**Que 1 : Understanding data types: integers, floats, strings, lists, tuples, dictionaries, sets.**

**1. Integer (int)**

* Whole numbers (no decimal point)

age = 25

**2. Float (float)**

* Numbers with decimal points

price = 99.99

**3. String (str)**

* Text enclosed in quotes (single or double)

name = "Krishna"

**4. List (list)**

* Ordered, **changeable** collection of items
* Enclosed in square brackets []

fruits = ["apple", "banana", "mango"]

**5. Tuple (tuple)**

* Ordered, **unchangeable** collection of items
* Enclosed in parentheses ()

colors = ("red", "green", "blue")

**6. Dictionary (dict)**

* Collection of **key-value pairs**
* Enclosed in curly braces {}

student = {

"name": "Krishna",

"age": 20,

"grade": "A"

}

**7. Set (set)**

* **Unordered**, **no duplicates**
* Enclosed in curly braces {}

unique\_numbers = {1, 2, 3, 2, 1}

# Output: {1, 2, 3}

**Que 2 : Python variables and memory allocation.**

**Variable**

A variable is a **name** used to **store a value**.

name = "Krishna"

age = 20

**Memory Allocation**

* Python stores values as **objects in memory**.
* Variables are just **labels (references)** pointing to those objects.

x = 5

y = x # Both point to same value in memory

Use id(x) to see the memory address.

**Que 3 : Python operators: arithmetic, comparison, logical, bitwise.**

**Arithmetic Operator:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Meaning** | **Example** | **Result** |
| + | Addition | 5 + 3 | 8 |
| - | Subtraction | 5 - 2 | 3 |
| \* | Multiplication | 4 \* 2 | 8 |
| / | Division | 10 / 2 | 5.0 |
| // | Floor Division | 10 // 3 | 3 |
| % | Modulus | 10 % 3 | 1 |
| \*\* | Exponentiation | 2 \*\* 3 | 8 |

**Comparison Operators**

Used to compare values:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Meaning** | **Example** | **Result** |
|  | Equal to | 5 == 5 | 1 |
| != | Not equal to | 5 != 3 | 1 |
| > | Greater than | 5 > 2 | 1 |
| < | Less than | 3 < 5 | 1 |
| >= | Greater or equal | 5 >= 5 | 1 |
| <= | Less or equal | 2 <= 3 | 1 |

**Logical Operators**

Used to combine conditions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Meaning** | **Example** | **Result** |
| and | Both True | 5 > 2 and 4 > 1 | 1 |
| or | One True | 5 > 2 or 4 < 1 | 1 |
| not | Opposite | not(5 > 2) | 0 |

**Bitwise Operators**

Work at **binary level**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Meaning** | **Example** | **Result** |
| & | AND | 5 & 3 | 1 |
| ` | ` | OR | `5 |
| ^ | XOR | 5 ^ 3 | 6 |
| ~ | NOT | ~5 | -6 |
| << | Left shift | 5 << 1 | 10 |
| >> | Right shift | 5 >> 1 | 2 |