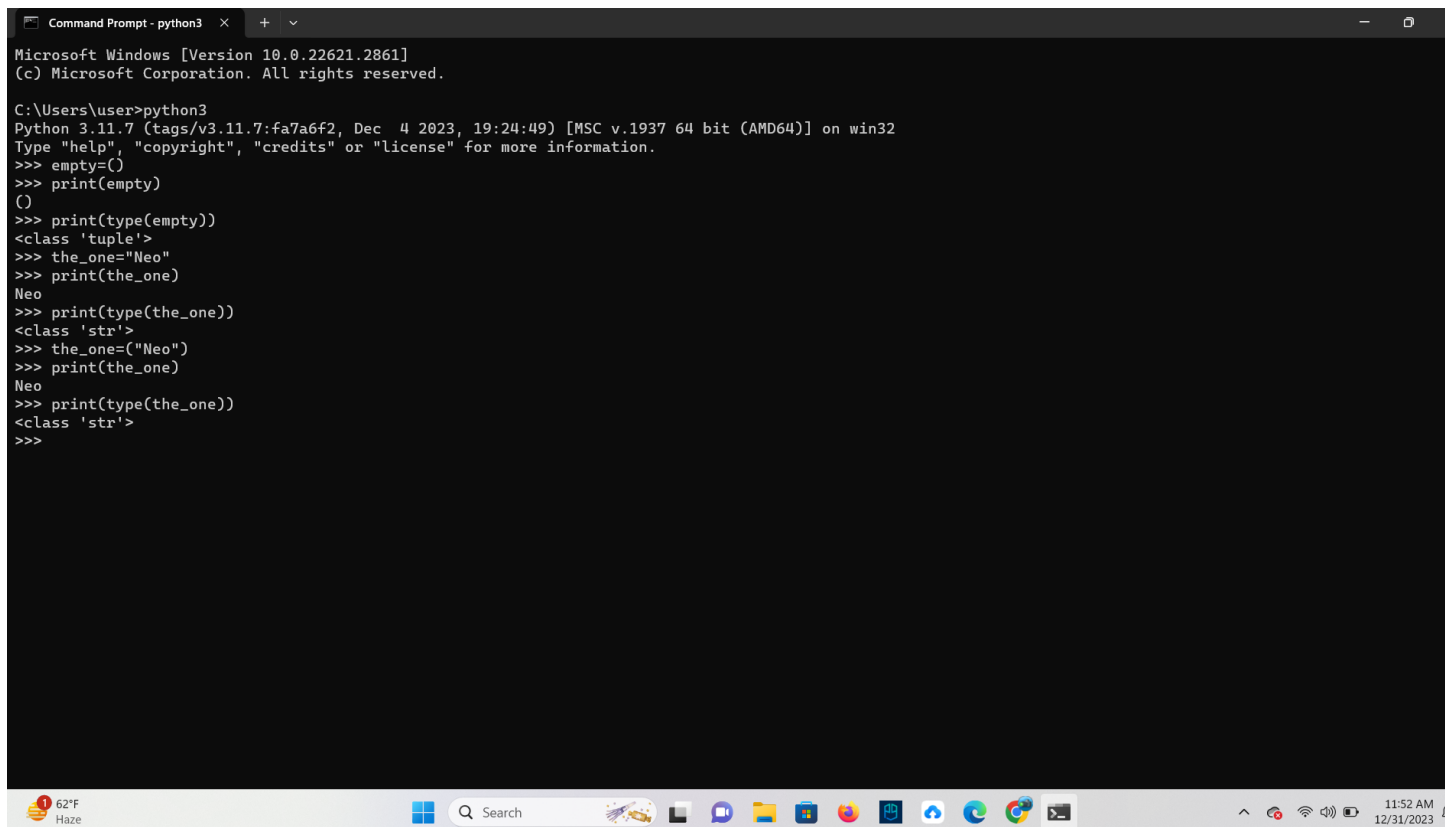
```
empty = ()
```

```
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



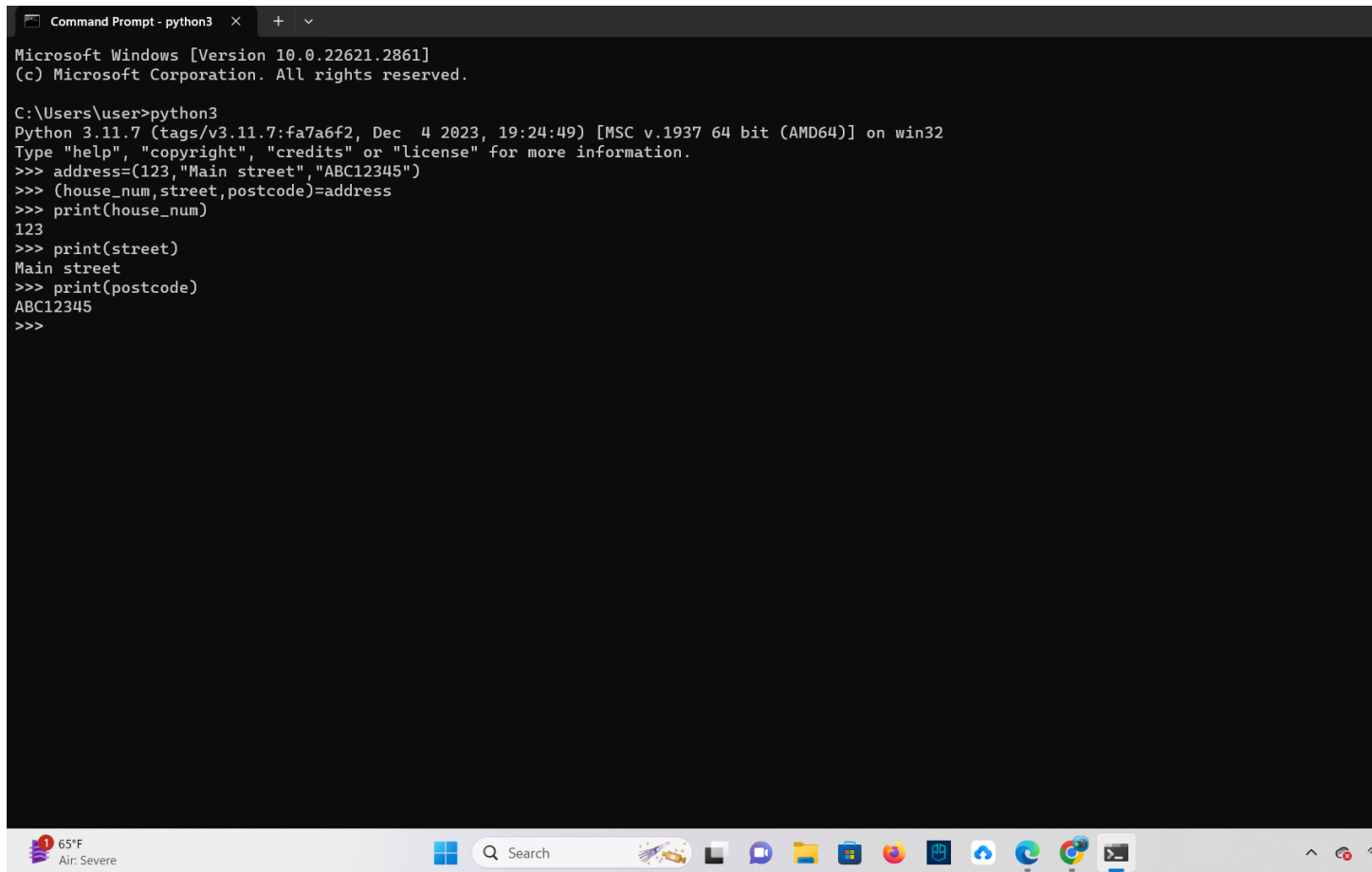
```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> empty=()
>>> print(empty)
()
>>> print(type(empty))
<class 'tuple'>
>>> the_one="Neo"
>>> print(the_one)
Neo
>>> print(type(the_one))
<class 'str'>
>>> the_one=("Neo")
>>> print(the_one)
Neo
>>> print(type(the_one))
<class 'str'>
>>>
```

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



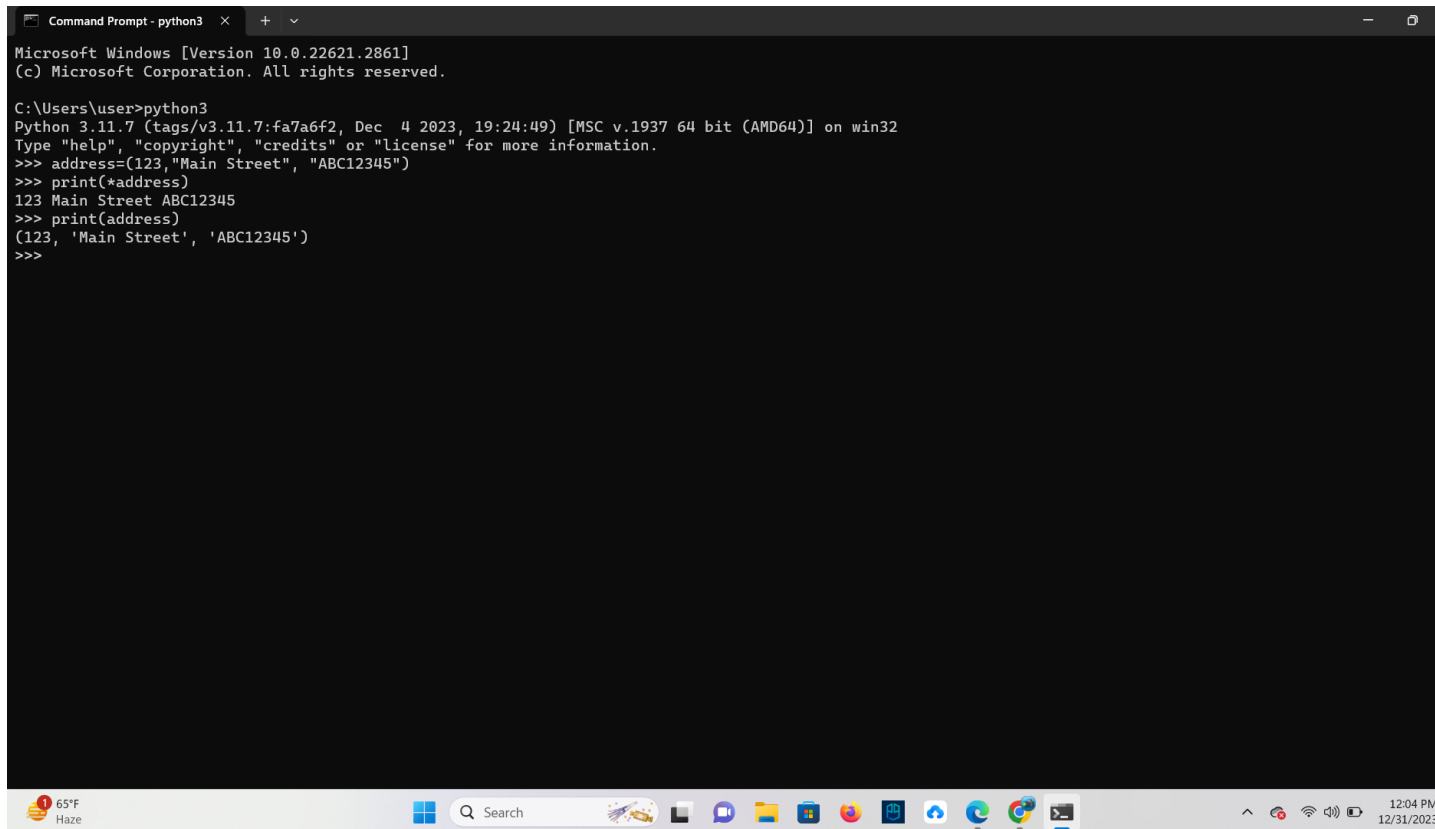
```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> address=(123,"Main street","ABC12345")
>>> (house_num,street,postcode)=address
>>> print(house_num)
123
>>> print(street)
Main street
>>> print(postcode)
ABC12345
>>>
```

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



```
Command Prompt - python3
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec 4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> address=(123,"Main Street", "ABC12345")
>>> print(*address)
123 Main Street ABC12345
>>> print(address)
(123, 'Main Street', 'ABC12345')
>>>
```

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).