1. What's being introduced?

The instructor is starting a new section about **data types** in Python.
They say when you learn a programming language, you mainly learn **two things**:

1. **Data types** → what kind of data is being stored (numbers, text, etc.).
2. **How to manipulate data** → how to work with that data (add numbers, change text, validate emails, etc.).

---

2. Examples of Data Types

- **Numbers**: integers (2, 10), decimals (3.14), imaginary numbers (complex numbers with `i` or `j` in math).
- **Strings**: text like `"Hitesh"`, `"chai"`, `"Python"`.
- **Other types** exist, but these are the basics.

---

3. Objects in Python

The teacher then introduces a **core Python concept**:
👉 *"Everything in Python is an object."*

An **object** has 3 key properties:

1. **Identity** → a unique ID in memory (like its address).
2. **Type** → what kind of object it is (int, str, list, etc.).
3. **Value** → the actual content (like `2`, `"chai"`, etc.).

So if you create `x = 2`, Python sees it as:

- Identity → memory address where `2` is stored.

- Type → `int` (integer).
- Value → `2`.

---

4. Mutability vs Immutability

This is a **very important concept**:

Mutable objects → can be changed in place (e.g., lists, dictionaries).
Example:

```
my_list = [1, 2, 3]
my_list.append(4)  # same object, just changed
```

-

Immutable objects → cannot be changed once created (e.g., strings, numbers, tuples).
Example:

```
x = "hello"
x = "world"   # a new object is created, not changing "hello"
```

-

The teacher says:
🔑 Don't check mutability by looking at values. Instead, check if the **object identity changes**.

- If identity stays same → object was modified (mutable).
- If identity changes → a new object was created (immutable).

---

5. First Python Program Example

They give a simple example:

```
sugar = 2
print(f"initial sugar: {sugar}")
```

- `sugar` is a variable storing a number.
- `f"..."` is an f-string, which allows inserting variables directly into text using `{}`.

Output:

```
initial sugar: 2
```

If you change `sugar = 12`, output becomes `initial sugar: 12`.

Behind the scenes:

- Python stores `2` in memory.
- `sugar` is just a label (variable) pointing to that memory location.

✅ **In summary**:

- You're learning the basics of **data types**.
- In Python, **everything is an object** → has identity, type, and value.
- Objects can be **mutable (changeable)** or **immutable (unchangeable)**.
- First code example shows variables + f-strings.