

# Chapter 4: Variables

Tuesday, August 2, 2022

4:30 PM

## 1. Variables?

A variable is a name given to a memory location in a program. For an example:

```
name = "Randrita"
```

- **Variables** : Container to store a value
- **Keywords**: Reserve Words in Python
- **Identifier**: class/ function/ variable names

## 2. Variable Names:

- Variable names can consist of letters (a-z, A-Z), digits (0-9), and underscores (\_).
- Variable names must start with a letter or an underscore. They cannot start with a digit.
- Variable names are case-sensitive, so `myVar`, `myvar`, and `MYVAR` are considered different variables.

## 3. Reserved Words:

- You cannot use Python's reserved words (keywords) as variable names. For example, you cannot use `if`, `while`, `for`, `class`, `def`, etc., as variable names.

## 4. Valid Variable Names:

- Examples of valid variable names: `my\_variable`, `count`, `\_private\_variable`, `Var123`.
- Variable names should be descriptive and give a clear idea of what the variable represents. This makes your code more readable and maintainable.

## 5. Assignment

- Variables are created when you assign a value to them using the `=` operator. For example: `x = 5`.

## 6. Type Inference:

- Python is dynamically typed, which means you don't need to declare the data type of a variable explicitly. Python infers the data type based on the value assigned to it.

## 7. Variable Reassignment:

- You can change the value of a variable by assigning a new value to it. For example: `x = 10` and later `x = "Hello"` is valid.

## 8. Variable Names Convention:

- It's common to follow a naming convention for variables in Python:
- Use lowercase letters and underscores for variable names (e.g., `my\_variable`).
- Use uppercase letters for constants (e.g., `PI = 3.14159`).
- Use descriptive names that convey the purpose of the variable (e.g., `total\_count` instead of `tc`).

Happy Coding :) , with love Randrita :)