



# Copy Dictionary



# Table of contents

**01** Introduction

**03** Shallow Copy

**02** Direct Assignment

**04** Deep Copy

# Introduction

- **Copying** a dictionary isn't as straightforward as **assigning** one variable to another.
- If you do a **direct assignment** using `=`, you're only **copying** the **reference**, not the **actual data**.
- That means **both variables** will point to the **same dictionary in memory**.
- To create a **true copy**, Python offers **several methods**.

# Direct Assignment (=) – Shallow Reference

! Note: Both original and copy\_ref point to the same object.

<u>Python</u>	<u>Output</u>
<pre>original = {"a": 1, "b": 2} copy_ref = original  copy_ref["a"] = 99 print(original)</pre>	<pre>{'a': 99, 'b': 2}</pre>

# Using .copy() Method – Shallow Copy

- **.copy()** creates a **new dictionary** with the same key-value pairs.
- Changes in the copy **don't affect** the original.
- But it's still a **shallow copy** — it doesn't **recursively copy** nested objects.

<u>Python</u>	<u>Output</u>
<pre>original = {"a": 1, "b": 2} copy_dict = original.copy()  copy_dict["a"] = 99 print(original) print(copy_dict)</pre>	<pre>{'a': 1, 'b': 2} {'a': 99, 'b': 2}</pre>

# Using dict() Constructor - Shallow Copy

- This is functionally **equivalent** to **.copy()**.

<u>Python</u>	<u>Output</u>
<pre>original = {"x": 10, "y": 20} copy_dict = dict(original)  copy_dict["x"] = 50 print(original) print(copy_dict)</pre>	<pre>{'x': 10, 'y': 20} {'x': 50, 'y': 20}</pre>

# Using copy Module – Deep Copy

- If your **dictionary** contains **nested dictionaries**, **shallow copying** won't be enough.
- Use **copy.deepcopy()** to clone the entire structure:

<u>Python</u>	<u>Output</u>
<pre>import copy  original = {"emp": {"name": "Raj", "id": 101}} deep_copy = copy.deepcopy(original)  deep_copy["emp"]["name"] = "Amit" print(original) print(deep_copy)</pre>	<pre>{   'emp': {'name': 'Raj', 'id': 101} }  {   'emp': {'name': 'Amit', 'id': 101} }</pre>



Practice Set – 2

# Download Link in **Description** and **Pinned Comment**



# WATCH

Level up your coding with each episode in this focused Python series.



# Next Video!

**Practice Set - 2  
Solution**

