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?

=> **Variable** is something that holds a value that may change in future. In simple terms its like fridge in your house in which you can put different type of stuff

Example:

sun = 2

print (sun)

2

**String** is a list of characters from a-z , it doesn’t have any limit or rules.

Example:

print ('My name is Aayushi')

'My name is Aayushi'

3. Describe three different data types.

1. **Numbers**

Python numbers variables are created by the below method:

var = 382

Under certain circumstances that a specific number type is needed (ie. complex, hexidecimal), the format can be forced into a format by using additional syntax in the table below:

Type Format Description

int a = 10 Signed Integer

long a = 345L (L) Long integers

float a = 45.67 (.) Floating point real values

complex a = 3.14J (J) Contains integer in the range 0 to 255.

1. **List**

A list can contain a series of values.

List variables are declared by using brackets [ ] .

Eg: x = [ ] # This is a blank list variable

y = [12, 22, 2, 23] # this list creates an initial list of 4 numbers.

1. **Tuple**

Tuples are a group of values like a list and are manipulated in similar ways. But, tuples are fixed in size once they are assigned.Tuples are defined by parenthesis ().

Eg:

fruits = ('Mango', 'Graper', 'Strawberry', 'kiwi')

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

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

Example :

length = 5

breadth = 2

area = length \* breadth

print('Area is', area)

print('Perimeter is', 2 \* (length + breadth))

Output

:Area is 10

Perimeter is 14

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

=> If you can print it, or assign it to a variable, it’s an expression. If you can’t, it’s a statement.

Here are some examples of expressions:

2 + 2

3 \* 7

1 + 2 + 3 \* (8 \*\* 9) - sqrt(4.0)

min(2, 22)

max(3, 94)

round(81.5)

"foo"

"bar"

"foo" + "bar"

None

True

False

2

3

4.0

All of the above can be printed or assigned to a variable.

Here are some examples of statements:

if CONDITION:

elif CONDITION:

else:

for VARIABLE in SEQUENCE:

while CONDITION:

try:

except EXCEPTION as e:

class MYCLASS:

def MYFUNCTION():

return SOMETHING

raise SOMETHING

with SOMETHING:

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

bacon = 22

bacon + 1

* Given code is not complete: Correct code should be like one below

**INPUT:**

bacon = 22

print(bacon + 1)

**OUTPUT**: 23

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

'spam' + 'spamspam'

'spam' \* 3

* No output

The bold part is incorrect we cannot append two words like this.

'spam' + **'spamspam'**

'spam' \* 3

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

=> The value of variable should be stored somewhere, here eggs is a string in which we can store anything but in 100 we cant store.

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

=> Doesn’t understood the question

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

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

* This comes under string formatting , here we use format() method.

**Given code** misses the variable in which we can define 99 and the print statement , format method and the text should be defined before printing.

**Fixed code:**

INPUT:

quantity = 99

txt = 'I have eaten {} burritos.'

print(txt.format(quantity))

OUTPUT:

I have eaten 99 burritos.