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

**Topic 32 - Looping Through Keys in Python Dictionaries**

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

**What**

Python allows you to loop through the **keys** of a dictionary using the .keys() method, which returns each key in the dictionary in sequence. This approach is helpful when you need to access or display the dictionary’s keys.

**Why**

Looping through dictionary keys is useful when you want to perform actions based on the keys alone or use them to retrieve their associated values in another step. For example, in situations where the keys provide meaningful labels (like “name” or “address”), looping through them simplifies code and improves readability.

**How**

To iterate over the keys in a dictionary, use a for loop with the .keys() method. Here’s an example:

python

Copy code

customer\_29876 = {

"first name": "David",

"last name": "Elliott",

"address": "4803 Wellesley St."

}

# Loop through and print each key

for each\_key in customer\_29876.keys():

print(each\_key)

In this code:

* **Line 4**: The for loop iterates over each key in customer\_29876.
* **Line 5**: print(each\_key) outputs each key, resulting in:

sql

Copy code

first name

last name

address

**Summary**

Looping through dictionary keys with .keys() is an efficient way to access or display the dictionary’s structure without touching values. This makes it particularly useful when working with dictionaries where keys hold critical labels, enhancing code readability and organization.

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