# **Day 2**

## Variables

Introduction to Variables

* Variables are like temporary storage spaces in a computer's memory.
* They hold data values, and when a variable's value changes, it affects the program's current state.

Functions of Variables

1. **Container for Data**: Variables serve as containers to store different data items or values.
2. **Data Transfer**: They are crucial for moving data between functions, a concept that will be discussed later.

Variable States

1. Variable creation (Declaration)
2. Variable assignment (Initialization)
3. Variable change (Execution)

Once the code creating the variable finishes executing, the variable **is destroyed**.

Defining Variables in Python

In Python, variables are defined using the assignment operator **=** This assigns a value to the variable.

Naming Conventions

Naming conventions define how variables should be named for readability and understanding.

Variable names can consist of upper or lower case letters (A–Z, a–z), numbers, or underscores, but cannot start with a number or contain spaces.

* Continuation characters can be any characters except whitespace characters like tabs and spaces.

Examples of Valid and Invalid Variable Names

**Valid Variable Names**

* c
* ref\_number
* admin
* aVeryLongName

**Invalid Variable Names**

* True
* $name
* 12Graph

Case Sensitivity and Keywords

* In Python, identifiers (including variable names) are case sensitive.
* Variables cannot have the same name as Python's keywords.
* You can check Python's keywords using the **dir(\_\_builtins\_\_)** function.

Special Identifiers

Identifiers starting and ending with one or two underscores are special methods.

**Note**: Methods, exceptions, and functions from **dir(\_\_builtins\_\_)**, **dir(\_\_doc\_\_)**, **dir(\_\_name\_\_)**, and **dir(\_\_package\_\_)** cannot be used as variable names.

## \*Assignment of variables

All variables in Python must be assigned a data type, such as string (sequence of characters) or integer (whole numbers, positive or negative).

Python features dynamic typing, where variables are assigned data types automatically. For instance, if an input or value contains letters or words, Python will assign the variable a string data type.

There are other data types:

Explicit Type Casting

**num1 = 15** #num1 is automatically assigned as an integer value

**num2 = 30** #num2 is also assigned as an integer value

**print (num1 + num2)** #answer is printed

## Casting

Implicit Casting

* Compiler automatically converts value from one data type to another when it's confident there won't be data loss.
* Example: Casting from integer to floating-point or from one integer variable to another.

Implicit Casting Values

**num1 = 15.60  # Floating point number**

**num2, num3 = 30, 32  # Integer variables**

**# Converted to floating point**

**ans = num1 + num2 + num3**

**print("The answer:", ans)**

Explicit Casting

* Value cannot be automatically cast if it leads to data loss. Extra code is required to ensure value remains same, only data type changes.
* Example: **Casting from floating-point value to an integer value.**

Explicit String and Integer Casting

**num1 = 15**

**num2 = "30"**

# concatenation vs. addition ouputs

# cast num2 to int

**ans = num1 + int(num2)**

**print("num2 cast to integer:", ans)**

# cast num1 to str

**ans = str(num1) + num2**

**print("num1 cast to string:", ans)**

Casting example

**i\_number = int(15)**

**s\_number = str("12.543")**

**s\_characters = str("five")**

**f\_number = float(123.5675)**

**f = float(i\_number)**

**print("An Integer cast as a floating point number:", f)**

**i = int(f\_number)**

**print("A floating point number cast as an Integer:", i)**

**f = float(s\_number)**

**print("A number string cast as a floating point number:", f)**

#ValueError occurs as 'five' can't be converted to float

**f = float(s\_characters)**

**print("A number string cast as a floating point number:", f)**

## Casting Program

**num1 = 2**

**num2 = 4**

**num3 = 3**

**add = 2 + 4 # answer = 6**

**subtract = add - num3** # answer = 3

**print(**

**"The first number is " + str(num1) + "\n" +**

**"The Second number is " + str(num2) + "\n" +**

**str(num1) + " + " + str(num2) + " = " + str(add) + "\n" +**

**str(add) + " - " + str(num3) + " = " + str(subtract) + "\n" +**

**"The concatenated result is: " + '"' + str(add) + str(subtract) + '"'**

**)**