1. What are the differences between operators and values in the following?

\*

'hello'

-87.8

-

/

+

6

**Ans:** The difference between operators and values:

\* : It is an Arithmetic operator which multiply two operands

'hello' : It is value of type string. The [Quotation marks](https://en.wikipedia.org/wiki/Quotation_mark) (‘ ’) is used to indicate String value (hello).

-87.8 : ‘-’ sign indicates that number is negative. The value is negative number. ‘-’ is unary minus.

- : ‘-’ sign is arithmetic operator, it is used for subtraction or it is unary minus.

/ : ‘/’ sign is arithmetic operator used for division, always give float number as output.

+ : ‘+’ sign is Arithmetic operator which add two operands.

6 : It is numeric value.

2. What is the difference between string and variable?

spam

'spam'

Ans:

spam : It is not a keyword in python, hence it is variable name. Variable can store any value.

‘spam’: It is enclosed in quotation marks (‘ ’), hence it string value.

3. Describe three different data forms.

Ans:

Data types of python:

1. Numeric types : int, float, complex

int: Integer values (except fractional values). e.g. -256, -431, 0, 15456, 234 etc.

float: Any real number with a floating point representation. e.g. 1.23, 4.5334, etc.

complex: A real and imaginary numbers represented as X+2Y. e.g., 2+ 4i

2. String type: str

str: It is sequence of characters enclosed with single, double, or triple quotes.

3. Mapping type: dict

dict: It is collection of data in a key:value pair form.

e.g., {1: “Hello”, 2:”Hi”, 3:”Hey”}

4. What makes up an expression? What are the functions of all expressions?

An expression is made up of values, variables, operators and method invocations, which are constructed according to the syntax of the langauge, that evaluates to a single value.

e.g., x \* 1 + y

**Types of expression:**

**1. List comprehension:**

The syntax is:

[ compute(var) for var in iterable ]

e.g., The following code will construct list of numbers within 10.

[ x for x in range(10) ]

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

**2. Dictionary comprehension:**

The syntax is:

{ k, v for k in iterable }

e.g., The following code will construct list of square of numbers within 5.

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

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

**3. Generator expression:**

The syntax is:

( compute(var) for var in iterable )

e.g., The following code will construct a generator object that returns list of numbers within 10 when it is called.

( x for x in range(10) )

output: <generator object <genexpr> at 0x0000023B7951FC80>

list (x for x in range(10))

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

**4. Conditional expressions:**

The syntax is:

true\_value if condition else false\_value

e.g.,

x=1 if True else 2

Conditional statement evaluates to True hence, x will be assigned with vale 1.

5. In this chapter, assignment statements such as spam = 10 were added. What's the difference between a declaration and an expression?

Ans:

An expression evaluates to a value. A statement does something. The statement is an instruction that python interpreter can execute. Expressions need to be evaluated, after evaluation interpreter displays result.

Python does not have a command for declaring a variable. A variable is declared when it is initialized with value.

e.g.,

x + 2 # an expression

x = 1 # a declaration statement

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

bacon = 22

bacon + 1

Ans.

>>> bacon = 22 # 22 is stored in bacon

>>> bacon +1 # 1 is added to 22 and will be displayed

output: 23

>>> bacon # 1 is added to bacon but not assigned back to bacon. Hence the value in bacon will not be updated.

output: 22

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

'spam' + 'spamspam'

'spam' \* 3

Ans:

Input: 'spam' + 'spamspam'

Output: 'spamspamspam'

Input: 'spam' \* 3

Output: 'spamspamspam'

8. Why is it that eggs is a true variable name but 100 is not?

Ans:

Variable is the name that can represent different values during the execution of the programs. e.g., eggs is a variable. (if not written in quotes.)

On the other hand, a constant represents a same value throughout a program. Literal is not a name, it is value itself. e.g., 100 is literal.

Hence 100 can not be used as variable.

9. Which of the following three functions may be used to convert a value to an integer, a floating-point number, or a string?

Ans:

>>> x=10

>>> x

10

>>> float(x)

10.0

float(variable\_name) used to convert value to floating-point number

>>> int(x)

10

int(vaiable\_name) used to convert value to integer number

>>> str(x)

'10'

str(variable\_name) used to convert value to string

10. What is the error caused by this expression? What would you do about it?

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

Ans:

Error: Concatenate can be performed on ‘str’ only not on ‘int’

Solution: There can be two ways:

1. Remove 99

2. Put 99 in double quotes or use str(99) to convert number to string