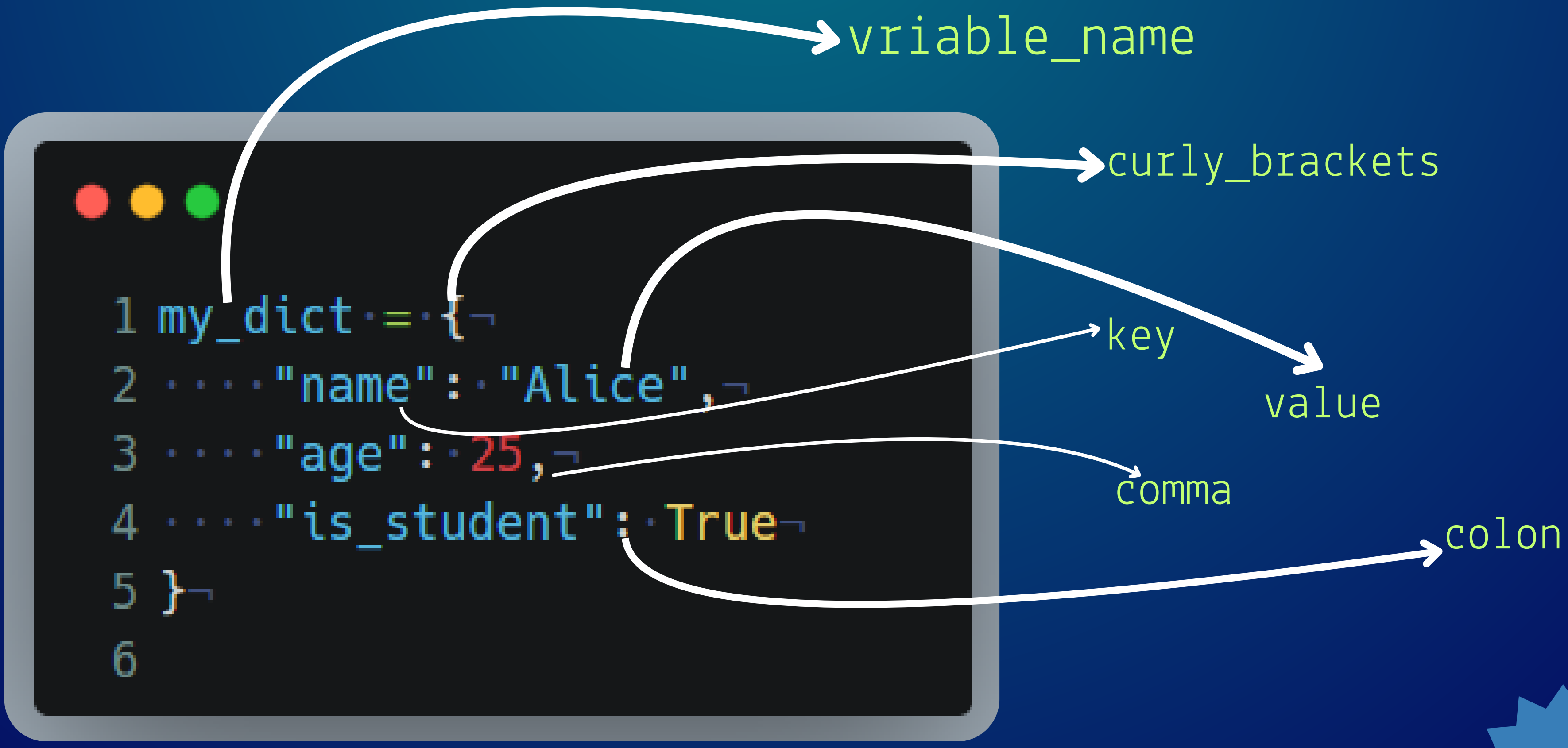
Python Dictionary - Your Complete Guide

- A **Dictionary** in Python is a collection of *key-value* pairs.
- Keys must be **unique** and immutable (*string, number, tuple*).
- Values can be of any data type (*string, list, number*, etc.).

Example →

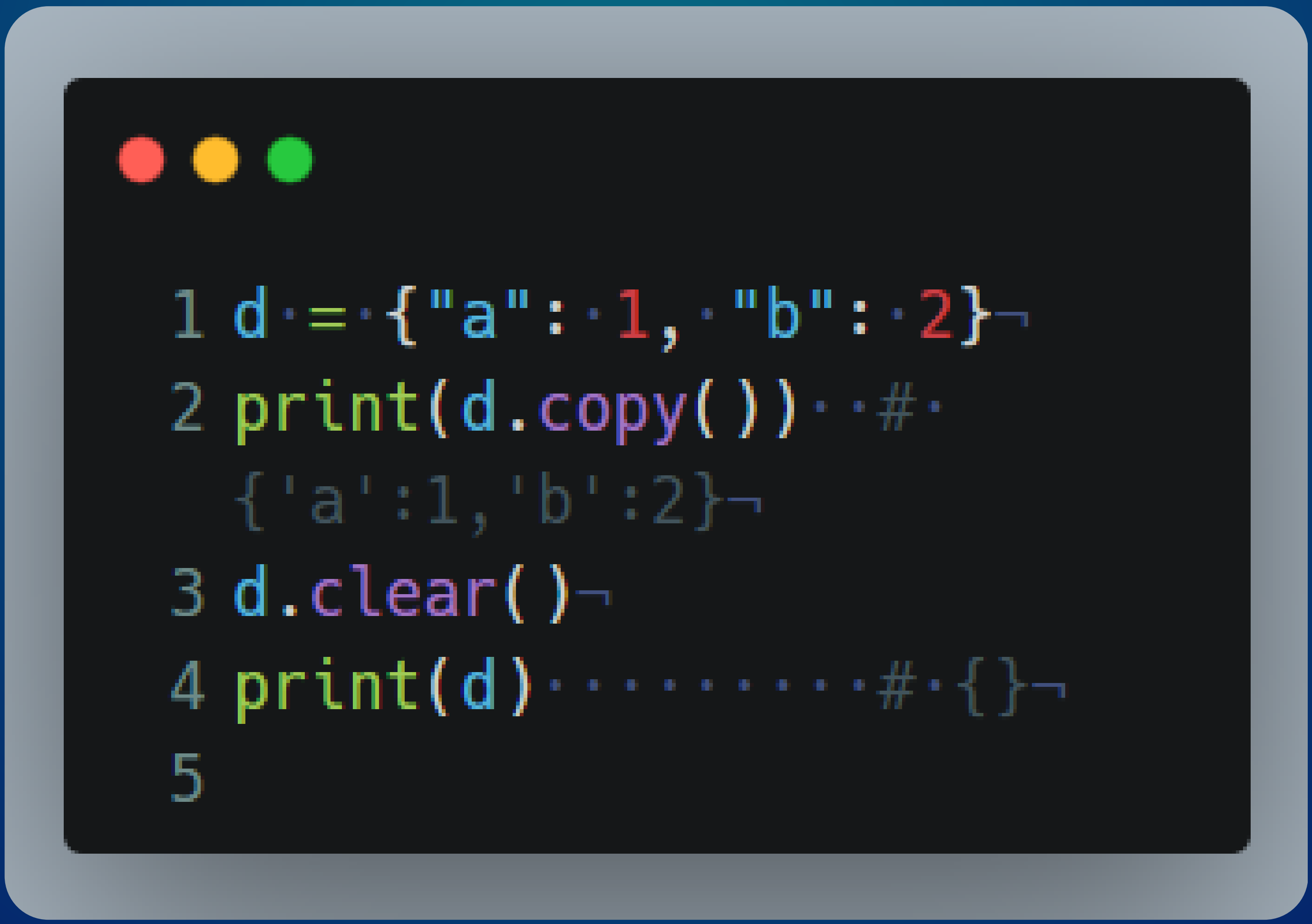


```python
my_dict = {
    "name": "Alice",
    "age": 25,
    "is_student": True
}
```

1) clear() + copy()

- ◆ clear() → Removes all items.
- ◆ copy() → Returns a shallow copy.

Example →



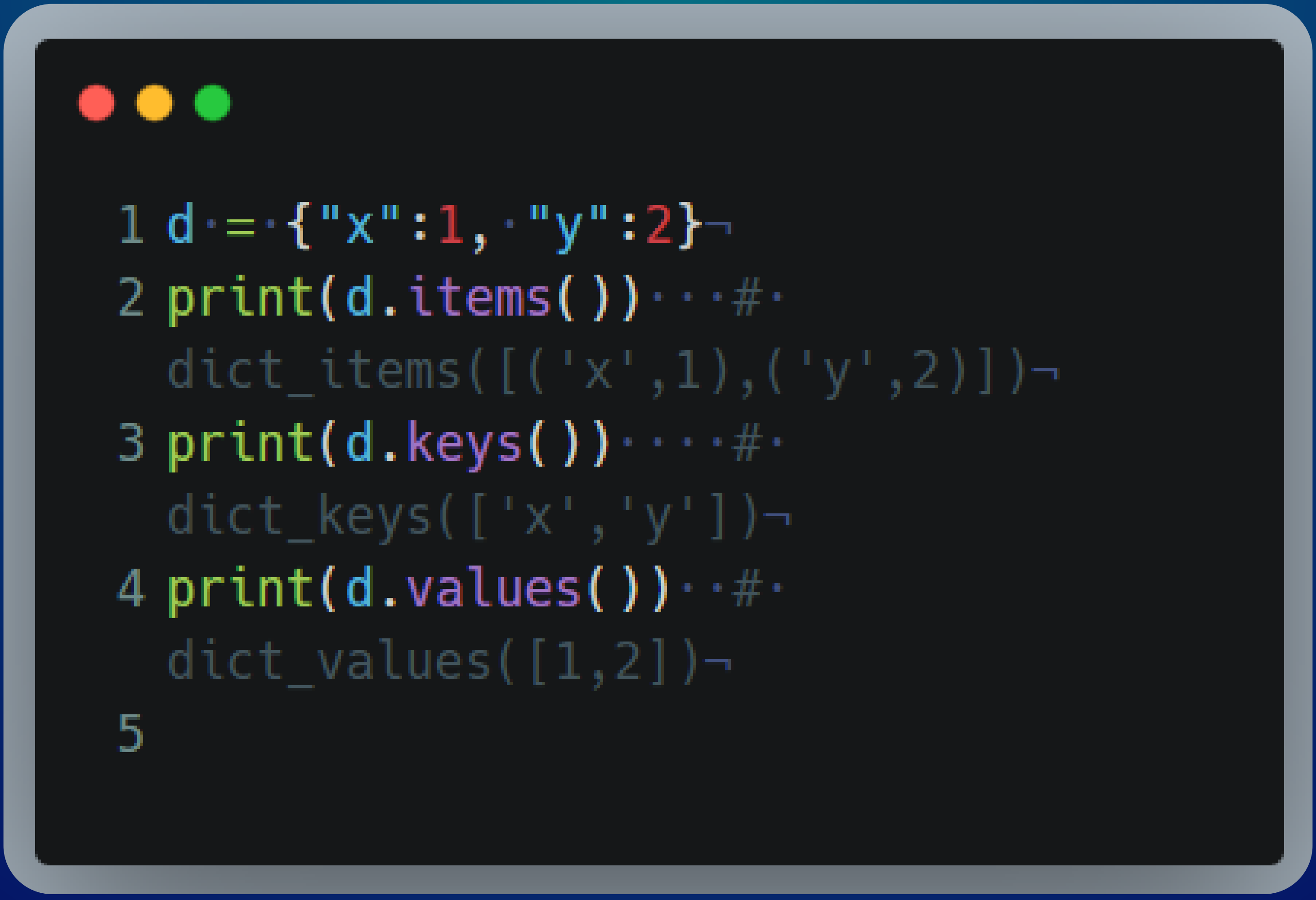
```python
d = {"a": 1, "b": 2}
print(d.copy())  # {'a':1,'b':2}
d.clear()
print(d)         # {}
```

Abdullah Shayed

2) items() + keys() + values()

- items() → Key - value pairs.
- keys() → All keys.
- values() → All values.

Example →

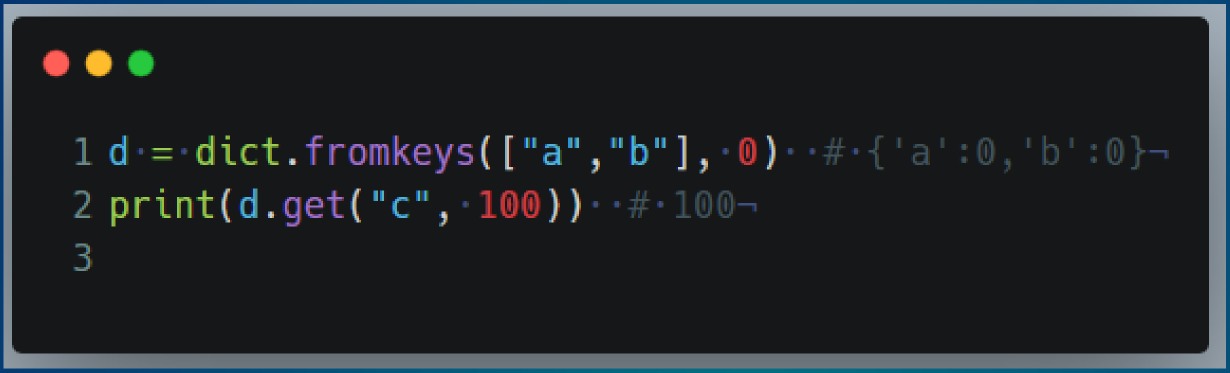


```python
d = {"x":1, "y":2}
print(d.items())   # dict_items([('x',1),('y',2)])
print(d.keys())    # dict_keys(['x','y'])
print(d.values())  # dict_values([1,2])
```

3) fromkeys() + get()

- fromkeys() → Creates dict with default values.
- get() → Returns value or default.

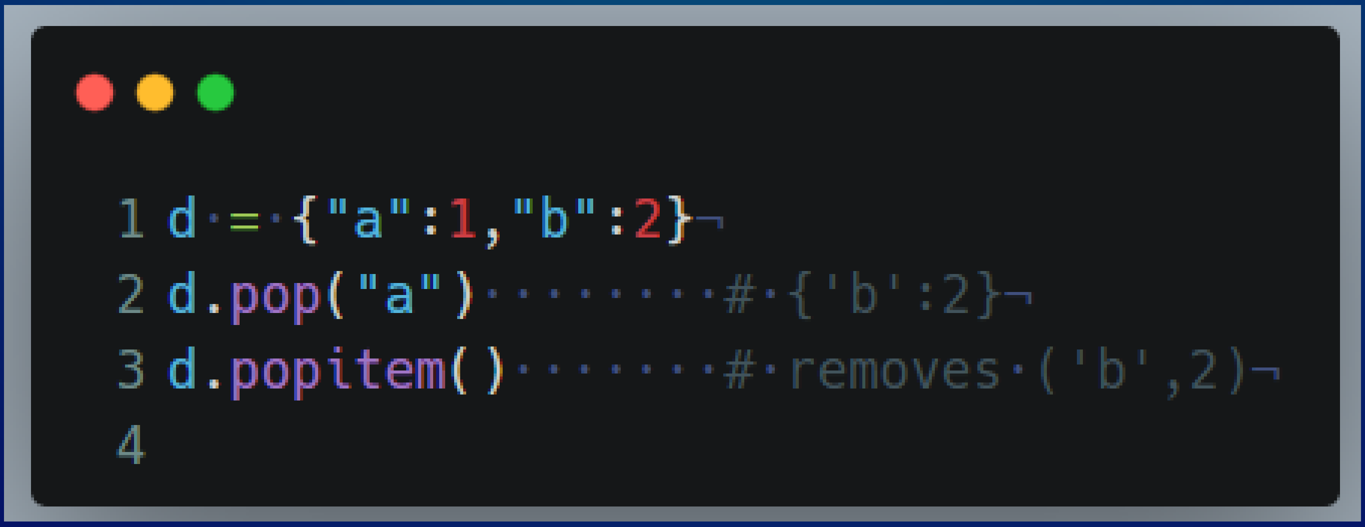
Example →



```python
d = dict.fromkeys(["a","b"], 0)  # {'a':0,'b':0}
print(d.get("c", 100))  # 100

```

4) pop() + popitem()

- pop(k) → Removes given key.
- popitem() → Removes last item.

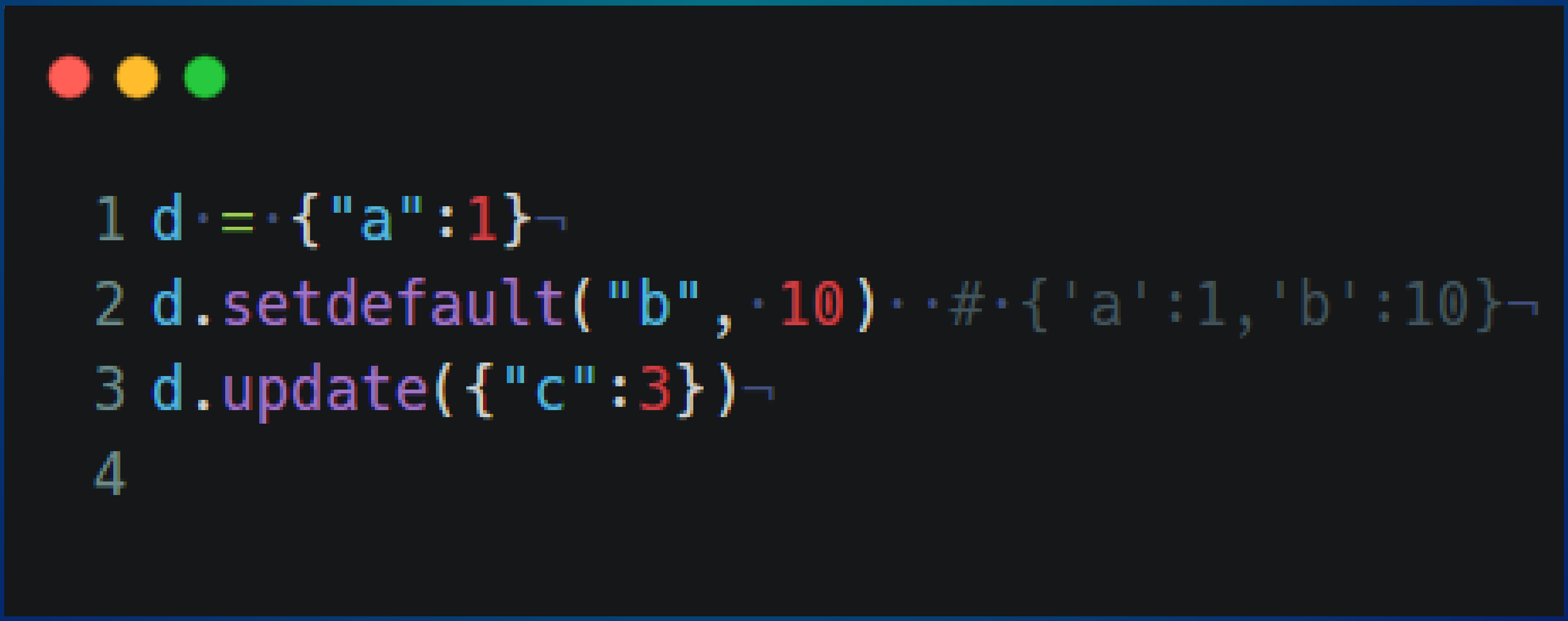


```python
d = {"a":1,"b":2}
d.pop("a")        # {'b':2}
d.popitem()       # removes ('b',2)

```

5) setdefault() + update()

- ◆ setdefault() → Returns value or inserts default.
- ◆ update() → Adds / updates multiple items.

Example →



```python
d = {"a":1}
d.setdefault("b", 10)  # {'a':1,'b':10}
d.update({"c":3})

```

6) Dunder Methods

◆ Dunder methods (short for Double *UNDERSCORE* methods) are special built-in methods in Python that start and end with **__**, like *__len__() or __getitem__()*.

◆ They let you define how objects of your class or data type should behave with operators and built-in functions (e.g., *len(obj)* actually calls obj.*__len__()*).

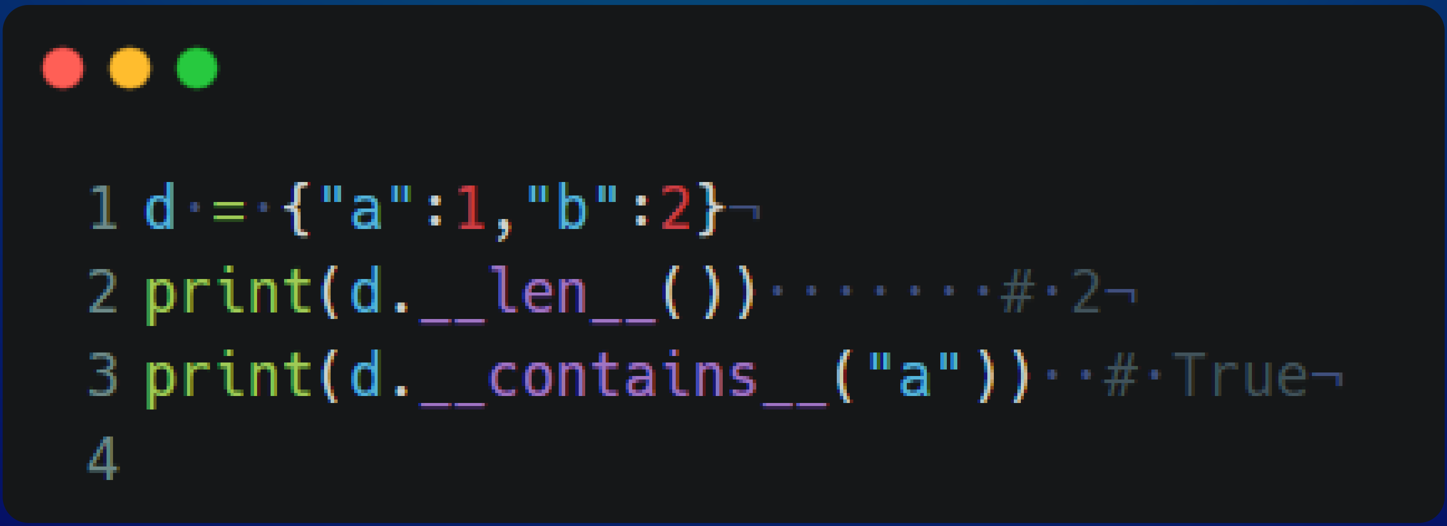
◆ Dunder methods aren't usually called directly — instead, they're triggered behind the scenes when you use Python's standard syntax.

6.1) Dunder Methods Part-One

◆ __len__() → Number of items.

◆ __contains__() → Check if key exists.

Example →

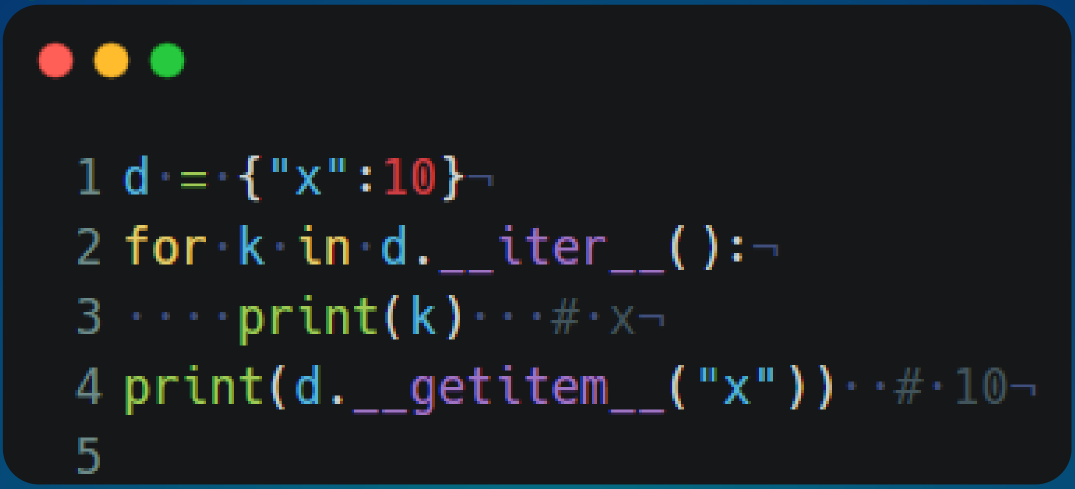


```python
d = {"a":1,"b":2}
print(d.__len__())       # 2
print(d.__contains__("a"))  # True
```

6.2) Dunder Methods Part-Two

- __iter__() → Iterate keys.
- __getitem__() → Get value by key.

Example →



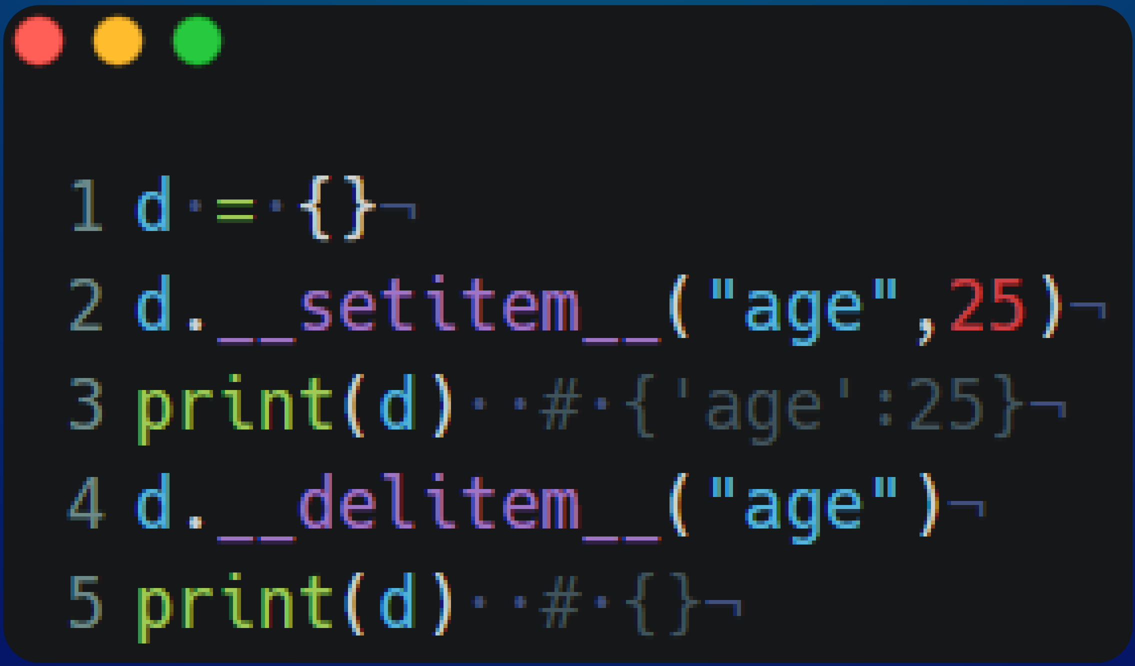
```python
d = {"x":10}
for k in d.__iter__():
    print(k)   # x
print(d.__getitem__("x"))  # 10

```

6.3) Dunder Methods Part-Three

- __setitem__() → Add/modify item.

- __delitem__() → Delete item.

Example →



```python
d = {}
d.__setitem__("age",25)
print(d)  # {'age':25}
d.__delitem__("age")
print(d)  # {}
```

Hey Developer! In this slides

✅ 17 Dictionary methods & attributes covered

✅ From basics (*clear(), get()*) to dunders (*__getitem__()*, **__setitem__()**)

✅ Use this guide as your quick Python reference

💡 Save this post 📕 for quick revision.

👉 Comment your most-used method below!

Thanks for reading !💻

Keep learning, keep coding.

🚀 Follow for me more tips!

#HappyCoding #PythonDictionary #DeveloperNotes