1. In the below elements which of them are values or an expression? E.g.: - 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?

**Variables** can be defined as a placeholder to hold various types of data like string, integer, float etc. in the main memory. When defining a variable we have to follow certain naming conventions. A variable name cannot start with a number and cannot contain wild characters (&, %, \*, $, #, @). It can start with an underscore. Variables in python need not be declared with a data type.

Examples of valid variable names include username, email etc. while %abcd, #abcd, @abcd, 123abcd are invalid variable names.

**String** is one of the data types in python. A sequence of characters forms a string.   
When a string is being defined, it has to be enclosed in a double quote or single quote.   
Examples of string are “1234”, “abcd”, ‘123abcd’.

3. Describe three different data types.

The three different data types are:

1. **Int**: Int data type in python is positive or negative whole number without any decimals. All integer variables belong to the int() class. An integer cannot be defined with leading zeros, for example a = 098679 is not allowed. Integers in python 3 are of **unlimited size**.

Examples of valid integers in python: a = 10, b = -100, c = 1234568117.

1. **Float**: Float data type in python is composed of positive or negative real numbers including decimals. All float variables belong to float() class.

Examples of float data type are a = 10.0, b = -100.34, c = 123456.789.

1. **String**: A sequence of Unicode characters forms a string. A string has to be enclosed in a double quote, single quote or a triple quote while defining.

For example, syntax for creating a string is:

a = “Hello World”

b = “123Python”

Python doesn’t have the character data type. A single character is treated as a string of length 1.

The in-built **type()** function can be used to know the data type of the variable that has been defined.

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

**Expression** is a combination of operators and operands that the python interpreter evaluates to an output value and this value can be assigned to any other variable.   
An expression can be a part of the statement.

1. **Arithmetic expressions**: These perform arithmetic operations like addition, subtraction, multiplication, division, modulo, exponentiation, floor division.

Examples of arithmetic expressions:

a = 10 + 4, a = 10//4, a = 10\*\*4, a = 10\*4 etc.

1. **Comparison expressions**: These expressions contain relational operators like (<, >, ==, >=, <=, != ). They will return true or false after evaluating the expression on both the sides of the operator.

Examples of comparison expression: (a + b) >= (c - d)

1. **Combinational expressions:** These are composed of different type of expressions in a single expression. In combinational expressions operator precedence takes effect.

Example: a+=5, a + b > c\*\*d.

1. **Logical expressions:** AND, OR, NOT are logical operators. They return true or false based on the conditions.

Example: p AND q 🡪 returns true if both p and q are true

p OR q 🡪 returns true if either p or q are true

not p 🡪 returns true if p is false.

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

**Expression** is a combination of operators and operands that the python interpreter evaluates to an output value and we can assign this value to a new variable. An expression can be a part of the statement.

**Statement** on the other hand represents some action like printing a value, assigning a value, if statement for checking conditions.

spam = 10 is a statement in this scenario.

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

bacon = 22

bacon + 1

The variable bacon contains the value **22**.

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

'spam' + 'spamspam' = ‘spamspamspam’

'spam' \* 3 = ‘spamspamspam’

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

100 is an invalid variable name because variable names can’t begin with a number. Below are the points that should be checked while naming variables:

1. Variable names can’t begin with number.
2. Variable names can’t begin with special characters, like @, #, $, %, &

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

**int()** function is used to get the integer of a value. For example, int(9.0) = 9

**float()** function is used to get the floating-point number. For example, float(22) = 22.0

**str()** function is used to get the string version of a given input.

For example, str(1234) = ‘1234’

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

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

The above expression gives an error because we can’t concatenate a string and an integer value. We can fix this problem by converting integer value 99 to string version ‘99’.

'I have eaten ' + ‘99’ + ' burritos.'

Output: ‘I have eaten 99 burritos.’