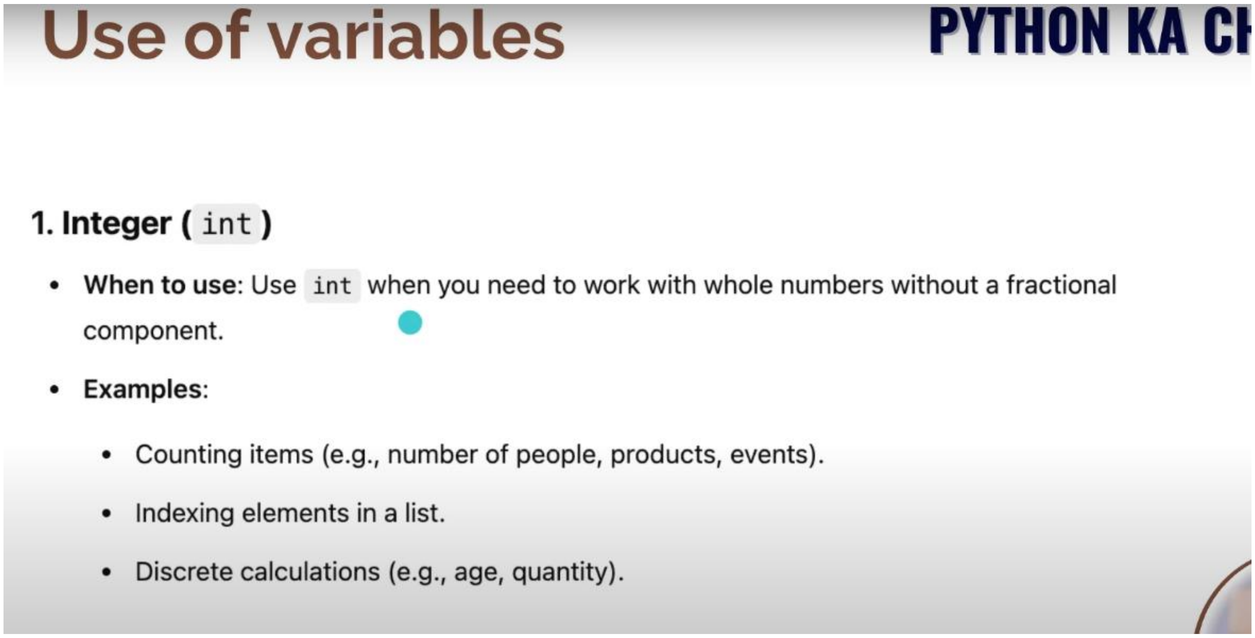
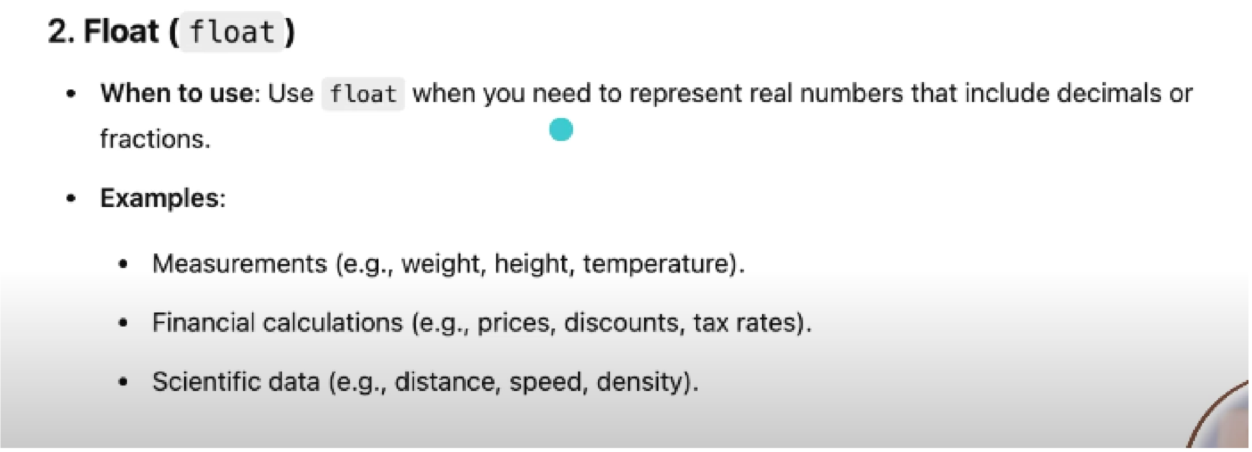


Use of variables

PYTHON KA CH

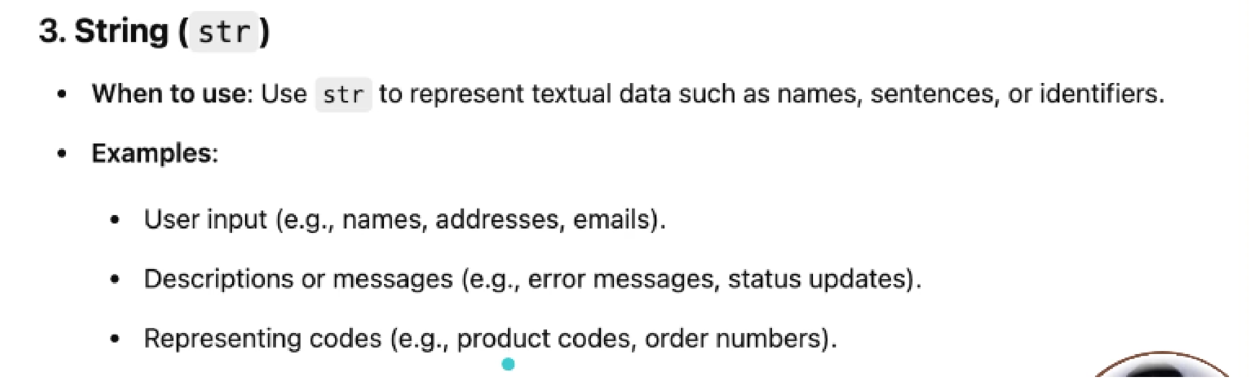
1. Integer (`int`)

- **When to use**: Use `int` when you need to work with whole numbers without a fractional component.
- **Examples**:
  - Counting items (e.g., number of people, products, events).
  - Indexing elements in a list.
  - Discrete calculations (e.g., age, quantity).



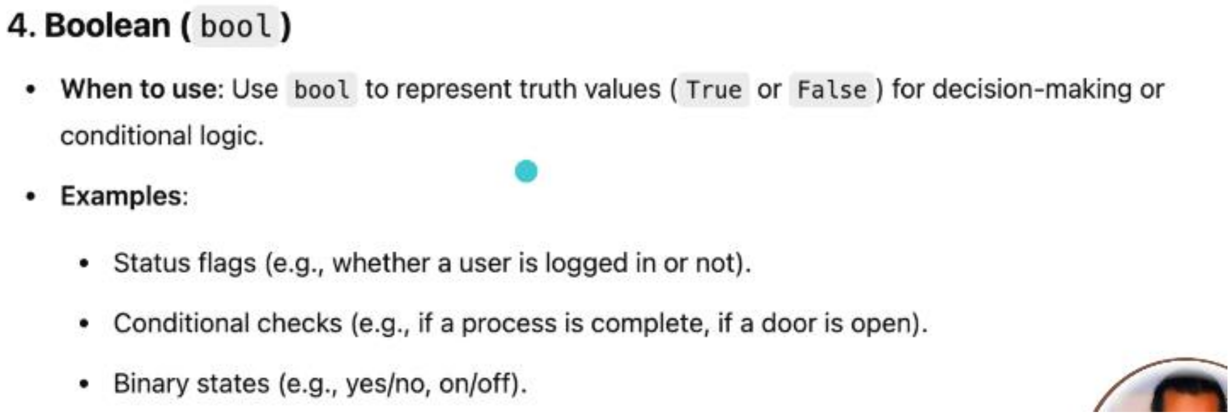
2. Float (`float`)

- **When to use:** Use `float` when you need to represent real numbers that include decimals or fractions.
- **Examples**:
  - Measurements (e.g., weight, height, temperature).
  - Financial calculations (e.g., prices, discounts, tax rates).
  - Scientific data (e.g., distance, speed, density).



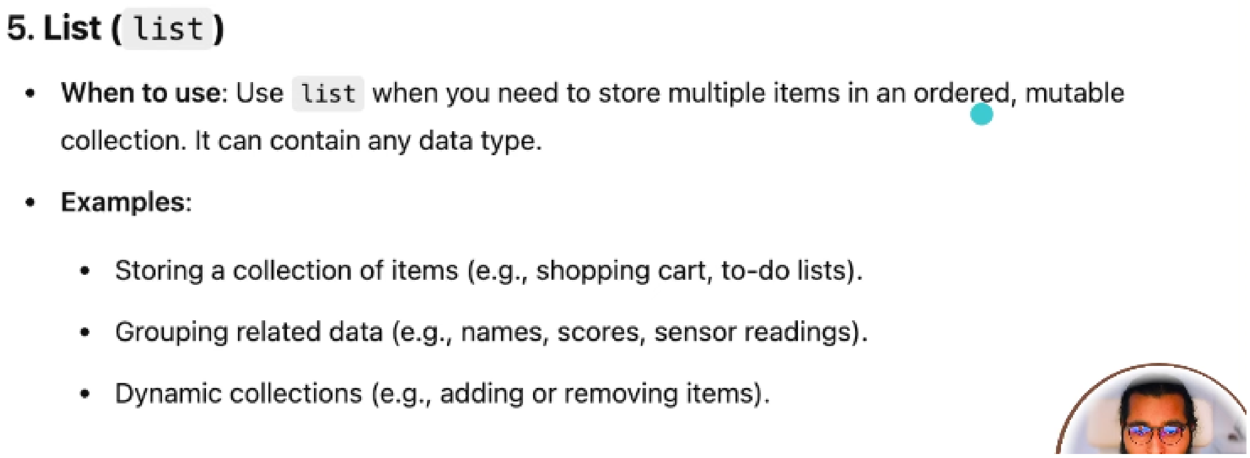
3. String (`str`)

- **When to use**: Use `str` to represent textual data such as names, sentences, or identifiers.
- **Examples:**
  - User input (e.g., names, addresses, emails).
  - Descriptions or messages (e.g., error messages, status updates).
  - Representing codes (e.g., product codes, order numbers).



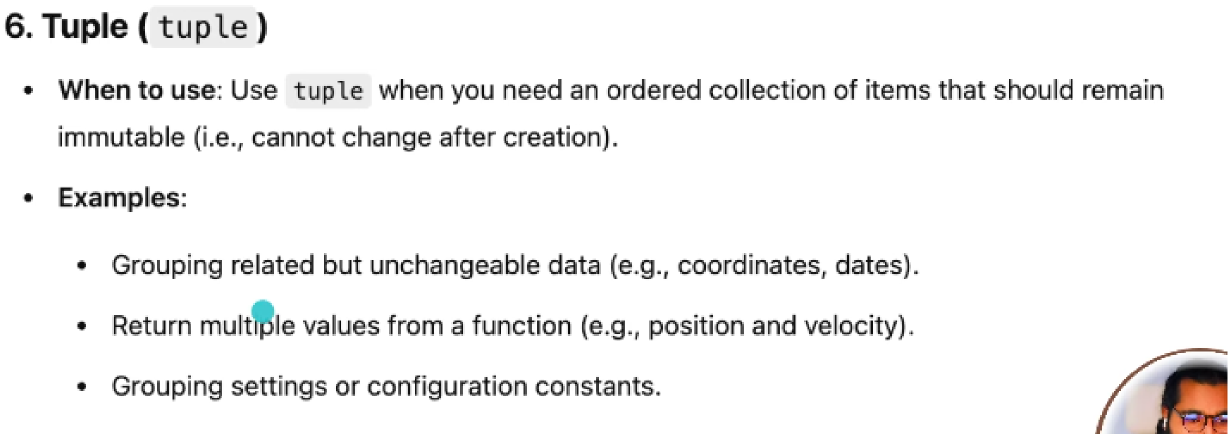
4. Boolean (`bool`)

- **When to use**: Use `bool` to represent truth values (`True` or `False`) for decision-making or conditional logic.
- **Examples**:
  - Status flags (e.g., whether a user is logged in or not).
  - Conditional checks (e.g., if a process is complete, if a door is open).
  - Binary states (e.g., yes/no, on/off).



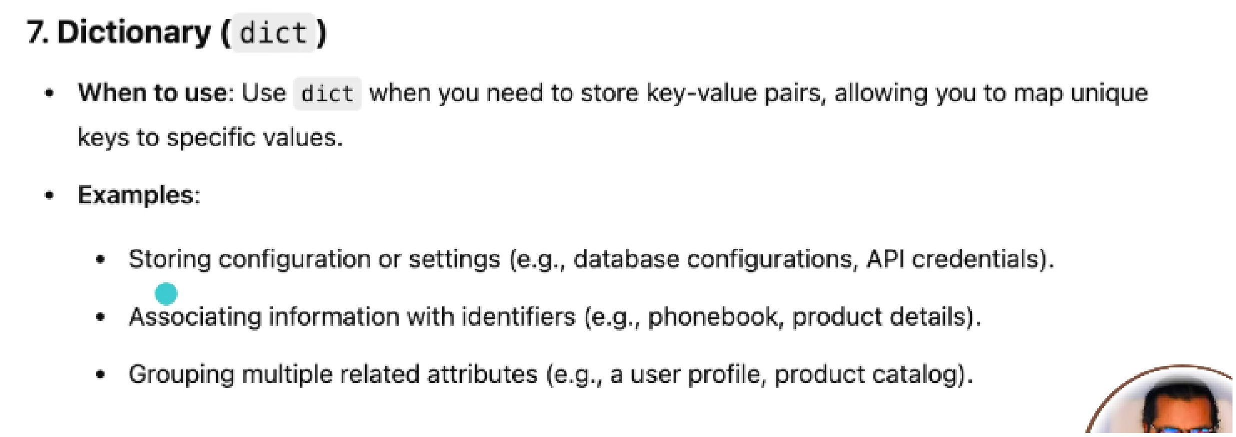
5. List (`list`)

- **When to use**: Use `list` when you need to store multiple items in an ordered, mutable collection. It can contain any data type.
- **Examples**:
  - Storing a collection of items (e.g., shopping cart, to-do lists).
  - Grouping related data (e.g., names, scores, sensor readings).
  - Dynamic collections (e.g., adding or removing items).



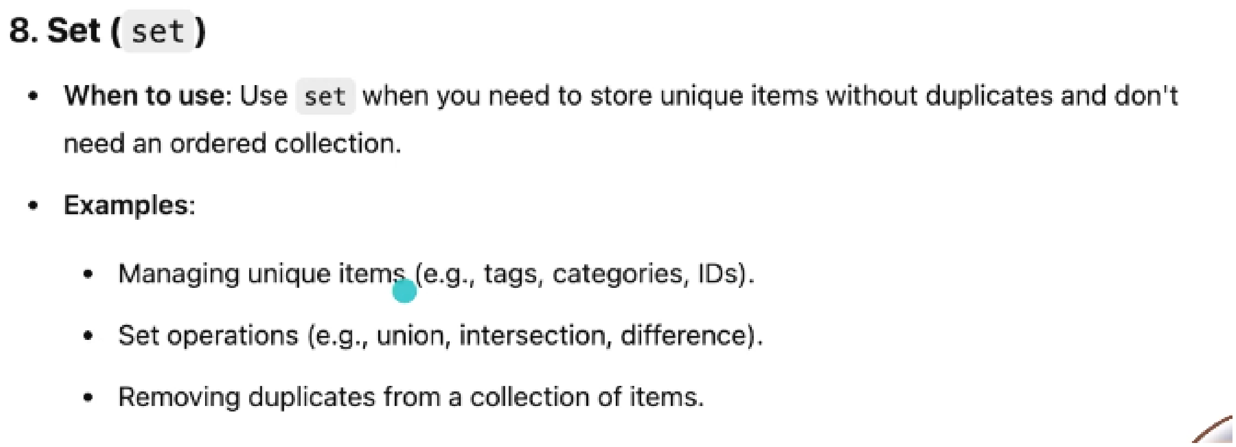
6. Tuple (`tuple`)

- **When to use**: Use `tuple` when you need an ordered collection of items that should remain immutable (i.e., cannot change after creation).
- **Examples**:
  - Grouping related but unchangeable data (e.g., coordinates, dates).
  - Return multiple values from a function (e.g., position and velocity).
  - Grouping settings or configuration constants.



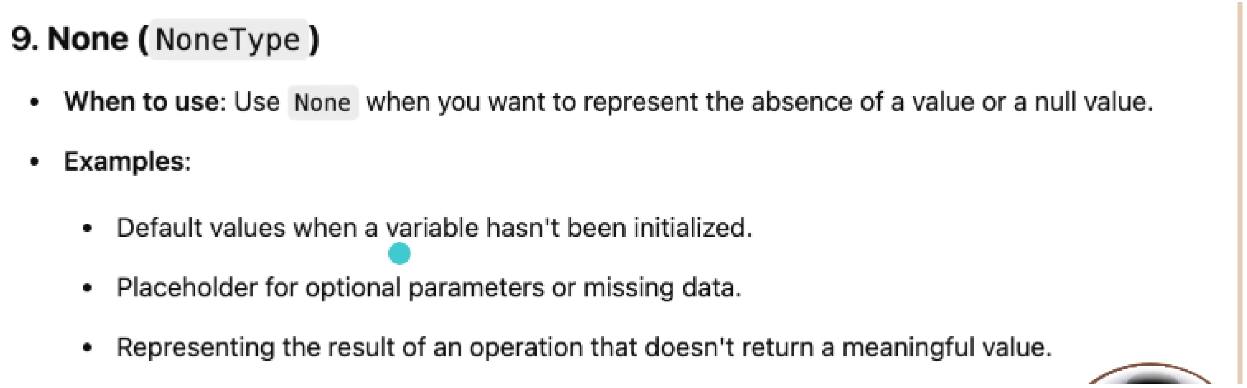
7. Dictionary (`dict`)

- **When to use**: Use `dict` when you need to store key-value pairs, allowing you to map unique keys to specific values.
- **Examples**:
    - Storing configuration or settings (e.g., database configurations, API credentials).
    - Associating information with identifiers (e.g., phonebook, product details).
    - Grouping multiple related attributes (e.g., a user profile, product catalog).



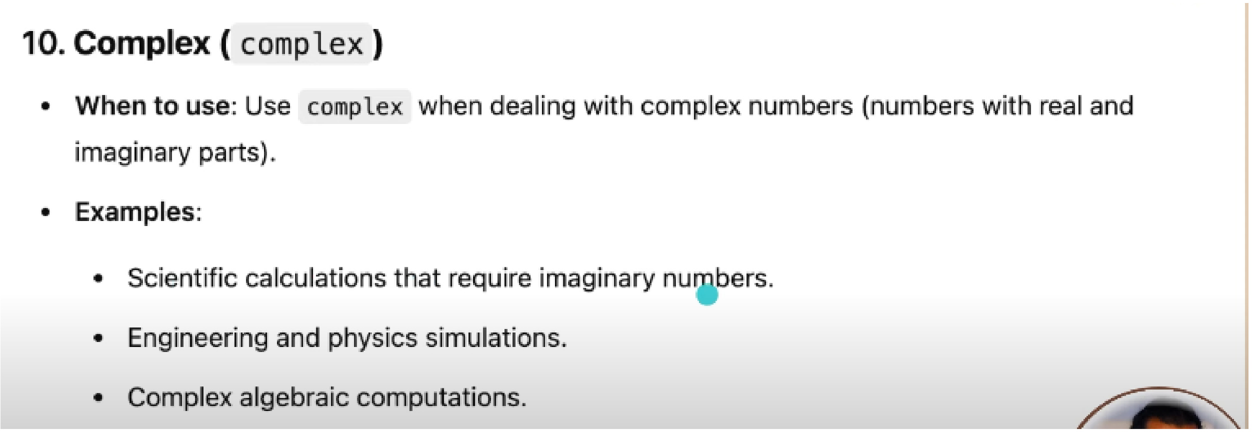
8. Set (`set`)

- **When to use:** Use `set` when you need to store unique items without duplicates and don't need an ordered collection.
- **Examples**:
    - Managing unique items (e.g., tags, categories, IDs).
    - Set operations (e.g., union, intersection, difference).
    - Removing duplicates from a collection of items.



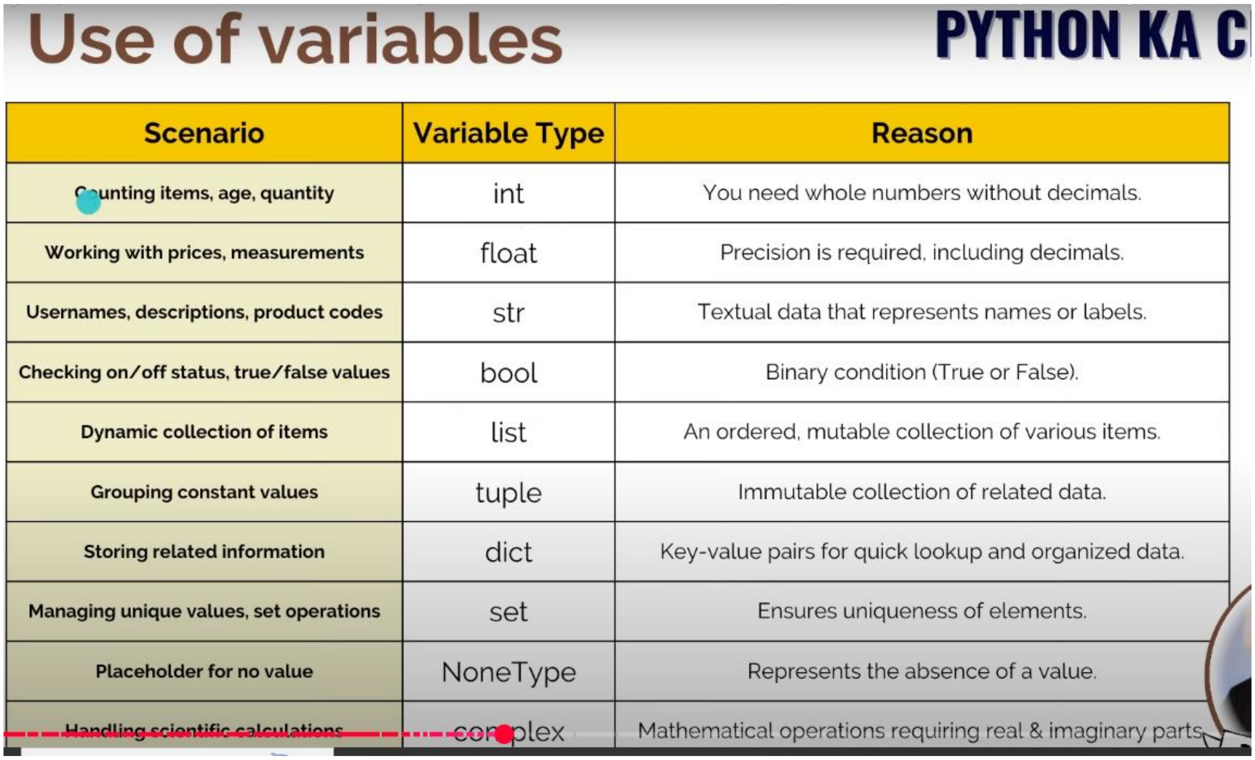
9. None (`NoneType`)

- **When to use**: Use `None` when you want to represent the absence of a value or a null value.
- **Examples**:
    - Default values when a variable hasn't been initialized.
    - Placeholder for optional parameters or missing data.
    - Representing the result of an operation that doesn't return a meaningful value.



10. Complex (`complex`)

- **When to use**: Use `complex` when dealing with complex numbers (numbers with real and imaginary parts).
- **Examples**:
  - Scientific calculations that require imaginary numbers.
  - Engineering and physics simulations.
  - Complex algebraic computations.



Use of variables

PYTHON KA C

Scenario	Variable Type	Reason
Counting items, age, quantity	int	You need whole numbers without decimals.
Working with prices, measurements	float	Precision is required, including decimals.
Usernames, descriptions, product codes	str	Textual data that represents names or labels.
Checking on/off status, true/false values	bool	Binary condition (True or False).
Dynamic collection of items	list	An ordered, mutable collection of various items.
Grouping constant values	tuple	Immutable collection of related data.
Storing related information	dict	Key-value pairs for quick lookup and organized data.
Managing unique values, set operations	set	Ensures uniqueness of elements.
Placeholder for no value	NoneType	Represents the absence of a value.
Handling scientific calculations	complex	Mathematical operations requiring real & imaginary parts