@pythonessdatadiaries

🚨 Catch This Python Bug!

🔍 **Most developers make this mistake without realizing it!** 🤯
This tricky Python bug confuses even experienced programmers. **Interviewers love asking such questions** to test your understanding of Python's behavior. Can you spot the issue? 👀

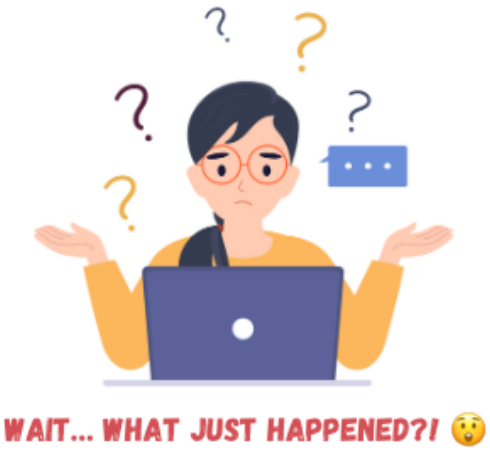
📌 Code Snippet:

```
try:
    mydict = dict()
    mydict = mydict.get('a',10)
    print(mydict['a'])
except Exception as e:
    print("Error:",e)
finally:
    print(mydict)
```

❌ Output:

```
In [17]: try:
             mydict = dict()#empty dict
             mydict = mydict.get('a',10)
             print(mydict['a'])
         except Exception as e:
             print("Error:",e)
         finally:
             print(mydict)

Error: 'int' object is not subscriptable
10
```



🔎 Explanation:

1️⃣ mydict = dict() → Creates an empty dictionary ✅
2️⃣ mydict = mydict.get('a', 10) → **Tricky part!** get('a', 10) returns **10** (default value) since 'a' key doesn't exist.
But now mydict is no longer a dictionary—it's an integer (10)! 😵
3️⃣ print(mydict['a']) → ❌ Throws an error because 10 (an integer) **isn't subscriptable!**

💡 Fix:

Instead of overwriting the dictionary, update the line # 2 to update value inside the key 'a' of dictionary, mydict.

```python
try:
    mydict = dict()#empty dict
    mydict['a'] = mydict.get('a',10)
    print(mydict['a'])
except Exception as e:
    print("Error:",e)
finally:
    print(mydict)
```

Run the code now.▶

✅Output :

```
In [18]: try:
             mydict = dict()#empty dict
             mydict['a'] = mydict.get('a',10)
             print(mydict['a'])
         except Exception as e:
             print("Error:",e)
         finally:
             print(mydict)

         10
         {'a': 10}
```

🚀 Lesson:

Be careful when using .get()—don't overwrite your dictionary with a default value!

💬 Have you faced this issue before?
🔔 Follow for more Python tricks & interview tips! ✨