What is programming?

Just like we use Nepali or English to communicate with each other, we use a programming language like **python** to communicate with the computer.

Programming is a way to instruct the computer to perform various tasks.

Introduction to Python

Python is one of the most popular programming languages. It’s simple to use, packed with features and supported by a wide range of libraries and frameworks. Its clean syntax makes it beginner-friendly.

- **Libraries:** A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs.

- **Frameworks:** a framework is a collection of modules, packages, and tools that provide a standardized structure and a set of pre-written components to facilitate the development of applications

01

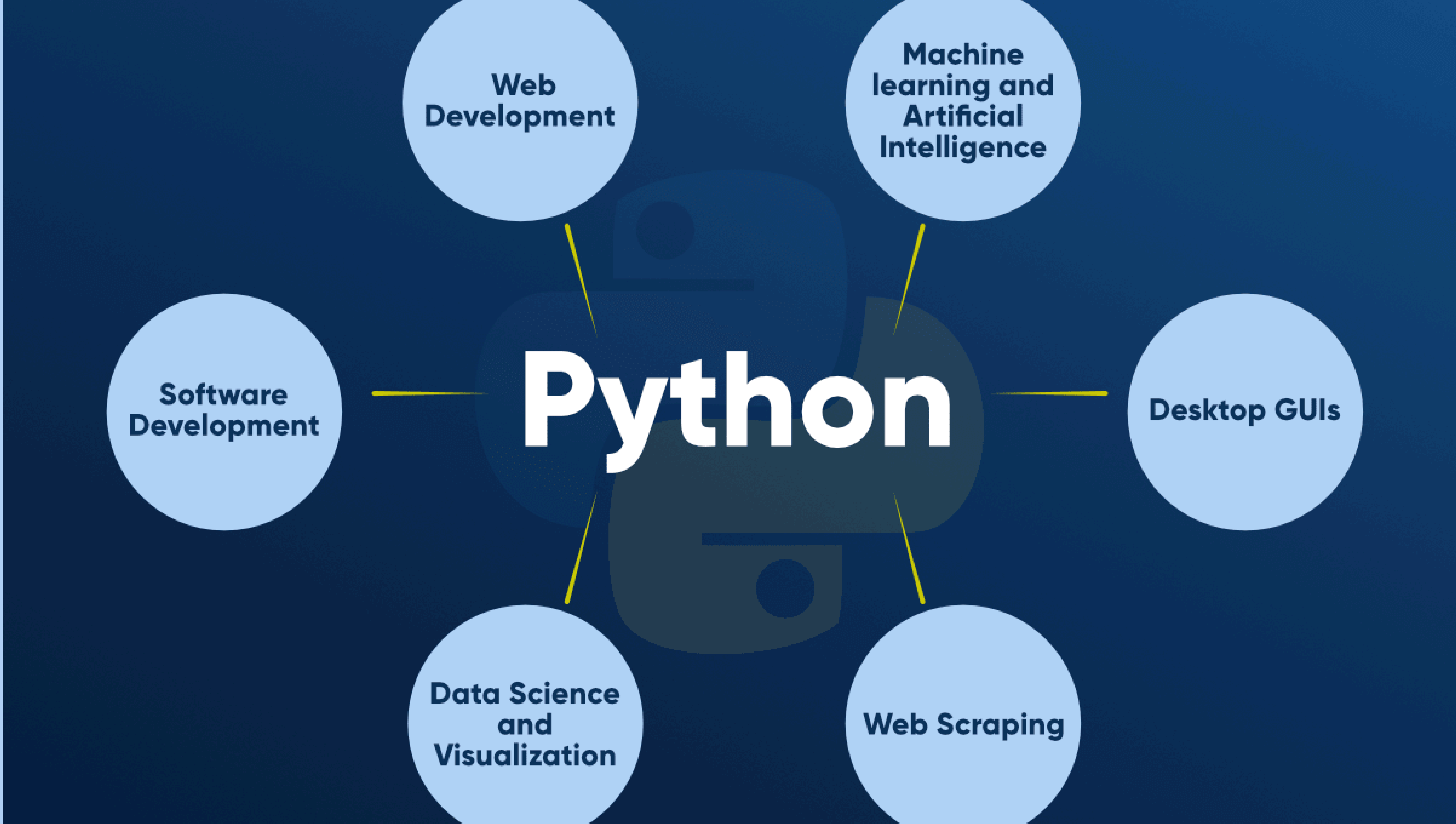
A high-level language, used in web development, data science, automation, AI and more.

02

Known for its readability, which means code is easier to write, understand and maintain.

03

Backed by library support, so we don’t have to build everything from scratch, there’s probably a library that already does what we need.



Python

Web Development

Machine learning and Artificial Intelligence

Software Development

Desktop GUIs

Data Science and Visualization

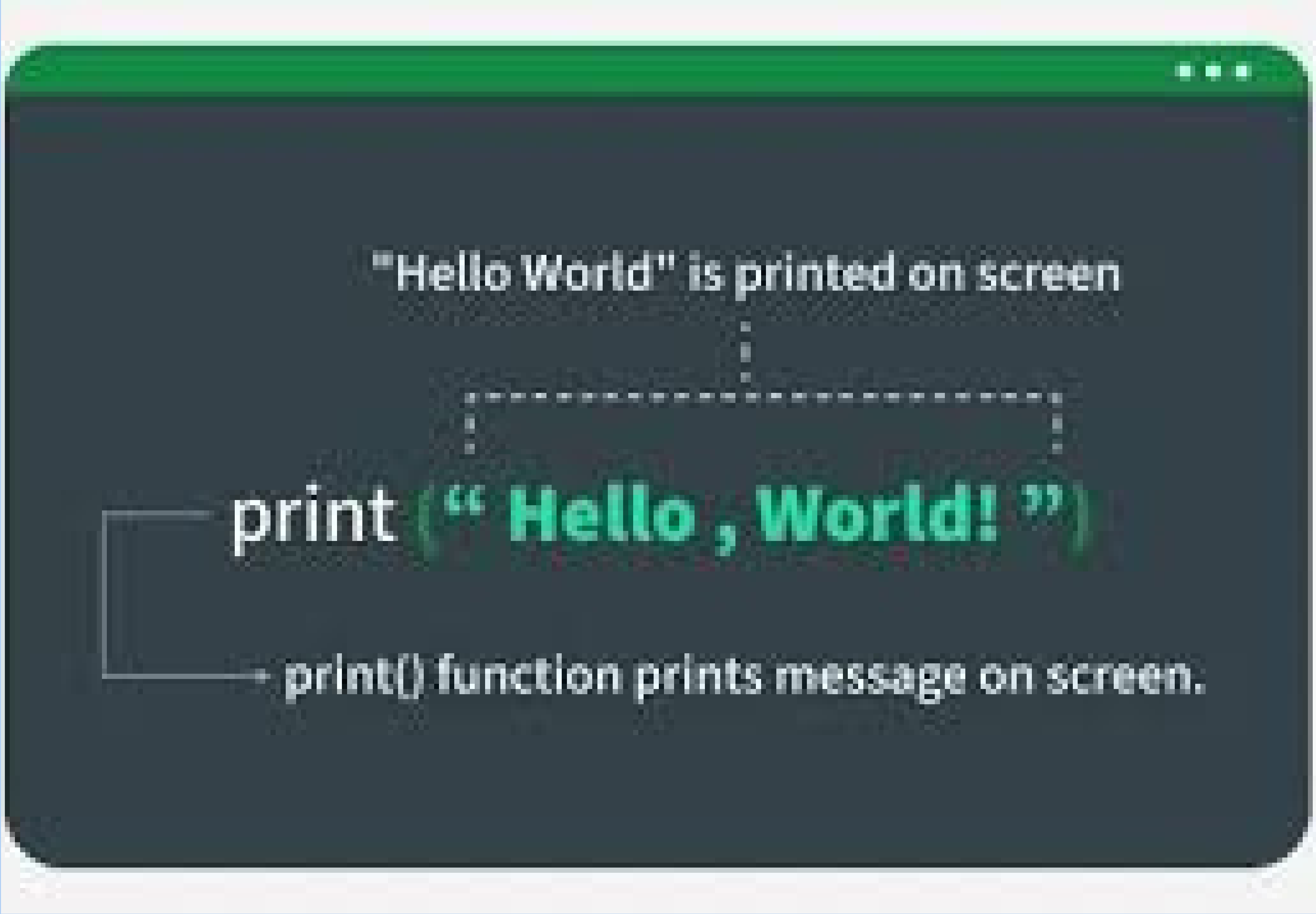
Web Scraping

Write your first python program

Print your name using **print()** function:
same like **print(“Hello world”)**

Output is appeared like
Hello world.

- **print()** is a built-in Python function that instructs the computer to display text on the screen.



```
print(“ Hello , World! ”)
```

"Hello World" is printed on screen
print() function prints message on screen.

Comments in python

- Anything after a **#** symbol is a comment
- Comments in Python can be used to explain any program code.
- It can also be used to hide the code as well.
- They are useful for explaining code to human readers.

```
# This is single line comment.
’’’This is
   multiline
   comment.’’’
```

Modules in Python

A module is the file containing code written by somebody else (usually) which can be imported and used in our programs.

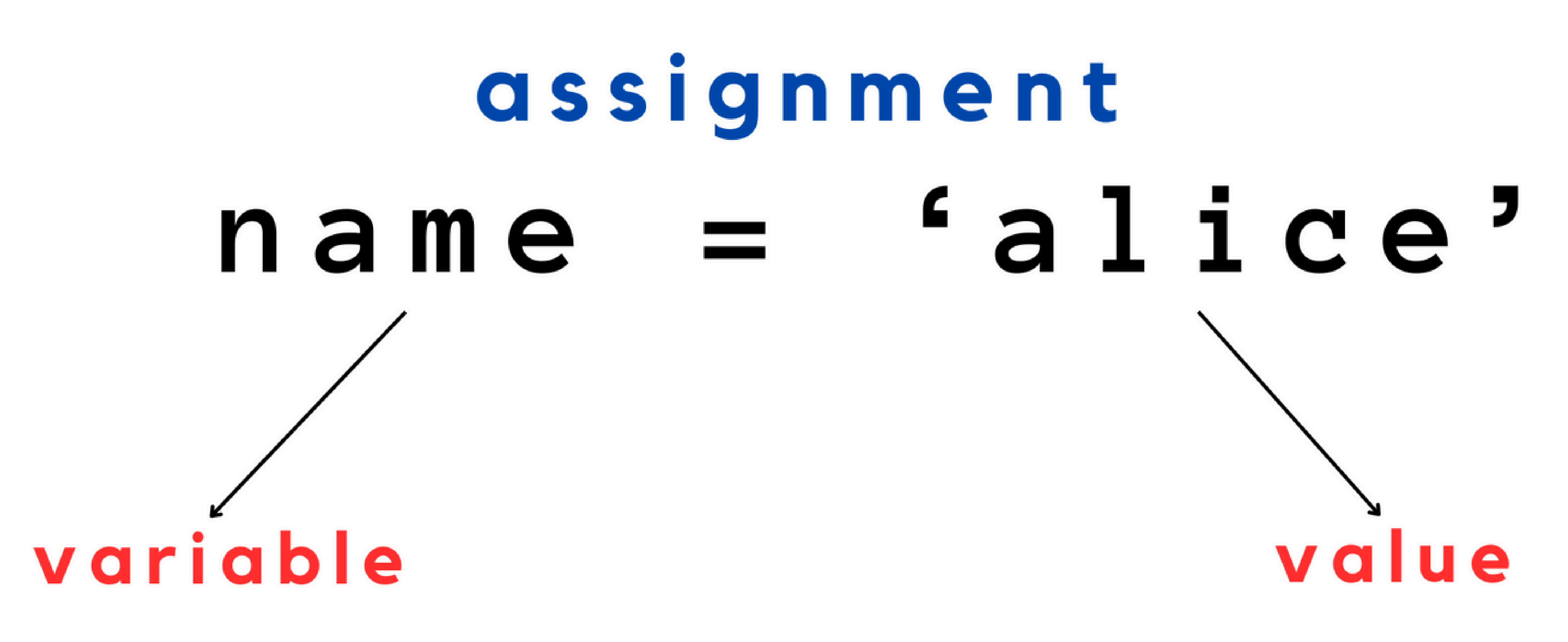
You can use pip to install module:

```
-> pip install module_name
```

Install one module and use it.

Python Variables

- A variable is a named memory location in which we can store values for the particular program.
- In Python, variables are used to store data that can be referenced and manipulated during program execution.
- A variable is essentially a name that is assigned to a value.



rules for naming variables:

- Variable names can be a group of both letters and digits, but they have to begin with a letter or an underscore.

- It is recommended to use lowercase letters for variable name. ‘SUM’ and ‘sum’ both are two different variables.

Variable Declaration in Python

○ Assign single value to multiple variables:

```
x = y = z = 50
```

○ Assign multiple values to multiple variables:

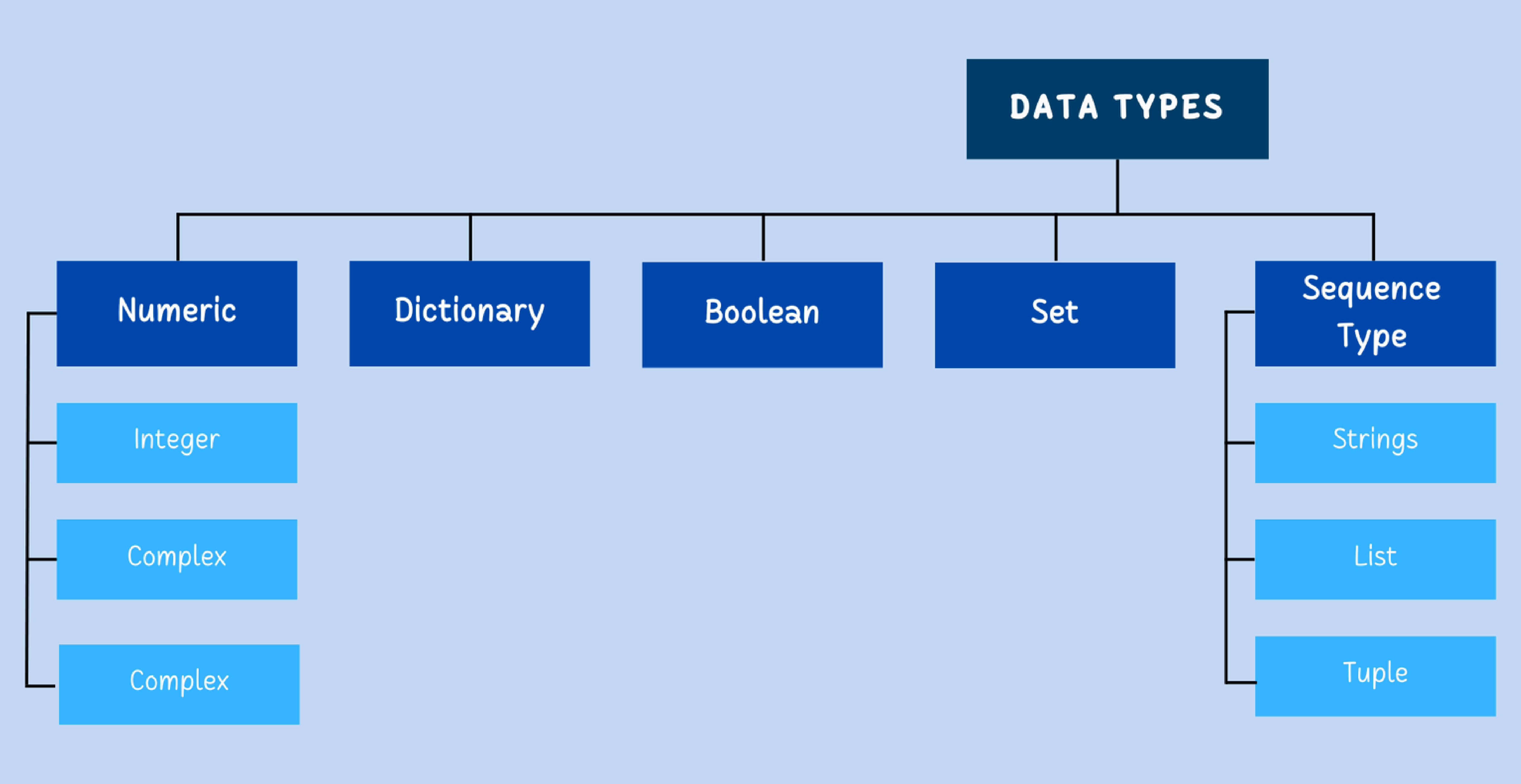
```
a, b, c = 80, 90, 100
```

Delete a Variable Using del Keyword:

→ **del x**

Data Types in python

- In general, Data Types specifies what type of data will be stored in variables. Variables can hold values of different data types



Name	Type	Description
Integers	int	Whole numbers, such as: **3 300 200**
Floating point	float	Numbers with a decimal point: **2.3 4.6 100.0**
Strings	str	Ordered sequence of characters: **"hello" 'Sammy' "2000" "楽しい"**
Lists	list	Ordered sequence of objects: **[10,"hello",200.3]**
Dictionaries	dict	Unordered Key:Value pairs: **{"mykey" : "value" , "name" : "Frankie"}**
Tuples	tup	Ordered immutable sequence of objects: **(10,"hello",200.3)**
Sets	set	Unordered collection of unique objects: **{"a","b"}**
Booleans	bool	Logical value indicating **True** or **False**

Type conversion:

Python provides Explicit type conversion functions to directly convert one data type to another. It is also called as **Type Casting** in Python

Python supports following functions:

1. **int ()**: This function convertsany data type to integer.
2. **float()**: This function is used to convert any data type to a floating point number.
3. **str()** : This function is used to convertany data type to a string.

Input and Output in Python

Understanding input and output operations is fundamental to Python programming. With the **print()** function, we can display output in various formats, while the **input()** function enables interaction with users by gathering input during program execution.

Taking input in Python

- Python's input() function is used to take user input. By default, it returns the user input in form of a string.

```
name = input("Enter your name: ")
print(name)
```

Operators

Python operators are special symbols or keywords that perform operations on variables and values.
They are categorized into several types:

1. Arithmetic Operators:

These perform mathematical calculations.

- + (Addition)
- - (Subtraction)
- * (Multiplication)
- / (Division)
- % (Modulus - remainder of division)
- ** (Exponentiation - raised to the power of)
- // (Floor Division - division that rounds down to the nearest whole number)

2. Assignment Operators:
These assign values to variables, often combining an operation and assignment.

- = (Assigns value)
- +=, -=, *=, /=, %=, **=, //= (Combine an arithmetic operation with assignment)

3. Comparison Operators:
These compare two values and return a Boolean result (True or False).

- == (Equal to)
- != (Not equal to)
- > (Greater than)
- < (Less than)
- >= (Greater than or equal to)
- <= (Less than or equal to)

4. Logical Operators:

These combine conditional statements and return a Boolean result.

- and (Returns True if both conditions are True)
- or (Returns True if at least one condition is True)
- not (Reverses the Boolean state of a condition)

5. Identity Operators:

These check if two variables refer to the exact same object in memory.

- is (Returns True if both variables point to the same object)
- is not (Returns True if both variables do not point to the same object)

6. Membership Operators:

These test whether a value exists within a sequence (like a string, list, or tuple).

- in (Returns True if a value is found in the sequence)
- not in (Returns True if a value is not found in the sequence)

Let’s Practice

- write a Program to input 2 numbers & print their sum.
- WAP to input side of a square & print its area.
- WAP to input 2 floating point numbers & print their average.
- WAP to input 2 int numbers, a and b. Print True if a is greater than or equal to b. If not print False
- Ask the user for their name and age, then display a welcome message.
- Take two numbers as input and display their sum, difference, product, and quotient.
- Ask the user for their age in years and convert it to months. (Assignment)