Q1. Create one variable containing following type of data:

(i) string (ii) list (iii) float (iv) tuple

In [6]:
```
# (i) string
string_variable = "Hello, World!"

# (ii) list
list_variable = [1, 2, 3, 4, 5]

# (iii) float
float_variable = 3.14

# (iv) tuple
tuple_variable = (10, 20, 30)
```

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.

In [8]:
```
# (i) var1 = ' '
# Data type: string

# (ii) var2 = '[ DS , ML , Python]'
# Data type: string

# (iii) var3 = [ 'DS' , 'ML' , 'Python' ]
# Data type: list

# (iv) var4 = 1.
# Data type: float
```

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

(i) / (ii) % (iii) // (iv) **

(i) /: Division operator

In [9]:
```
result = 10 / 3
print(result)  # Output: 3.3333333333333335
```

```
3.3333333333333335
```

(ii) %: Modulus operator

In [10]:
```
remainder = 10 % 3
print(remainder)  # Output: 1
```

```
1
```

(iii) //: Floor division operator

In [11]:
```
quotient = 10 // 3
print(quotient)  # Output: 3
```

```
3
```

(iv) **: Exponentiation operator

In [12]:
```
power = 2 ** 3
print(power)  # Output: 8
```

```
8
```

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 [13]:
```
mixed_data_list = [1, "Hello", 3.14, True, [1, 2, 3], {'key': 'value'}, (4, 5, 6

for item in mixed_data_list:
    print(f"Element: {item}, Data Type: {type(item)}")
```

```
Element: 1, Data Type: <class 'int'>
Element: Hello, Data Type: <class 'str'>
Element: 3.14, Data Type: <class 'float'>
Element: True, Data Type: <class 'bool'>
Element: [1, 2, 3], Data Type: <class 'list'>
Element: {'key': 'value'}, Data Type: <class 'dict'>
Element: (4, 5, 6), Data Type: <class 'tuple'>
Element: None, Data Type: <class 'NoneType'>
Element: a, Data Type: <class 'str'>
Element: 10, Data Type: <class 'int'>
```

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 [15]:
```
A = 15
B = 3

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

print(f"A is divisible by B {count} times.")
```

```
A is divisible by 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 [16]:
```
int_list = list(range(1, 26))

for num in int_list:
    if num % 3 == 0:
        print(f"{num} is divisible by 3.")
    else:
        print(f"{num} is not divisible by 3.")
```

```
1 is not divisible by 3.
2 is not divisible by 3.
3 is divisible by 3.
4 is not divisible by 3.
5 is not divisible by 3.
6 is divisible by 3.
7 is not divisible by 3.
8 is not divisible by 3.
9 is divisible by 3.
10 is not divisible by 3.
11 is not divisible by 3.
12 is divisible by 3.
13 is not divisible by 3.
14 is not divisible by 3.
15 is divisible by 3.
16 is not divisible by 3.
17 is not divisible by 3.
18 is divisible by 3.
19 is not divisible by 3.
20 is not divisible by 3.
21 is divisible by 3.
22 is not divisible by 3.
23 is not divisible by 3.
24 is divisible by 3.
25 is not divisible by 3.
```

In [ ]:
```
Q7. What do you understand about mutable and immutable data types? Give examples
this property.
```

Mutable data types can be modified after creation, while immutable data types cannot be changed.

Mutable Example (List):

In [18]:
```
mutable_list = [1, 2, 3]
mutable_list[0] = 10
print(mutable_list)  # Output: [10, 2, 3]
```

```
[10, 2, 3]
```

Immutable Example (Tuple):

In [20]:
```
immutable_tuple = (1, 2, 3)
# The following line would raise an error:
# immutable_tuple[0] = 10
```