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.

\* ---: multiplication operator [used in expression]

- ---: subtraction operator [used in expression]

/ ---: division operator [used in expression]

+ ---: addition operator [used in expression]

'hello' ---: string value

-87.8 ---: float value [decimals are called as floats]

6 ---: integer value

2. What is the difference between string and variable?

string represents a sequence of characters enclosed inside quotes. It is one of the datatype supported by python. variable is something that basically stores the value/information.

For example: first\_name = ‘Harry17’

Here,

1. **first\_name is a variable** that stores the value/information of string data type i.e, ‘Harry17’.
2. **‘Harry17’ is of string data type**

**Tip:** We use type() built-in function to get the data type of variable here in this case print(type(first\_name)) give is <class ‘str’> i.e, first\_name is of string data type.

3. Describe three different data types.

**String :** string is a collection of characters enclosed inside quotes (‘, ‘’, ‘’’). It belongs to <class 'str'> (i.e, string class). This class <class 'str'> has lots of inbuilt methods that operates on string object.

Example: ‘Harry @ 1 is a string’

**Integer:** integer consists of positive or negative whole numbers including 0. It belongs to <class ‘int’> (i.e, integer class).

Example: -1 , 0, 1 are integers

-5.5 or 5.5 or 0.1 are not integers they are called as floats

**List:** list is a ordered collection of items (aka values) that belongs to single or multiple data types. It belongs to <class ‘list’>.

Fruits = [‘apple’, ‘mango’]

---: homogenous list collection of items that belongs to only 1 data type. In this case its of string data type

Fruit\_prices = [10, 12.5, 20, 45.5]

---: heterogenous list collection of items that belongs to multiple data types . In this case its collection of integer & float data types.

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

Expression is made up of operands and operators (like mathematical +,-,/,\* or logical etc). Expression will be evaluated in the proper order based on operator precedence and result into a one single final value.

All expressions will result into a one single final value.

For example:

* total\_subjects=3 ---: this is not an expression its an assignment statement
* average\_marks\_scored = ((90 + 95 + 100)/total\_subjects) ---: this is an expression
  + + & / are operators
  + 90 , 95, 99 & total\_subjects are operands
  + 75.0 is the resultant value

Python has various types of expressions like:

1. Arithmetic Expression
   1. supports mathematical operators like +, -, \*, /, %, //, \*\* etc
   2. Example: total\_subjects=3

average\_marks\_scored = ((90 + 95 + 99)/total\_subjects

1. Relational Expression
   1. Supports relational operators like >, >=, < , <= , ==, != etc
   2. The results value of these kind of expressions will be either True or False
   3. Example : 2 > 3 ---: results in False & 2!=3 --: results in True
2. Constant Expression
   1. It contains only constant values as operands
   2. Example: a = 1 + 2
3. Logical Expression
   1. Supports and, or, not operators
   2. The results value of these kind of expressions will be either True or False
   3. ((1<2) and (2<3)) ---: (True and True) ---: True
4. Integral Expression & Floating Expression
   1. type casting (only numeric string) to integer and integer to float or vice-versa
   2. total\_marks = int(‘50’) + float(10) + int(15.4) --: results in float value of 75.0
5. Bitwise Expression
   1. Supports 2 operators like >> (right shift) & << (left shift)
   2. Operation is performed at bit level for example: k = 8, k = k <<1 #left shift by 1 that means it shifts the bits left wards and add one 0 in the right most became 0000 1000 will become 0001 0000 (i.e, 16) and vice-versa for right shift 0000 1000 will become 0000 0100 (i.e, 4)
6. Combinational Expression
   1. It’s a combination of one or more expressions
   2. Example: (1>2) and ((5+2)==6)

(False) and ((7)==6)

(False) and (False)

False

(5+2) Arithmetic Expression , ((5+2)==6) Relational Expression

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

Assignment statement is used for variable creation Example: name = ‘harry’

Expression has operators and operands where it will be evaluated by the python interpreter and results into a single value. Example: score = 90 + 95 is an constant expression with arithmetic addition operator.

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

bacon = 22

bacon + 1

Variable bacon will still contain the original value i.e, 22, because bacon + 1 expression is not assigned to any variable.

**Note:** If it is something like bacon = bacon + 1 then bacon will be 23

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

'spam' + 'spamspam'. ---: this will result in concatenation of two strings i.e, 'spamspamspam'

'spam' \* 3. ---: this will concatenate the ‘spam’ string 3 times i.e, ‘spamspamspam’

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

Variable names cannot start with number (in this case 100 is a number that’s why this is invalid)

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

int() ---: to get the integer version value ,

float() ---: to get the floating-point number version value &

str() ---: to get the string version of value

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

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

We cannot concatenate (add) string and int data types that will lead to a “TypeError”. In order to make it work we need to convert the 99 i.e, integer to string data type by using builtin function str() to convert integer into string as shown below.

**'I have eaten ' + str(99) + ' burritos.'**