Data Types and Type Conversion

MUKESH KUMAR

AGENDA

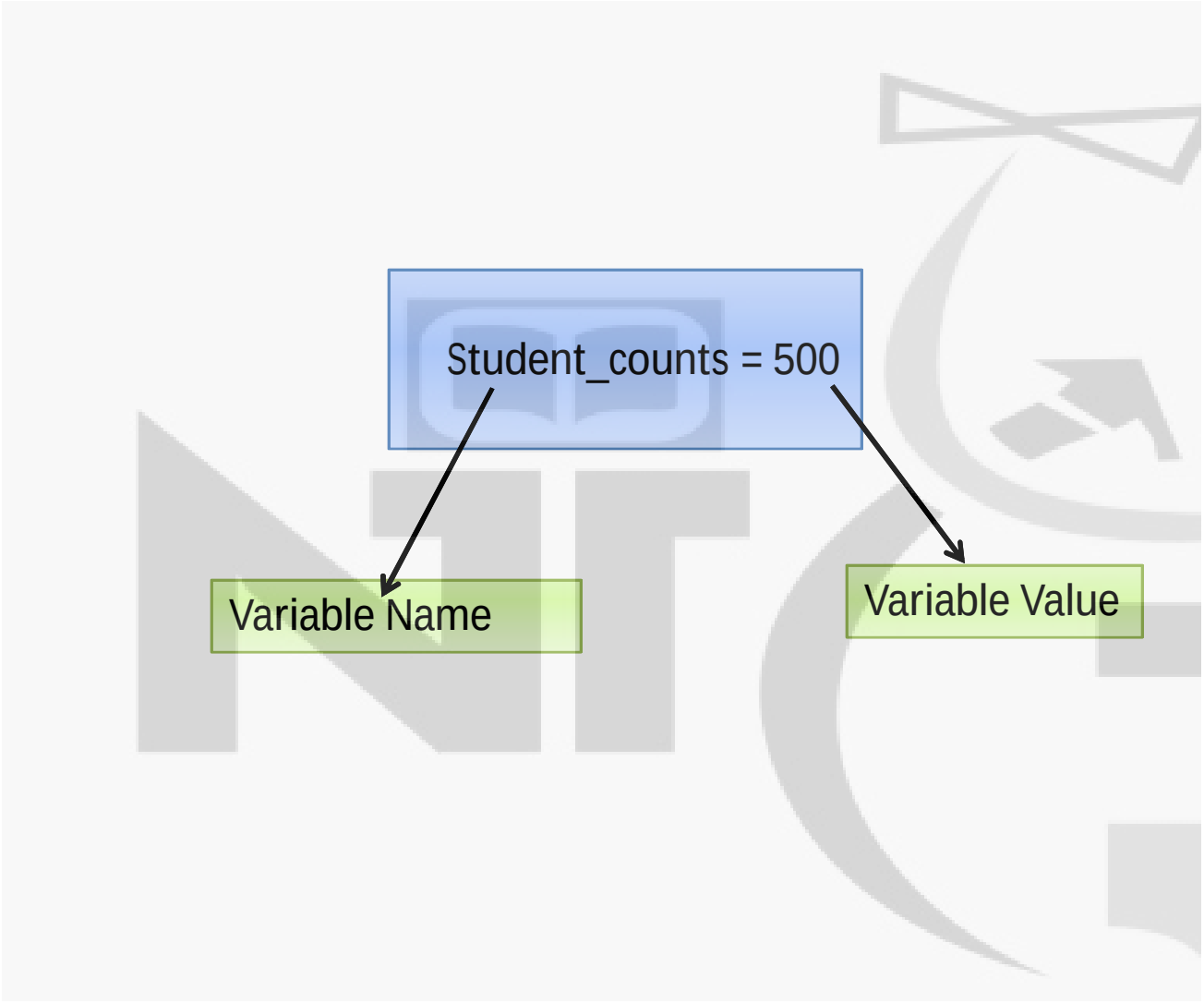
- Overview of basic data types
  - int, float, str, bool.
- Type conversion:
  - Implicit and explicit conversions.
  - Examples with int(), float(), str(), etc.

What is a Variable?

- Imagine a box. You can put anything inside that box – a toy car, a ball, or even some candy.

- Similarly, a variable is like a box in your computer's memory. You can store different types of information (like numbers, words, or even true/false values) inside it.

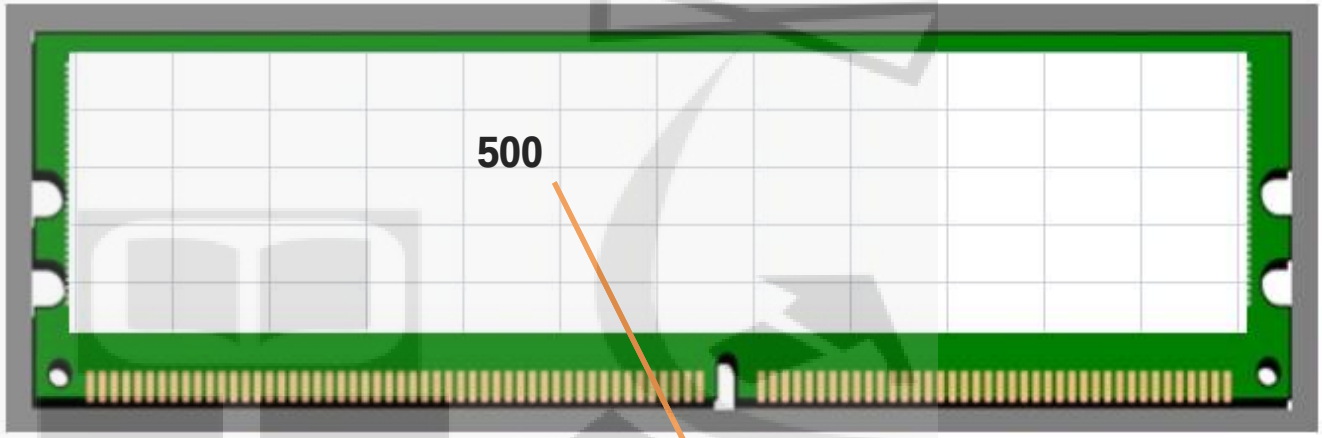
Variable



Variable

Student_counts = 500

Computer's Memory (RAM)



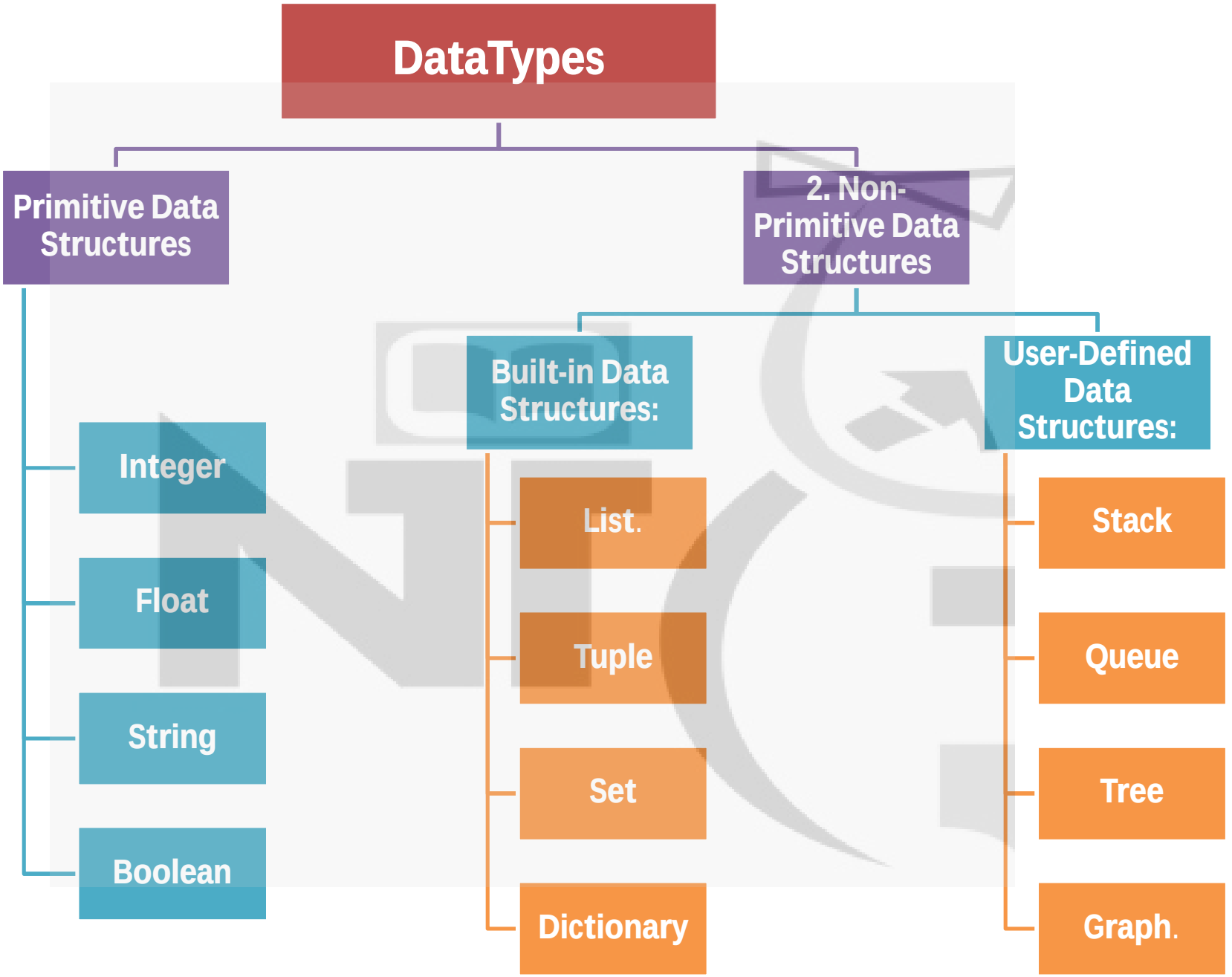
The variable **student_counts** points to that memory location. It acts as a shortcut for us to use the value 500.

The value **500** is saved in a specific memory location (like an address: e.g., 0x7ffabc123 in the computer's memory).

Understanding Data Types

- Data types define the kind of value a variable can store.
- Python has several built-in data types, including:
  - **int** (integer) – Whole numbers.
  - **float** (floating point) – Numbers with decimals.
  - **str** (string) – Sequences of characters.
  - **bool** (boolean) – Represents True or False values.

- Naming : Casing and format


DataTypes
Primitive Data Structures
2. Non-Primitive Data Structures
Integer
Float
String
Boolean
Built-in Data Structures:
List.
Tuple
Set
Dictionary
User-Defined Data Structures:
Stack
Queue
Tree
Graph.

Primitive Vs Non-Primitive Data Types

- **Primitive Data Structures**
  - **Basic Building Blocks:**
    - These are the simplest data structures that represent individual values.
  - **Directly Stored in Memory:**
    - They are stored directly in memory as a single unit.
  - **Examples:**
    - Integer (e.g., 10, -5), Float (e.g., 3.14, -2.5), Character (e.g., 'a', 'Z'), Boolean (True, False)

Primitive Vs Non-Primitive Data Types

- **Non-Primitive Data Structures**
  - **Collections of Primitive Data:**
    - These are more complex data structures that are made up of one or more primitive data types.
  - **Organize Data:**
    - They provide ways to organize and store collections of data.
  - **Examples:**
    - Arrays (lists, tuples), Linked Lists, Stacks, Queues, Trees, Graphs

What is Type Conversion?

- Type conversion is the process of converting one data type to another.
- Python supports both **implicit** and **explicit** type conversion.
- **Implicit conversion** happens automatically by Python.
- **Explicit conversion** requires the use of functions like int(), float(), str(), etc.

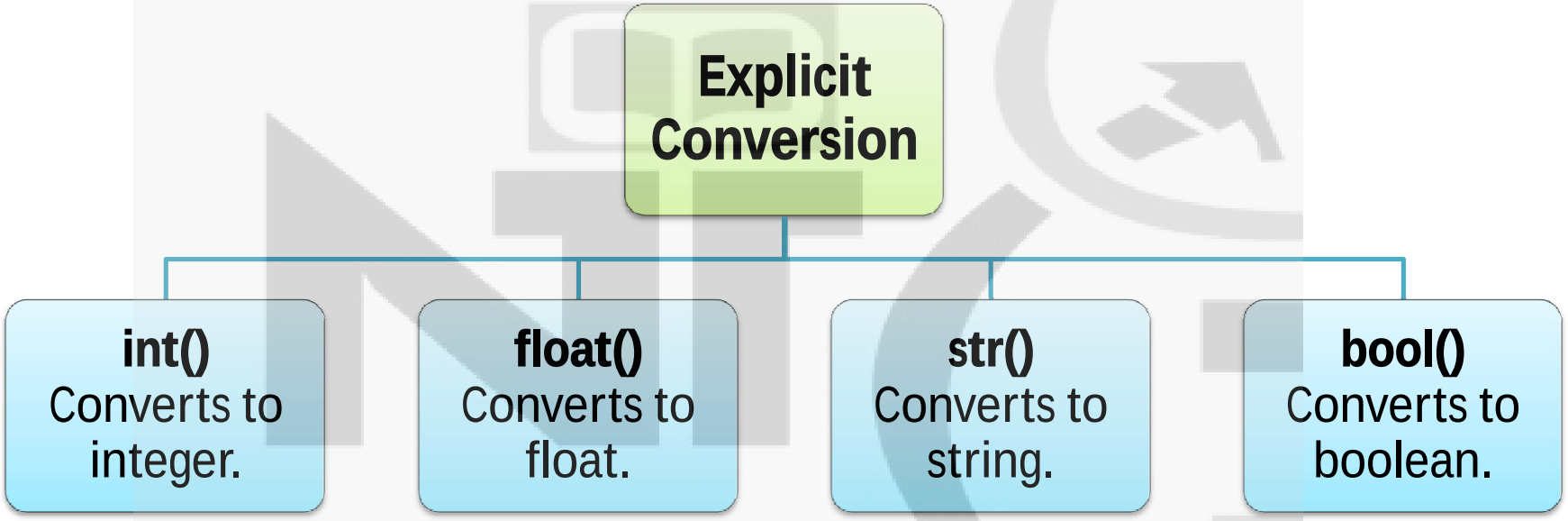
Implicit Type Conversion

- **Implicit type casting** in Python, happens when Python automatically converts one data type to another.
- Implicit casting ensures that the operation makes sense and doesn't lose precision. For example:
  - Adding an integer to a float results in a float because a float can represent fractional values, while an integer cannot.
  - Multiplying a float by an integer also results in a float.

Possible Implicit Conversion

From	To	When It Happens
int	float	During arithmetic with a float
int/float	complex	During operations with a complex number
bool	int	In arithmetic or logical operations
int	bool	In conditional checks (if, while)

Explicit Type Conversion



Examples of Type Conversion

- **int()**:Converts a string or float to an integer (if possible).
  - Converting a valid string to an integer
  - Converting a float to an integer
  - Converting an invalid string to an integer (Raises an error)
- **float()**:Converts a string or integer to a float.
  - Converting a valid string to a float
  - Converting an integer to a float
  - Converting an invalid string to a float (Raises an error)

Examples of Type Conversion

- **str()**:Converts any data type to a string.
  - Converting an integer to a string
  - Converting a float to a string
- **bool()**:Converts to True or False.
  - Anything non-zero or non-empty becomes True.
  - Example: bool(0) -> False, bool("Hello") -> True

Summary

- Python supports a variety of basic data types like int, float, str, and bool.
- **Implicit type conversion** is handled automatically by Python when required.
- **Explicit type conversion** uses built-in functions to convert data types.
- Familiarity with these conversions is essential for handling data effectively in Python.