* What is interpreter?

Interpreter can be termed as system software which has the capability to read and execute the program, rather you can say interpret programs. This interpretation includes the source code of high-level language, programs that are pre-compiled as well as scripts. It is to be noted that, interpreter interprets program line-by-line, which means it translates one statement at a single go. This feature makes easy for programmers to check any particular line at the time of debugging, but slows down the overall time of execution of the entire program.

An interpreter is a computer program that is used to directly execute program instructions written using one of the many high-level programming languages.

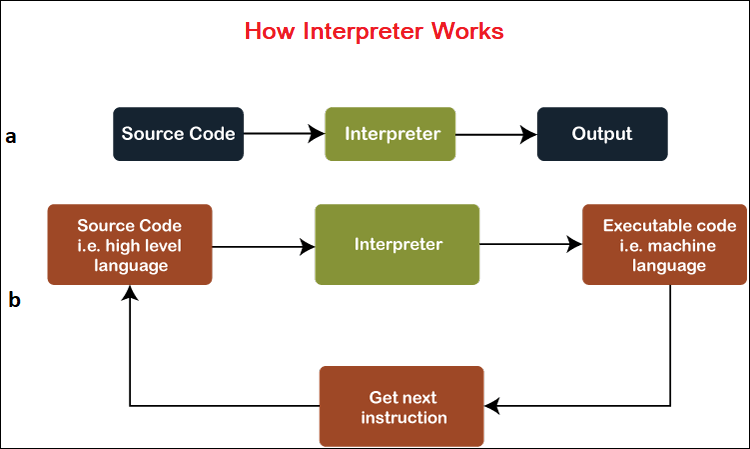
The interpreter transforms the high-level program into an intermediate language that it then executes, or it could parse the high-level source code and then performs the commands directly, which is done line by line or statement by statement.

* What is compiler?

A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. Typically, a programmer writes language statements in a language such as Pascal or C one line at a time using an editor. The file that is created contains what are called the source statements. The programmer then runs the appropriate language compiler, specifying the name of the file that contains the source statements.

* How interpreter works?

An interpreter is also a program that translates a high-level language into a low-level one, but it does it at the moment the program is run. You write the program using a text editor or something similar, and then instruct the interpreter to run the program. It takes the program, one line at a time, and translates each line before running it: It translates the first line and runs it, then translates the second line and runs it etc.



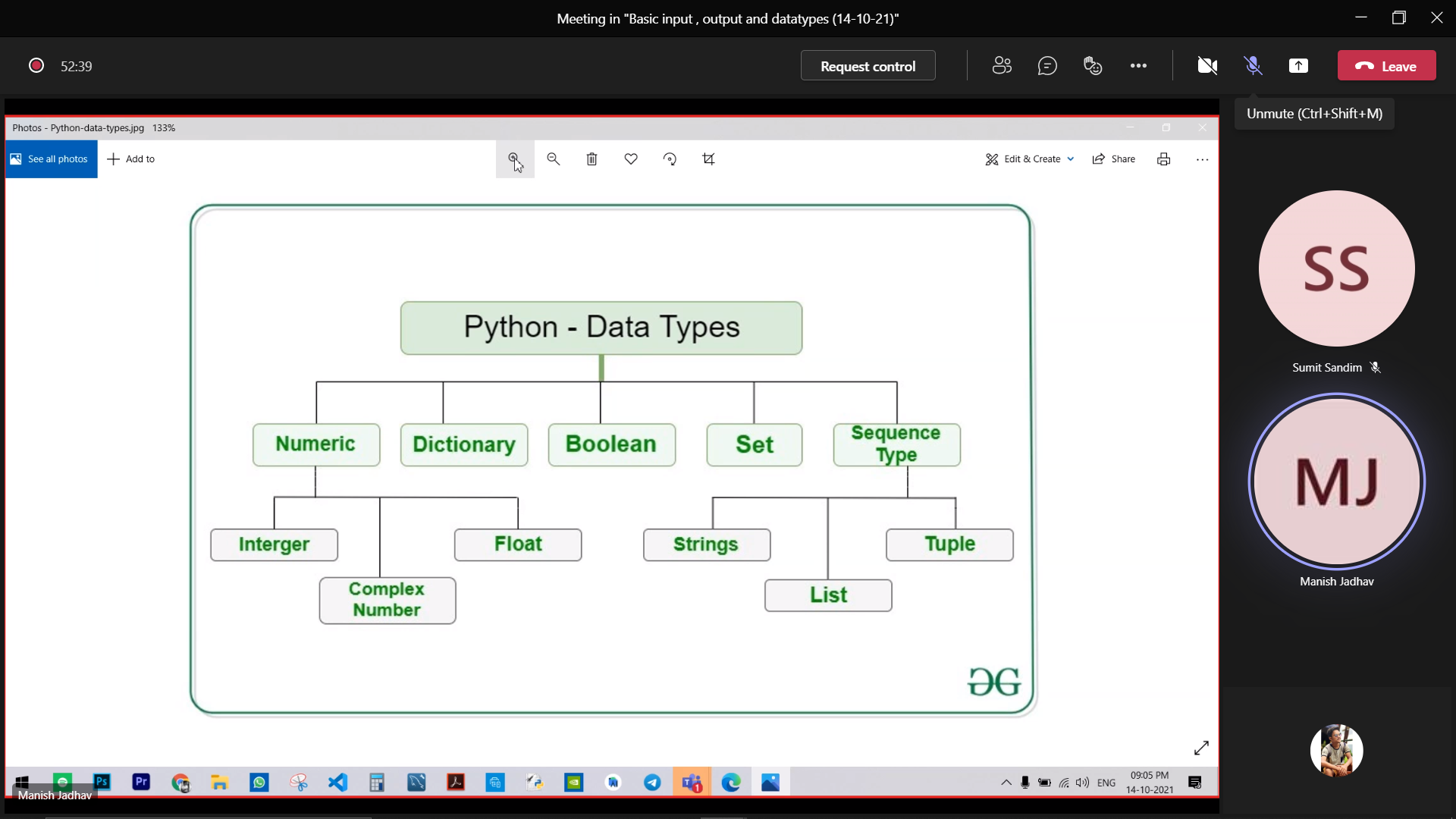
* difference between compiler and interpreter?
* Interpreter:

An interpreter is a program that translates a programming language into a comprehensible language.

* It translates only one statement of the program at a time.
* Interpreters, more often than not are smaller than compilers.
* Compiler:

It is a translator which takes input i.e., High-Level Language, and produces an output of low-level language i.e. machine or assembly language.

* A compiler is more intelligent than an assembler it checks all kinds of limits, ranges, errors, etc.
* But its program run time is more and occupies a larger part of memory. It has slow speed because a compiler goes through the entire program and then translates the entire program into machine codes.
* Datatypes in Python



### **1. Python Numbers**

We can find complex numbers, floating point numbers and integers in the category of Python Numbers. Complex numbers are defined as a complex class, floating point numbers are defined as float and integers are defined as an int in Python. There is one more type of datatype in this category, and that is long. It is used to hold longer integers. One will find this datatype only in Python 2.x which was later removed in Python 3.x.

“Type()” function is used to know the class of a value or variable. To check the value for a particular class, “isinstance()” function is used.

* **Integers:**
  + There is no maximum limit on the value of an integer. The integer can be of any length without any limitation which can go up to the maximum available memory of the system.
* **Integers can look like this:**
  + >>> print(123123123123123123123123123123123123123123123123123 + 1)

123123123123123123123123123123123123123123123123124

* **Floating Point Number:**
  + The difference between floating points and integers is decimal points. Floating point number can be represented as “1.0”, and integer can be represented as “1”. It is accurate up to 15 decimal places.
* **Complex Number:**
  + “x + yj” is the written form of the complex number. Here y is the imaginary part and x is the real part.

### **2. Python List**

An ordered sequence of items is called List. It is a very flexible data type in Python. There is no need for the value in the list to be of the same data type. The List is the data type that is highly used data type in Python. List datatype is the most exclusive datatype in Python for containing versatile data. It can easily hold different types of data in Python.

It is effortless to declare a list. The list is enclosed with brackets and commas are used to separate the items.

A list can look like this:

>>> a = [5,9.9,’list’]

One can also alter the value of an element in the list.

### **3. Python Tuple**

A Tuple is a sequence of items that are in order, and it is not possible to modify the Tuples. The main difference list and tuples are that tuple is immutable, which means it cannot be altered. Tuples are generally faster than the list data type in Python because it cannot be changed or modified like list datatype. The primary use of Tuples is to write-protect data. Tuples can be represented by using parentheses (), and commas are used to separate the items.

Tuples can look like this:

>>> t = (6,’tuple’,4+2r)

In the case of a tuple, one can use the slicing operator to extract the item, but it will not allow changing the value.

### **4. Python Strings**

A String is a sequence of Unicode characters. In Python, String is called str. Strings are represented by using Double quotes or single quotes. If the strings are multiple, then it can be denoted by the use of triple quotes “”” or ”’. All the characters between the quotes are items of the string.

One can put as many as the character they want with the only limitation being the memory resources of the machine system. Deletion or Updation of a string is not allowed in python programming language because it will cause an error. Thus, the modification of strings is not supported in the python programming language.

A string can look like this:

>>> s = “Python String”

>>> s = ”’a multi-string

Strings are also immutable like tuples and items can be extracted using slicing operators [].

If one wants to represent something in the string using quotes, then they will need to use other types of quotes to define the string in the beginning and the ending.

Such as:

>>> print(“This string contains a single quote (‘) character.”)

This string contains a single quote (‘) character.

>>> print(‘This string contains a double quote (“) character.’)

This string contains a double quote (“) character.

### **5. Python Set**

The Collection of Unique items that are not in order is called Set. Braces {} are used to defined set and a comma is used to separate values. One will find that the items are unordered in a set data type.

Duplicates are eliminated in a set and set only keeps unique values. Operations like intersection and union can be performed on two sets.

Python set will look like this:

>>> a = {4,5,5,6,6,6}

>>> a

{4, 5, 6}

### The slicing operator does not work on set because the set is not a collection of ordered items, and that is why there is no meaning to the indexing of set.**6. Python Dictionary**

Dictionary is a type of python data type in which collections are unordered, and values are in pairs called key-value pairs. This type of data type is useful when there is a high volume of data. One of the best functions of Dictionaries data type is retrieving the data for which it is optimized. The value can only be retrieved if one knows the key to retrieve it.

Braces {} (curly brackets) are used to define dictionaries data type in Python. A Pair in the dictionary data type is an item which is represented as key:value. The value and the key can be of any data type.

Python Dictionary can look like this:

>>> d = {3:’key’,4:’value’}

### **7. Boolean Type**

There can be only two types of value in the Boolean data type of Python, and that is True or False.

It can look like this:

>>> type(True)

<class ‘bool’>

>>> type(False)

<class ‘bool’>

The true value in the Boolean context is called “truthy”, and for false value in the Boolean context, it is called “false”. Truthy is defined by the objects in Boolean, which is equal to True, and in the same way, False is defined by the objects equal to false. One can also evaluate non-Boolean objects in a Boolean context.