

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'

-87.8

/ +

In []:

```
*          - Operator(Multiplication)
'hello'    - String Value
-87.8      - Double Value
-          - Operator(Subtraction)
/          - Operator(Division)
+          - Operator(Addition)
```

In [7]:

Example:

```
a = int(input("Enter 1st number: "))
b = int(input("Enter 2nd number: "))

sum = a + b
mul = a * b
div = a / b
sub = a - b
print()
print(f"The 'sum' of {a} & {b} is: {sum}")
print(f"The 'mul' of {a} & {b} is: {mul}")
print(f"The 'div' of {a} & {b} is: {div}")
print(f"The 'Sub' of {a} & {b} is: {sub}")
```

Enter 1st number: 2

Enter 2nd number: 4

The 'sum' of 2 & 4 is: 6

The 'mul' of 2 & 4 is: 8

The 'div' of 2 & 4 is: 0.5

The 'Sub' of 2 & 4 is: -2

2. What is the difference between string and variable?

3. Describe three different data types.

Text Type - str

Numeric Types - int, float, complex

Sequence Types - list, tuple, range

In [12]:

```
#Numeric type Example:
a = 5
print("Type of a: ", type(a))

b = 5.0
print("\nType of b: ", type(b))

c = 2 + 4j
print("\nType of c: ", type(c))
```

Type of a: <class 'int'>

Type of b: <class 'float'>

Type of c: <class 'complex'>

In []:

```
#Text Type:

print("Hi there")

print = "Hi there"
type(print)

print("Hi there")
```

In [15]:

```
#Sequence Types -list, tuple, Dict
List = [1,2,3,4,5]
print("\nList containing multiple values: ")
print(List[0])
print(List[2])

Tuple1 = (0, 1, 2, 3)
Tuple2 = ('python', 'Coding', 'DataScience')
Tuple3 = (Tuple1, Tuple2)
print("\nTuple with nested tuples: ")
print(Tuple3)

Dict = dict({1: 'Python', 2: 'For', 3: 'DataScience'})
print("\nDictionary with the use of dict(): ")
print(Dict)
```

List containing multiple values:

1
3

Tuple with nested tuples:

((0, 1, 2, 3), ('python', 'Coding', 'DataScience'))

Dictionary with the use of dict():

{1: 'Python', 2: 'For', 3: 'DataScience'}

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

An expression is a combination of operators and operands that is interpreted to produce some other value. In any programming language, an expression is evaluated as per the precedence of its operators

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

Expression is a combination of variables, operations, and values yields a result value.

Statements represent an action or command

Expression example:

`a+b`

statement example:

`print 'hello', x = 1`

In [22]:

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

```
bacon = 22
bacon + 1
print(bacon)
# 22 because we are just adding not assigning to variable
```

22

7. What should the values of the following two terms be? `'spam' + 'spamspam'` `'spam' * 3`

In [23]:

```
val1 = 'spam' + 'spamspam'
val2 = 'spam' * 3
print(val1)
print(val2)
```

```
spamspamspam
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)

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

Integer — `int()`

Floating-point — `float()`

String — `str()`

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

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

"99" is an integer and not a string, so the operations are incompatible until "99" is converted into a string.

Fix: use `str()` to convert "99" to a string.

In []: