Q1. Create one variable containing following type of data:

(i) string

(ii) list

(iii) float

(iv) tuple

In [1]:
```
a='Mirza yasar beg'
```

In [6]:
```
a
```

Out[6]:
```
'Mirza yasar beg'
```

In [7]:
```
type(a)
```

Out[7]:
```
str
```

In [3]:
```
l=[1,2,3,4,'yasar']
```

In [4]:
```
l
```

Out[4]:
```
[1, 2, 3, 4, 'yasar']
```

In [5]:
```
type(l)
```

Out[5]:
```
list
```

In [8]:
```
c=3.56
```

In [9]:
```
type(c)
```

Out[9]:
```
float
```

In [10]:
```
t=(1,56,'yasar',58,890)
```

In [11]:
```
t
```

Out[11]:
```
(1, 56, 'yasar', 58, 890)
```

In [12]:
```
type(t)
```

Out[12]:
```
tuple
```

In [ ]:
```
Q2. Given are some following variables containing data:
(i) var1 = ‘ ‘
(ii) var2 = ‘[ DS , ML , Python]’
(iii) var3 = [ ‘DS’ , ’ML’ , ‘Python’ ]
```

```
(iv) var4 = 1.
What will be the data type of the above given variable.
```

1.var1='' There is string data type stored in var1

2.var2='[DS,ML,python]' Ans. Since whatever writen in the codes that is string so in var2 stored string value

Ans3. since everything in square braces terat as list so therfore there is list stored in var2 means the data type of var2 is list

Ans4. var4 contains numeric value so the data type of var4 is int

Q3. Explain the use of the following operators using an example:

(i) /

(ii) %

(iii) //

(iv) **

In [1]:
```
a=5/2
```

In [2]:
```
a
```

Out[2]:
```
2.5
```

ans1. / is used to divide the two numbers

In [4]:
```
a=5%2
a
```

Out[4]:
```
1
```

Ans2. % is known as the modulo operator which is used to get the remainder

In [ ]:
```
Ans3. // is used to get the whole part of the decimal value
```

In [5]:
```
5//3
```

Out[5]:
```
1
```

In [ ]:
```
Ans4. ** is used to calculate the power of any no.
```

In [6]:
```
5**2
```

Out[6]:
```
25
```

Q4. Create a list of length 10 of your choice containing multiple types of data. Using for loop print the element and its data type.

In [8]:
```
l = [1, "two", 3.0, True, [4, 5], {"name": "John", "age": 30}, None, (6, 7), False,

for i in l:
    print(i, type(i))
```

```
1 <class 'int'>
two <class 'str'>
3.0 <class 'float'>
True <class 'bool'>
[4, 5] <class 'list'>
{'name': 'John', 'age': 30} <class 'dict'>
None <class 'NoneType'>
(6, 7) <class 'tuple'>
False <class 'bool'>
ten <class 'str'>
```

Q5. Using a while loop, verify if the number A is purely divisible by number B and if so then how many times it can be divisible

In [9]:
```
A = 24
B = 4

count = 0

while A % B == 0:
    A = A / B
    count += 1

print("Number A is divisible by number B", count, "times.")
```

```
Number A is divisible by number B 1 times.
```

Q6. Create a list containing 25 int type data. Using for loop and if-else condition print if the element is divisible by 3 or not.

In [10]:
```
my_list = [12, 7, 18, 5, 33, 42, 9, 15, 28, 2, 21, 36, 10, 14, 30, 8, 19, 27, 4, 17

for element in my_list:
    if element % 3 == 0:
        print(element, "is divisible by 3")
    else:
        print(element, "is not divisible by 3")
```

```
12 is divisible by 3
7 is not divisible by 3
18 is divisible by 3
5 is not divisible by 3
33 is divisible by 3
42 is divisible by 3
9 is divisible by 3
15 is divisible by 3
28 is not divisible by 3
2 is not divisible by 3
21 is divisible by 3
36 is divisible by 3
10 is not divisible by 3
14 is not divisible by 3
30 is divisible by 3
8 is not divisible by 3
19 is not divisible by 3
27 is divisible by 3
4 is not divisible by 3
17 is not divisible by 3
24 is divisible by 3
6 is divisible by 3
11 is not divisible by 3
22 is not divisible by 3
13 is not divisible by 3
```

Q7. What do you understand about mutable and immutable data types? Give examples for both showing this property.

Mutable data type: Mutable data types allow modifications to the object's state after it is created. Examples of mutable data types in Python include:

Lists:

In [11]:
```
list1 = [1, 2, 3]
list1.append(4)
print(list1)  # Output: [1, 2, 3, 4]

list2 = list1
list2[0] = 5
print(list1)
```

```
[1, 2, 3, 4]
[5, 2, 3, 4]
```

In the above example, the append() method modifies the original list (list1) by adding an element. Similarly, modifying list2 also affects list1 since they refer to the same list object.

immutable data type: immutable data types, such as strings and tuples in Python, cannot be changed after creation. You need to create a new object if you want to make any modifications for example string and tuple are immutable data typr

In [27]:
```
t=("yasar")
```

In [28]:
```
t
```

Out[28]:
```
'yasar'
```

In [29]:
```
t[0]="pasar"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 t[0]="pasar"

TypeError: 'str' object does not support item assignment
```

In [ ]:
```

```