**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Topic 35 - Retrieving Information from a List of Dictionaries in Python**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**What**

In Python, you can access information from dictionaries that are part of a list by using both the **list index** and the **dictionary key**. The list index helps locate the specific dictionary, while the key fetches the exact value within that dictionary.

**Why**

This method is essential for efficiently retrieving information in scenarios where data is stored across multiple dictionaries, such as a customer database. Using the index, you can locate each customer's data without having to create unique names for every dictionary, making your code more scalable and manageable.

**How**

Here’s how to retrieve data from a list of dictionaries. Suppose we have this list:

python

Copy code

customers = [

{

"customer id": 0,

"first name": "John",

"last name": "Ogden",

"address": "301 Arbor Rd.",

},

{

"customer id": 1,

"first name": "Ann",

"last name": "Sattermyer",

"address": "PO Box 1145",

},

{

"customer id": 2,

"first name": "Jill",

"last name": "Somers",

"address": "3 Main St.",

},

]

If you want to get the address of the customer with customer id 1:

1. Locate the dictionary at index 1 in the list, and assign it to a variable.
2. Use the key ["address"] to get the address from the dictionary.

Example code:

python

Copy code

# Step 1: Get the dictionary of the customer with id 1

dictionary\_to\_look\_in = customers[1]

# Step 2: Access the address

customer\_address = dictionary\_to\_look\_in["address"]

print(customer\_address) # Outputs: PO Box 1145

**Important:** Keep the list intact to maintain correct indexing that matches customer id values. If an entry becomes inactive, consider moving it to a separate list instead of deleting it.

**Summary**

By using indexes in a list of dictionaries, you can retrieve data from each dictionary without needing unique names for each. This approach is useful for organized data retrieval, as long as list integrity is preserved.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**