**1. In the below elements which of them are values or an expression? eg:- values can be integer or string and expressions will be mathematical operators.**

\*        ---- - expression

'hello'  ----- value

-87.8    ------ value

-    ----------- expression

/       -------- expression

+   ----------- expression

6 ------------- value

**2. What is the difference between string and variable?**

**String**: String can be written in quotes such: "apple", 'apple'

**Variable**: It is the thing in which we can store data just like a container.

**3. Describe three different data types.**

1. **Numeric** - In Python, numeric data type represent the data which has numeric value. Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

* **Integer** - This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.
* **Float** - This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
* **Complex Number** - Complex number is represented by complex class. It is specified as (real part) + (imaginary part)j. For example – 2+3j

1. **Sequence Type:** In Python, sequence is the ordered collection of similar or different data types. Sequences allow storing multiple values in an organized and efficient fashion. There are several sequence types in Python –

## String - In Python Strings are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by str class.

#### Creating String

Strings in Python can be created using single quotes or double quotes or even triple quotes.

* + - **List** - Lists are just like the arrays, declared in other languages which is a ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

#### Creating List

Lists in Python can be created by just placing the sequence inside the square brackets [].

## Tuple - Just like list, tuple is also an ordered collection of Python objects. The only difference between tuple and list is that tuples are immutable i.e. tuples cannot be modified after it is created. It is represented by tuple class.

#### Creating Tuple

In Python, tuples are created by placing a sequence of values separated by ‘comma’ with or without the use of parentheses for grouping of the data sequence. Tuples can contain any number of elements and of any datatype (like strings, integers, list, etc.).

**Note**– Creation of Python tuple without the use of parentheses is known as Tuple Packing.

1. **Boolean** - Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.

**Note**– True and False with capital ‘T’ and ‘F’ are valid booleans otherwise python will throw an error.

1. **Set** - In Python, Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

#### Creating Sets

Sets can be created by using the built-in set() function with an iterable object or a sequence by placing the sequence inside curly braces, separated by ‘comma’. Type of elements in a set need not be the same, various mixed-up data type values can also be passed to the set.

1. **Dictionary** : in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

#### Creating Dictionary

In Python, a Dictionary can be created by placing a sequence of elements within curly {} braces, separated by ‘comma’. Values in a dictionary can be of any datatype and can be duplicated, whereas keys can’t be repeated and must be immutable. Dictionary can also be created by the built-in function dict(). An empty dictionary can be created by just placing it to curly braces{}.

**Note**– Dictionary keys are case sensitive, same name but different cases of Key will be treated distinctly.

**4. What is an expression made up of? What do all expressions do?**

Expression is made up of values, containers, and mathematical operators (operands)

e.g :    a = 2 + 3     and expressions are used to evaluate the values or represent the result on the screen.

**5. This assignment statements, like spam = 10. What is the difference between an expression and a statement?**

Expression is made up of values, containers, and mathematical operators (operands) and the statement is just like a command that a python interpreter executes like print.

**6. After running the following code, what does the variable bacon contain?**

bacon = 22

bacon + 1

Ans : it gives 23 as execution code

**7. What should the values of the following two terms be?**

'spam' + 'spamspam' - spamspamspam

'spam' \* 3 - spamspamspam

It will be the same.

**8. Why is eggs a valid variable name while 100 is invalid?**

Because we can't start giving variable an integer name. If we, we should begin with, a string-like alphabet name then integer. e100 or eggs100 is valid.

**9. What three functions can be used to get the integer, floating-point number, or string version of a value?**

ans: str(), int(), float()

**10. Why does this expression cause an error? How can you fix it?**

'I have eaten ' + 99 + ' burritos.'

sol:  because 99 is an integer it cannot be concatenated with strings, if we have to concatenate it we need to do typecasting.