Python: Literals, Keywords, Data Types, and Identifiers

Literals in Python

- Literals are fixed values written directly in Python code.

Types of Literals in Python:

- Numeric Literals
  - Integers (int) → 5, -10, 0b1010 (binary), 0x1A (hexadecimal)
  - Floating-point (float) → 3.14, -2.5, 1.2e3 (scientific notation)
  - Complex numbers → 3 + 4j
- String Literals
  - Enclosed in 'single' or "double" quotes → "Hello", 'Python'
  - Multi-line strings → """Triple quotes""" or '''Triple quotes'''
  - Boolean Literals → True or False (case-sensitive)
  - None Literal → None (represents absence of value)
- Collection Literals
  - List → [1, 2, 3]
  - Tuple → (1, 2, 3)
  - Dictionary → {"key": "value"}
  - Set → {1, 2, 3}

Keywords in Python

Keywords are reserved words with special meanings. They cannot be used as variable names.

```
# you can get the list of keywords by

import keywords
print(keywords.kwlist)
```

Data Types in Python

Python has dynamic typing (variables can change types).

Basic Datatypes:

- int → for integers
- float → for float values
- bool → for boolean values
- str → for strings
- None → for None type

Collection Datatypes:

- list → [1 , 2 , 3]
- tuple → (1 , 2 , 3)
- set → { 1 , 2 , 3 }
- dict → { “a” : 1, “b” : 2, “c” : 3}

Identifiers in Python

- Identifiers are names given to variables, functions, classes, etc.

Rules for Naming Identifiers:

- Must start with a letter (a-z, A-Z) or underscore _.
  - Valid: name, _count, Age1
  - Invalid: 1var, @email
- Can contain letters, digits (0-9), and underscores (_).
  - Valid: user_name, total2
  - Invalid: user@name, first name
- Cannot be a Python keyword.
  - Invalid: if, for, class
  - Case-sensitive → Age ≠ age
  - No special symbols (@, #, $, etc.).

- Recommended: Use snake_case for variables/functions (user_age) and PascalCase for classes (ClassName).