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?

Variables are symbols that we can use to store data in a program, so they are empty boxes that you fill with some data or value.

Strings are data, so we can use them to fill up a variable.

3. Describe three different data types.

The standard or built-in data type of Python are- numeric (integer, float, complex number), sequence type (strings, list, tuple), Boolean, set, dictionary.

**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.

Integers – 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 Numbers – Complex number is represented by complex class. It is specified as (real part) + (imaginary part)j. For example – 2+3j

**Sequence Type:** Sequence is the ordered collection of similar or different data types. Sequences allows to store multiple values in an organized and efficient fashion. The sequence types are string, list, tuple.

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.

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.

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.

**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.

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

A combination of operands and operators is called an expression. The expression in Python produces some value or result after being interpreted by the Python interpreter. An expression in Python is a combination of operators and operands.

An example of expression can be :

x=x+10. In this expression, the first 10 is added to the variable x. After the addition is performed, the result is assigned to the variable x.

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

spam= 10 is a statement where the variable ‘spam’ is being assigned to a value of 10.

x = 25 # a statement

x = x + 10 # an expression

A statement is used for creating variables or for displaying values. Unlike expressions, a statement is not evaluated for some results.

|  |  |
| --- | --- |
| **Statement in Python** | **Expression in Python** |
| A statement in Python is used for creating variables or for displaying values. | The expression in Python produces some value or result after being interpreted by the Python interpreter. |
| A statement in Python is not evaluated for some results. | An expression in Python is evaluated for some results. |
| The execution of a statement changes the state of the variable. | The expression evaluation does not result in any state change. |
| A statement can be an expression. | An expression is not a statement. |
| Example :  x=3.  Output :  3 | Example:  x=3+6.  Output :  9 |

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

bacon = 22

bacon + 1

Output: 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?

Rules for Python variables:

* A variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* Variable names are case-sensitive (age, Age and AGE are three different variables)

Based on the above rules since ‘eggs’ start with a letter it is a valid variable name.

Since ‘100’ starts with number it is not a valid variable name.

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

Function to get integer version of a value is int().

Function to get floating-point number version of a value is float().

Function to get string version of a value is str().

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

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

The given expression cause an error because we can only concatenate strings to strings, and not strings to integers.

We can fix it by converting 99 from integer type to string type.

'I have eaten ' + ‘99’ + ' burritos.'

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