▧ Chapter 2: Basic Syntax, Operators, and Data Type Introduction

1. Python Syntax Overview

```
☑ Key Points:
   - Case-sensitive (age and Age are different).
   - Indentation (usually 4 spaces) is mandatory in Python.
   - No need for {} or ; like C/C++.
   - Use # for single-line comments and triple quotes (''' or """) for multi-line
strings or docstrings
      -  #This is a comment
         print("Hello, Deepak!")  # Indentation is required in blocks
```

- **Why does Python enforce indentation?**
  - For cleaner syntax and to improve readability. It's part of Python's design philosophy (PEP 8).

2. Python Keywords

```
These are reserved words and cannot be used as variable names.
    - and, or, not, if, else, elif, while, for, break, continue, def, class, try,
except, True, False,    None, is, in, import, pass, return, lambda.
    - import keyword
      print(keyword.kwlist)
```

3. Python Identifiers

```
- Valid names for variables, functions, classes.
- Must begin with a letter (A-Z, a-z) or underscore (_).
- Cannot start with a digit.
- Cannot use keywords as identifiers.
```

4. Python Data Types

```
    Type          Example               Description
    int           5, -10                  Integer
    float       3.14, -0.5              Decimal numbers
    bool        True, False              Boolean values
    str          "Hello"                Text or characters
    list         [1, 2, 3]           Ordered, mutable sequence
    tuple        (1, 2, 3)          Ordered, immutable sequence
```

```
dict         {"a": 1}              Key-value pairs
set          {1, 2, 3}          Unordered, no duplicates
None           None                Null-like object
```

5. Type Conversion

```
◇ Implicit:
   a = 10
   b = 2.5
   print(a + b)  # a becomes float → 12.5

◇ Explicit (Type Casting):
   x = int("123")     # String → Integer
   y = str(3.14)      # Float → String
```

What is the difference between implicit and explicit conversion?

- Implicit conversion is done automatically by Python to prevent data loss in operations, while explicit conversion is done manually by the programmer to control how data types are changed.

6. Python Operators

A. Arithmetic Operators

```
Operator   Example    Result
   +        2 + 3        5
   -        5 - 2        3
   *        4 * 2        8
   /        5 / 2      2.5
   //      5 // 2        2
   %        5 % 2        1
   **      2 ** 3        8
```

B. Comparison Operators

```
==, !=, >, <, >=, <=
```

C. Logical Operators

```
Operator              Example                    Meaning
   and            x > 0 and x < 10             Both True
   or              x > 0 or x < 0             Either True
   not                not(x > 5)                 Negation
```

D. Assignment Operators

```
x = 5
x += 1  # x = x + 1
Also: -=, *=, /=, //=, %= etc.
```

E. Membership & Identity

```
- Membership checks if a value exists within a sequence (in, not in), while
identity checks if two variables point to the same object in memory (is, is not)
- x = [1, 2, 3]  #membership
  print(2 in x)  # True

- a = [1, 2] #Identity
  b = a
  print(a is b)  # True (same object)
```