Python - Dictionaries

Shahriar's Intelligence Academy

Dictionaries

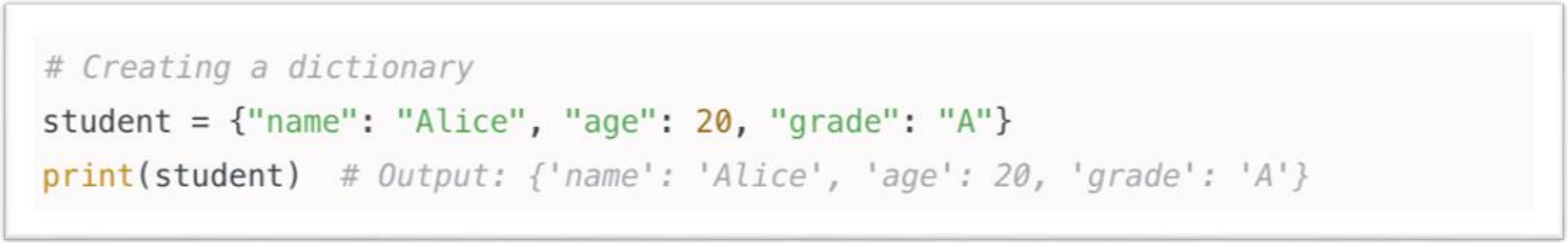
Definition:

A dictionary is an **unordered, mutable, and indexed** collection **of key-value pairs** in Python. It is used to store data in a structured format where each key is unique and associated with a specific value.

Characteristics:

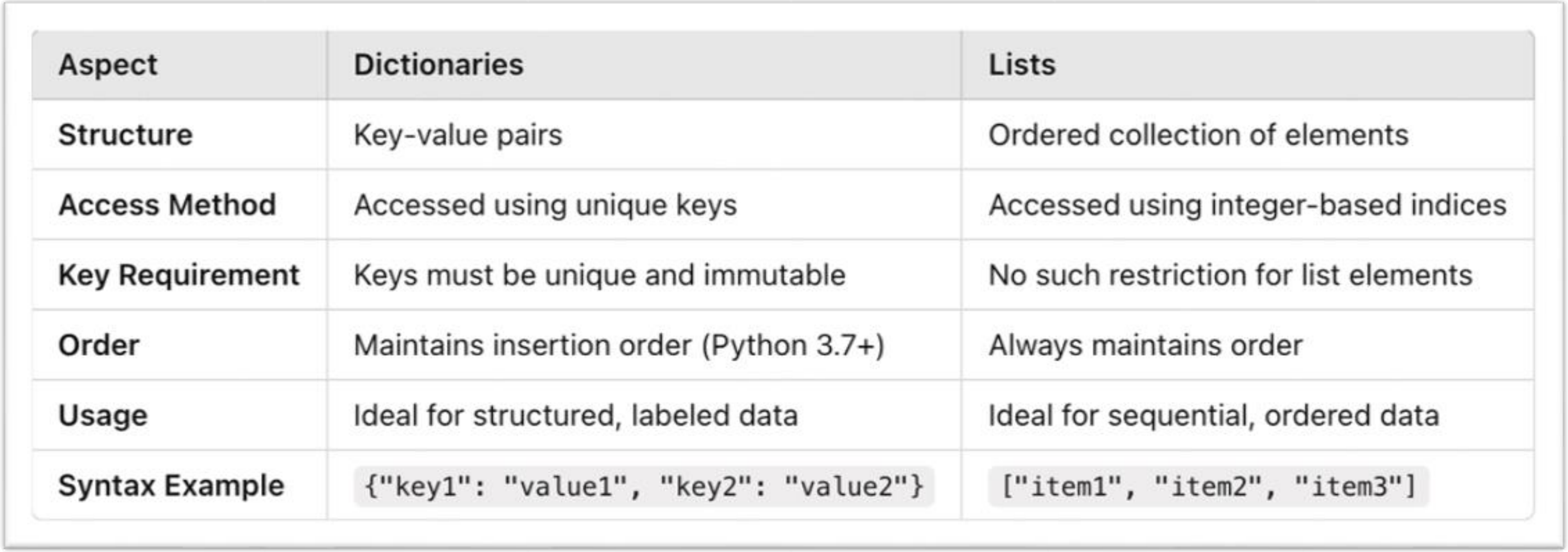
- Key-Value Structure
- Mutable and Unordered
- Unique keys

Example:



```python
# Creating a dictionary
student = {"name": "Alice", "age": 20, "grade": "A"}
print(student)  # Output: {'name': 'Alice', 'age': 20, 'grade': 'A'}
```

Dictionaries v/s Lists



Aspect	Dictionaries	Lists
Structure	Key-value pairs	Ordered collection of elements
Access Method	Accessed using unique keys	Accessed using integer-based indices
Key Requirement	Keys must be unique and immutable	No such restriction for list elements
Order	Maintains insertion order (Python 3.7+)	Always maintains order
Usage	Ideal for structured, labeled data	Ideal for sequential, ordered data
Syntax Example	`{"key1": "value1", "key2": "value2"}`	`["item1", "item2", "item3"]`

Dictionaries Slicing

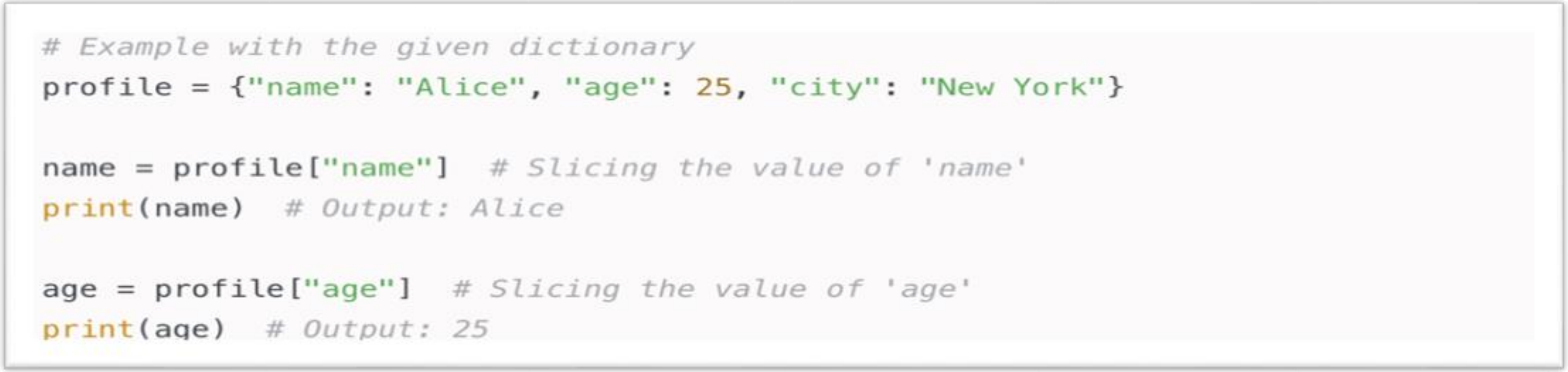
Definition:

Dictionary value slicing refers to **extracting specific values from a dic**tionary using their **keys**.

Syntax:

value = **dictionary[** “key” **]**

Example:



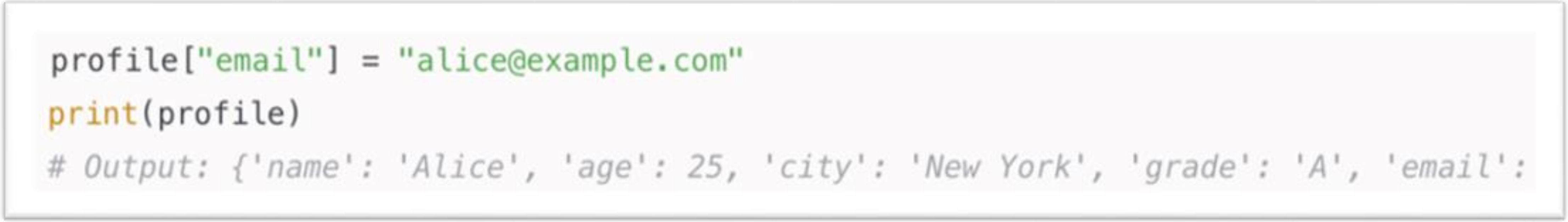
```
# Example with the given dictionary
profile = {"name": "Alice", "age": 25, "city": "New York"}

name = profile["name"]  # Slicing the value of 'name'
print(name)  # Output: Alice

age = profile["age"]  # Slicing the value of 'age'
print(age)  # Output: 25
```

Value Add/Remove

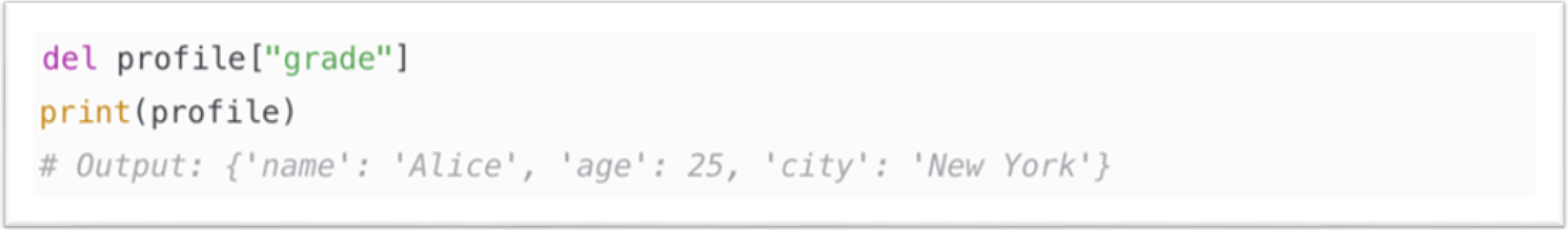
Key Features:

- Add a new key-value pair by **assigning a value** to a new key
- Remove a key-value pair using the **del** statement.

Example:



```
profile["email"] = "alice@example.com"
print(profile)
# Output: {'name': 'Alice', 'age': 25, 'city': 'New York', 'grade': 'A', 'email':
```



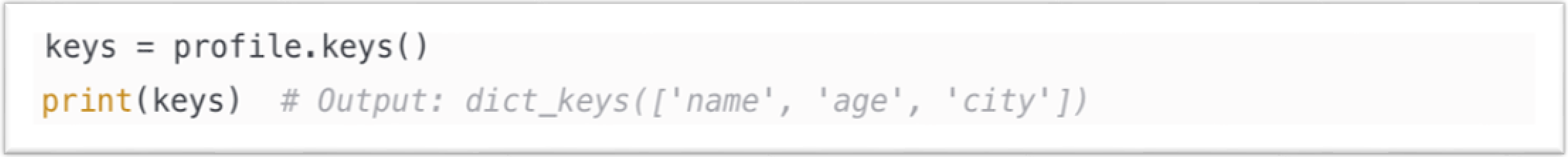
```
del profile["grade"]
print(profile)
# Output: {'name': 'Alice', 'age': 25, 'city': 'New York'}
```

Dictionaries Extractions

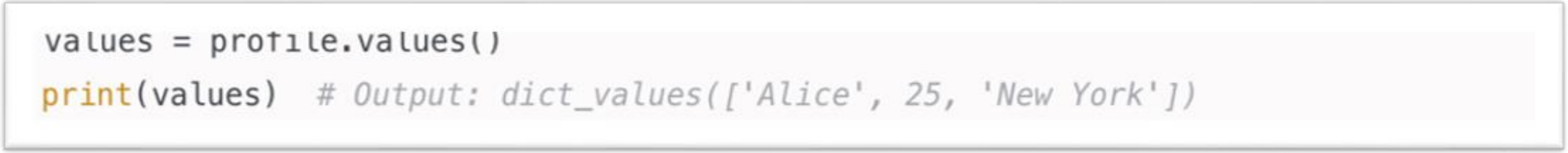
Key Features:

- The **keys()** method retrieves all the keys in a dictionary.
- The **values()** method retrieves all the values in a dictionary.
- The **items()** method retrieves all key-value pairs as tuples.

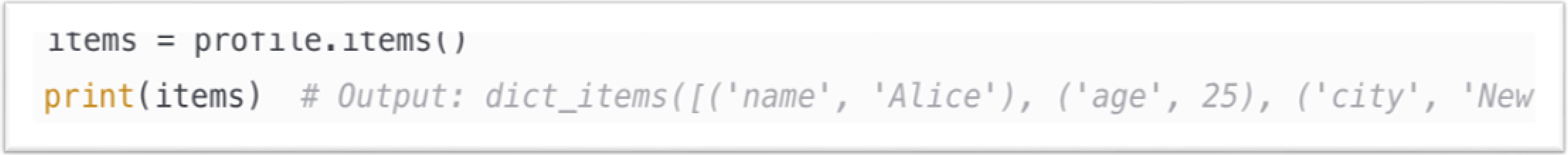
Example:



```python
keys = profile.keys()
print(keys)  # Output: dict_keys(['name', 'age', 'city'])
```



```python
values = profile.values()
print(values)  # Output: dict_values(['Alice', 25, 'New York'])
```



```python
items = profile.items()
print(items)  # Output: dict_items([('name', 'Alice'), ('age', 25), ('city', 'New
```

Thank You!

Shahriar's Intelligence Academy