**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.**

\*

'hello' 🡪 string value

-87.8 🡪 Integer value

- 🡪Expression (Operator)

/ 🡪 Expression (Operator)

* 🡪 Expression (Operator)

6 🡪 Expression (Operator)

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

|  |  |
| --- | --- |
| **String** | **Variable** |
| String is a **datatype**, thus specifies the type of a variable.  It’s a datatype, which specifies sequence of character / character and numeric / character and numeric and special characters. | It can be of some specific datatype (int / float / string / list / dictionary / tupple / set). |
| In order to specify a value to a string-type variable, one has to write the data inside Quotation mark.  i.e., Var = ‘Eagle’ | Value to a variable can be assigned, by using Equal to operator.  E.g., Var1 = 1  Var2 = ‘Eagle’, Var3 = 3.0 |
| A string-type variable holds a default hidden character ‘NULL-character’  String-type variable ends with ‘NULL-character’ | Variable can be of any type |
| String datatype is a sequence type. |  |

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

Python supports following built-in datatypes:

1. Numeric (Variable of this type can hold the numeric values)
   1. Integer (Variable of this type can hold the integer numeric values. E.g., x = 5, y = -5)
   2. Complex number (Variable of this type can hold the complex numeric values. E.g., x = 5+7j, y = -5+8j)
   3. Float (Variable of this type can hold the decimal numeric values. x = 4.0000, y= 0.015)
2. Boolean (Variable of this type can hold the boolean values. E.g., x = True, y = False)
3. Dictionary (Variable of this type holds key-value pair. E.g., x = {a:5, b:3, c:6, 1: [1, 2, 3, 4]},)
4. Set (Variable of this type holds un-ordered collection of values).
5. Sequence Type
   1. Strings
   2. List
   3. Tuple

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

Ans. An expression is a combination of operators and operands. In any programming language, an expression is evaluated as per the precedence of its operators. So that if there is more than one operator in an expression, their precedence decides which operation will be performed first. We have many different types of expressions in Python. They are:

**Constant Expression**: Expression contains “Constant value”.

**Arithmetic Expression**: Combination of numeric values, operators, and sometimes parenthesis. The result of this type of expression is also a numeric value. Here are some arithmetic operators:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Operators** | | |  | | --- | | **Syntax** | | **Functioning** |
| + | x + y | Addition |
| |  |  |  | | --- | --- | --- | | – |  |  | | x – y | Subtraction |
| |  |  |  | | --- | --- | --- | | \* |  |  | | x \* y | Multiplication |
| |  |  |  | | --- | --- | --- | | / |  |  | | x / y | Division |
| |  | | --- | | // | | x // y | Quotient |
| |  | | --- | | % | | x % y | Remainder |
| |  |  | | --- | --- | | \*\* |  | | x \*\* y | Exponentiation |

**Integral Expression**: These are the kind of expressions that produce only integer results after all computations and type conversions

a = 13

b = 12.5 # float

c = a+b # c becomes float

int ( c) # Type conversion

**Floating Expression:**

These are the kind of expressions which produce floating point numbers as result after all computations and type conversions.

**Relational Expressions:**

**Relational operators (>, <, >=, <=) are used in this expression.**

**Logical Expressions:**

**Expressions results either True or False.**

|  |  |
| --- | --- |
| **Operator** | **Syntax** |
| **and** | **P and Q** |
| **or** | **P or Q** |
| **not** | **not P** |

**Bitwise Expressions:**

**Expressions, in which computations are performed at bit level.**

**Combinational Expressions:**

**Combination of different expressions.**

**E.g., c = a + (b >>1)**

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

Ans. Expression is a combination of variables (operands), operators. Expression yields a result value.

A statement represents an action or command. E.g., Print statement, Assignment statement etc.

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

**bacon = 22**

**bacon + 1**

Ans. Variable bacon contains value = 22.

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

**'spam' + 'spamspam'**

**'spam' \* 3**

Ans. spamspamspam

spamspamspam

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

Ans. A variable name must start with a letter or the underscore character. It can’t start with a number. However variable name can contain alpha-numeric characters and underscores. Thus variable name ‘100’ is invalid and the variable name ‘eggs’ is valid.

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

Ans. To get integer version of a value, function used is **int(value)**

To get floating-point number version of a value, function used is **float(value)**.

To get string version of a value, function used is **string(value).**

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

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

Ans. Concatenation of values of different type is not possible. So 99 needs to be a string type for the expression to be come error free. i.e., 'I have eaten ' + **‘**99**’** + ' burritos.'