Variables & Identifiers

MUKESH KUMAR

AGENDA

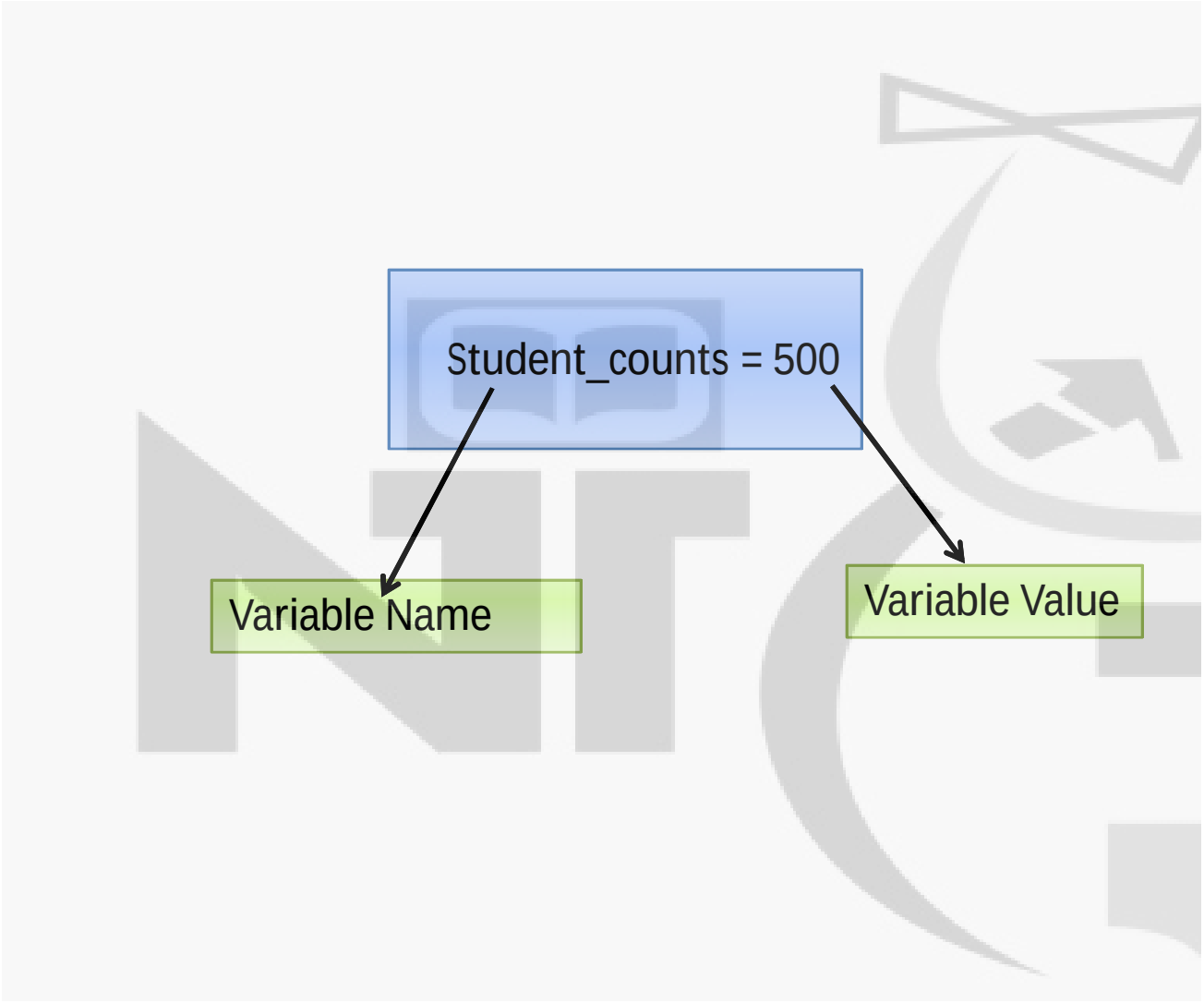
- What are variables
- Syntax
- Declaring and Initializing Variables
- Python Identifiers
- Rules and Naming Conventions
- Reserved Keywords in Python

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.

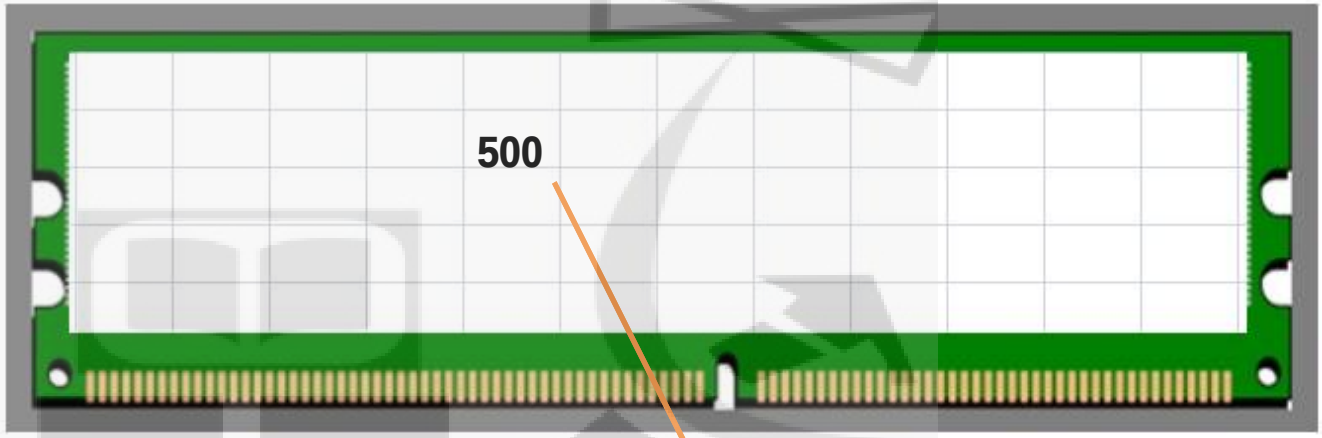
Variable



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).

Declaring Variables

Syntax:

- variable_name = value
- Example: age = 25

Variables can hold data of specific types (int, float, str, bool, etc.).

```
customer_count = 1000
rating = 4.99
is_stock_available = True
product_name = "Mobile"
```

Initializing Variables

- **What is Initialization?**
  - Initialization is the process of assigning an initial value to a variable when it is declared.

- if you reference a variable without assigning a value to it, it will result in a NameError.

```
# Declaration without initialization
x  # This will raise a NameError

# Correct way
x = 10  # Now x is initialized
```

Identifiers

- Identifiers are names used for variables, functions, classes, etc.

Rules for Python Identifiers

- **Case-Sensitive:**
  - **myVariable** and **myvariable** are different.
- **Start with a Letter or Underscore:**
  - Cannot start with a number (e.g., 1variable is invalid).
- **Can contain letters, numbers, and underscores:**
  - Examples: my_variable, count123, _private_var
- **Cannot use keywords:**
  - keywords are reserved words with special meanings in Python (e.g., if, for, while, def, class).

Naming Conventions (PEP 8)

- **Snake Case:** variable_name (**recommended**)
- **Camel Case:** variableName
- **Pascal Case:** VariableName (typically for class names)
- **Underscores for Private Variables:** _private_variable

What is PEP8?

- PEP 8 is the **Python Enhancement Proposal.**
- Provides guidelines and best practices on how to write clean, readable, and consistent Python code.
- It is the official style guide for Python programming, created to ensure that Python code remains uniform and easy to read across different projects and developers

Best Practices

- Use meaningful variable names (e.g., customer_name instead of c).
- Avoid single-letter variable names (except for temporary loop variables like i, j).
- Follow naming conventions consistently for better readability.

Understanding Reserved Keywords

- Reserved keywords have predefined meanings in Python.
- Cannot be used as identifiers.
- **Examples:**
  - Common keywords: if, else, while, for, def, return
  - Special keywords: None, True, False
- **Full List:** Use help("keywords") in Python to view all keywords.

List of Python Keywords

```
>>> help("keywords")

Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

Summary

- Variables store data and are initialized with values.
- Identifiers must follow naming rules and conventions.
- Reserved keywords cannot be used as identifiers.

Hands-on Practice

- Declare and initialize three variables: name, age, is_student.
- Print their values and types.
- Try using an invalid identifier and observe the error.
- Use help("keywords") to display Python's reserved keywords.