

Dictionaries in Python

Dictionaries in Python

This note is a compilation of information from two videos on Python dictionaries.

Introduction to Dictionaries

Dictionaries are a data structure that stores data as **key-value pairs**. They are ordered collections as of Python 3.7 and provide fast access to data.

Creating and Accessing Dictionaries

Code Example: Creating and Accessing

...

```
# Creating a dictionary
student = {
    "name": "Alice",
    "age": 20,
    "major": "Computer Science"
}
```

```
# Accessing a value using a key
print(student["name"]) # Output: Alice
```

```
# Using the .get() method
print(student.get("age")) # Output: 20
print(student.get("city")) # Output: None
...
```

Dictionary Methods and Operations

Modifying Dictionaries

- **Adding/Updating:** Use the `update()` method to add new key-value pairs or update existing ones.
- **Removing:** Use `pop()` to remove a specific key-value pair, or `popitem()` to remove the last

one.

Code Example: Updating and Removing

...

```
student.update({"age": 21})
print(student) # Output: {'name': 'Alice', 'age': 21, 'major': 'Computer Science'}
```

```
student.pop("major")
print(student) # Output: {'name': 'Alice', 'age': 21}
...
```

Accessing Items

- `keys()`: Returns a view of the dictionary's keys.
- `values()`: Returns a view of the dictionary's values.
- `items()`: Returns a view of the dictionary's key-value pairs as tuples.