1. Variables in Python

1.1 What is a Variable?

- A variable is a name that refers to a value stored in memory.
- Python is dynamically typed—no need to declare type explicitly.

```
name = "Alice"
age = 25
price = 99.99
```

1.2 Rules for Naming Variables

- Must start with a letter or underscore.
- Can contain letters, digits, underscores.
- Cannot use Python keywords (`if`, `else`, `True`, etc.).
- Case-sensitive: `Name` and `name` are different.

```
valid_name = "ok"
_invalid = "still ok"
3name = "invalid"   # SyntaxError
```

2. Data Types in Python

2.1 Overview of Standard Data Types

Type	Description	Mutable	Example	Use Case / Purpose

`int`	Integer values	No	`x = 10`	Counting, indexing, math operations
`float`	Decimal numbers	No	`pi = 3.14`	Scientific, financial, and precise calculations
`complex`	Complex numbers (real + imaginary)	No	`z = 2 + 3j`	Engineering, scientific computing
`str`	Sequence of Unicode characters	No	`name = "Alice"`	Text manipulation, messages, file I/O
`list`	Ordered, mutable sequence	Yes	`fruits = ["apple", "mango"]`	Storing collections of related items
`tuple`	Ordered, immutable sequence	No	`colors = ("red", "blue")`	Fixed data (e.g., coordinates, constants)
`set`	Unordered, no duplicates	Yes	`unique = {1, 2, 3}`	Removing duplicates, mathematical operations
`dict`	Unordered collection of key-value pairs	Yes	`student = {"name": "John"}`	Structured data, fast lookup by key

3. Type Casting (Conversion)

3.1 Converting Between Data Types

```
x = int("10")
y = float("5.5")
z = str(123)
```

4. Type Checking Functions

4.1 `type()` and `isinstance()`

```
x = 5
print(type(x))          # <class 'int'>
print(isinstance(x, int))  # True
```

5. Summary Table

Data Type	Syntax Example	Mutable	Notes
int	`x = 10`	No	Whole number
float	`x = 3.14`	No	Decimal number
str	`x = "abc"`	No	String (text)
list	`[1, 2, 3]`	Yes	Changeable, ordered
tuple	`(1, 2, 3)`	No	Immutable, ordered
set	`{1, 2, 3}`	Yes	Unordered, no duplicates
dict	`{"a": 1}`	Yes	Key-value pairs