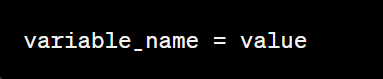
# VARIABLES :-

In Python, variables are used to store and manage data. You can think of them as containers that hold values or references to objects.



Syntax :

‘variable\_name’ is the name you choose for your variable.

[ Variable names should be meaningful and follow certain naming conventions.

Use lowercase letters and underscores for readability.]

‘value’ is the data or object that the variable holds.

# DATA TYPES :-

Python has several built-in data types to represent different kinds of data:

1. **Integers (int):** e.g., 5, -10, 100.
2. **Floating-Point Numbers (float):** Numbers with decimal points, e.g., 3.14, -0.5, 2.0.
3. **Strings (str):** Sequences of characters enclosed in single or double quotes, e.g., "Hello, World!".
4. **Boolean (bool):** Represents either True or False.
5. **Lists:** Ordered sequences of values, e.g., **[1, 2, 3]**.
6. **Dictionaries:** Unordered collections of key-value pairs, e.g., **{'name': 'Alice', 'age': 30}**.
7. **Tuples:** Ordered, immutable sequences, e.g., **(1, 2, 3)**.
8. **Sets:** Unordered collections of unique elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | List | Tuple | Set | Dictionary |
| Basics | Dynamically sized array that gets declared in other languages | Collections of various objects of Python separated by commas between them | Unordered collection of data types. | Collection (unordered) of various data types. |
| Representation | [] | () | {} | {} |
| Duplication | Allow | Allow | Deny | Deny |
| Example | [6, 7, 8, 9, 10] | (6, 7, 8, 9, 10) | {6, 7, 8, 9, 10} | {6, 7, 8, 9, 10} |
| Function | We can create a list using the list() function. | We can create a list using the tuple() function. | We can create a list using the set() function. | We can create a list using the dict() function. |
| Mutation | Mutable | Immutable | Mutable | Mutable0 |
| Order | Ordered | Ordered | Unordered | Ordered |
| Empty elements | I = [] | T = () | A = set()  B = set(A) | D = {} |

# OPERATIONS :-

**1) Arithmetic Operators:**

* **+** (addition)
* **-** (subtraction)
* **\*** (multiplication)
* **/** (division)
* **//** (integer division)
* **%** (modulo, remainder)
* **\*\*** (exponentiation)

**2) String Operations:**

* Concatenation using +
* Repetition using \*
* Indexing and slicing to access characters or substrings

**3)** **Comparison Operators:**

* **==** (equal)
* **!=** (not equal)
* **<** (less than)
* **>** (greater than)
* **<=** (less than or equal to)
* **>=** (greater than or equal to)

**4) Logical Operators:**

* **and** (logical AND)
* **or** (logical OR)
* **not** (logical NOT)

**5)** **Assignment Operators:**

* **=** (assign value)
* **+=**, **-=**, **\*=**, **/=** (modify and assign)

**# PRACTICE QUESTIONS :-**

1. Concatenate two strings to form a full name. Concatenate two strings to create a sentence and print it.
2. Check if 7 is greater than 3 and print the result. Determine if a given number is even or odd using the modulo operator.
3. Create a list of your favourite fruits and access the third element in the list.
4. Calculate the result of 7 to the power of 3 using the exponentiation operator.
5. Create a variable called age and assign your age to it. Print the value of age.
6. Calculate the area of a rectangle with a length of 5 and a width of 8. Store the result in a variable and print it.