Call by Value:- It means that when you pass an argument to a function, a copy of the value is passed.
The function operates on this copy, & any changes to the parameter inside the function do not affect the original variable.

Call by Reference:- It means that when you pass an argument to a function, the function operates on the original value (memory address), & any changes to the parameters inside the function will affect the original variable.

Data types in Python 3

1. ~~int~~ Numeric Types

- int (Integer) — y, x = 10, -5
- float (Floating Point) — y = -10.5
- complex (Complex Number) — x = 3+4j

2. Sequence Types

str (string) text = "Hello, world"

- list: A mutable, ordered collection of items

```
my_list = [1, 2, 3, 'Hi', 3.14]
```

- tuple: A immutable, ordered collection of items

```
my_tuple = (1, 2, 3, 'apple', -1.47)
```

3. Mapping Type

- dict (Dictionary): An unordered collection of key-value pairs.

Keys are unique & are used to retrieve corresponding value.

```
my_dict = {'name': 'Ash', 'age': 25}
```

apsara

Date: ___.___.___

4. Set Types

- set: An unordered collection of unique items.

```
my_set = {1, 2, 3, 4, 5}
```

- frozenset: Similar to set, but immutable.

```
frozen_set = frozenset([1, 2, 3, 4])
```

5. Boolean Type

bool: Represent either True or False

```
is_active = True
is_empty = False
```

6. None Type

- None: Represent the absence of a value or a null value.

It is often used for default function return values or when a variable has not been initialized.

```
x = None
```

apsara

Date: ___.___.____

7. Binary Types.

- **bytes**: Immutable sequence of bytes (often used for binary data).

```
my_bytes = b"hello"
```

- **bytearray**: Mutable sequence of bytes.

```
my_bytearray = bytearray([65, 66, 67])
```

- **memoryview**: A memory view object allows you to access the internal data of an object that supports the buffer protocol (like bytearray)

```
mv = memoryview(my_bytearray)
```

apsara

Date:

is and == operator

1. == (Equality Operator)

The == operator checks weather the values of 2 object (variable) are equal. & return True or False.

```
a, b = 10, 20
print(a==b)          # False.

a = [1,2,3]
b = [1,2,3]
print(a==b)          # True.
```

2. is (Identity Operator)

The is operator checks weather 2 objects refer to the same object in memory (i.e, it check for identity)

It compare the memory addresses of the objects, & if both objects point to the same location in memory (i.e, they are the exact same object), it returns True

Page No.:

apsara

Date:

```
a = 10
b = a
c = 10
```

→ a == b # True
→ a == c # True
→ b == c # True

→ a is b # True
→ b is c # False
→ c is a # False.

Feature	ASCII	UTF
Character Set	128 character (7-bit encoding)	Over 1.1 millon character
Encoding Size	1 byte per character (7-bits used)	1-4 bytes (UTF-8), 2 or 4 bytes (UTF-16), 4 bytes (UTF-32)
Language Support	Primarily English	Support all language Scripts, emojis
Backward Compatibility	Not backward-compatible with unicode	UTF-8 is backward-compatible with ASCII

Page No.:

Memory Usage	Low memory usage (1 byte per character)	More memory usage (upto 4 bytes per character)
Control characters	Includes basic control characters (0-31)	Include all unicode control characters
Example	A = 65	A = 65 (UTF-8), A = 0x0041 (UTF-16)