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.

Ans-)

\* ……………………..Mathematical operator

'hello' ……………… string

-87.8 ……………….. int

- ……………………….. Mathematical operator

/………………………… Mathematical operator

+ ………………………. Mathematical operator

6………………………… int

2. What is the difference between string and variable?

Ans-) A **Variable** is a store of information

e.g. x = 10, name = 45

**String** is a data type, which reflect type of information you would store **in a Variable**.

e.g. int, String, float, double.

3. Describe three different data types.

Ans-)

Python has the following data types built-in by default, in these categories:

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | Int, float, Complex |
| Sequence Types: | List, tuple, Range |
| Mapping Type: | dict |
| Set Types: | Set, frozenset |
| Boolean Type: | bool |
| Binary Types: | Bytes, bytesarray, memoryview |

## Sequence Type

A sequence is an ordered collection of similar or different data types. Python has the following built-in sequence data types:

* [String](https://www.tutorialsteacher.com/python/python-string): A string value is a collection of one or more characters put in single, double or triple quotes.
* [List](https://www.tutorialsteacher.com/python/python-list): A list object is an ordered collection of one or more data items, not necessarily of the same type, put in square brackets.
* [Tuple](https://www.tutorialsteacher.com/python/python-tuple): A Tuple object is an ordered collection of one or more data items, not necessarily of the same type, put in parentheses.

## Mapping Type

[Dictionary](https://www.tutorialsteacher.com/python/python-dictionary): A dictionary Dict() object is an unordered collection of data in a key:value pair form. A collection of such pairs is enclosed in curly brackets. For example: {1:"Steve", 2:"Bill", 3:"Ram", 4: "Farha"}

## Set Types

* [set](https://www.tutorialsteacher.com/python/python-set): Set is mutable, unordered collection of distinct hashable objects. The set is a Python implementation of the set in Mathematics. A set object has suitable methods to perform mathematical set operations like union, intersection, difference, etc.
* **frozenset**: Frozenset is immutable version of set whose elements are added from other iterables.

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

Ans-) Expressions are representations of value. They are different from statement in the fact that statements do something while expressions are representation of value.

Python expressions only contain identifiers, literals, and operators. So, what are these?

**Identifiers**: Any name that is used to define a class, function, variable module, or object is an identifier.

**Literals**: These are language-independent terms in Python and should exist independently in any programming language. In Python, there are the string literals, byte literals, integer literals, floating point literals, and imaginary literals.

**Operators**: In Python you can implement the following operations using the corresponding tokens.

| **Operator** | **Token** |
| --- | --- |
| add | + |
| subtract | - |
| multiply | \* |
| power | \*\* |
| Integer Division | / |
| remainder | % |
| decorator | @ |
| Binary left shift | << |
| Binary right shift | >> |
| and | & |
| or | \ |
| Binary Xor | ^ |
| Binary ones complement | ~ |
| Less than | < |
| Greater than | > |
| Less than or equal to | <= |
| Greater than or equal to | >= |
| Check equality | == |
| Check not equal | != |

Following are a few types of python expressions:

### List comprehension

The syntax for list comprehension is shown below:

[ compute(var) for var in iterable ]

For example, the following code will get all the number within 10 and put them in a list.

>>> [x for x in range(10)]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

### Dictionary comprehension

This is the same as list comprehension but will use curly braces:

{ k, v for k in iterable }

For example, the following code will get all the numbers within 5 as the keys and will keep the corresponding squares of those numbers as the values.

>>> {x:x\*\*2 for x in range(5)}

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16}

### Generator expression

The syntax for generator expression is shown below:

( compute(var) for var in iterable )

For example, the following code will initialize a generator object that returns the values within 10 when the object is called.

>>> (x for x in range(10))

<generator object <genexpr> at 0x7fec47aee870>

>>> list(x for x in range(10))

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

### Conditional Expressions

You can use the following construct for one-liner conditions:

true\_value if Condition else false\_value

Example:

>>> x = "1" if True else "2"

>>> x

'1'

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

Ans-) An expression evaluates to a single value. A statement does not.

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

Ans-) bacon = 22:- It means in memory file name is created bacon and in that bacon my variable 10 is stored.

bacon + 1:- It will give you result 22+1=23.

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

Ans-)

'spam' + 'spamspam' :- Result will be 'spamspamspam'

'spam' \* 3 :- Result will be 'spamspamspam'

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

Ans-) Variable names cannot begin with a number.

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

Ans-) The int(), float(), and str() functions will evaluate to the integer, floating-point number, and string versions of the value passed to them

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

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

Ans-) 99 is an integer. To solve error you should have to convert int into str.

Eg. ‘I have eaten’ + str(99) + ‘burritos’