

```
class DictionaryManager:

    def __init__(self):
        self.data = {}

    def add_entry(self, key, value):
        if key in self.data:
            print(f"Key '{key}' already exists. Use update_entry to modify it.")
        else:
            self.data[key] = value
            print(f"Entry added: {key} -> {value}")

    def update_entry(self, key, value):
        if key in self.data:
            old_value = self.data[key]
            self.data[key] = value
            print(f"Entry updated: {key} -> {old_value} changed to {value}")
        else:
            print(f"Key '{key}' does not exist. Use add_entry to add it.")

    def delete_entry(self, key):
        if key in self.data:
            del self.data[key]
            print(f"Entry with key '{key}' deleted.")
        else:
            print(f"Key '{key}' not found.")

    def search_entry(self, key):
        if key in self.data:
            print(f"Found: {key} -> {self.data[key]}")
        else:
            print(f"Key '{key}' not found in the dictionary.")
```

```
def display_all_entries(self):  
    if self.data:  
        print("All entries in the dictionary:")  
        for key, value in self.data.items():  
            print(f"{key} -> {value}")  
    else:  
        print("The dictionary is empty.")
```

```
def key_exists(self, key):  
    return key in self.data
```

```
def get_value(self, key):  
    return self.data.get(key, None)
```

```
# Creating an object of DictionaryManager  
my_dict_manager = DictionaryManager()
```

```
# Interactive menu for user input
```

```
while True:  
    print("\nChoose an operation:")  
    print("1. Add entry")  
    print("2. Update entry")  
    print("3. Delete entry")  
    print("4. Search entry")  
    print("5. Display all entries")  
    print("6. Check if a key exists")  
    print("7. Get value for a key")  
    print("8. Exit")
```

```
choice = input("Enter your choice (1-8): ")
```

```
if choice == '1':
```

```
    key = input("Enter the key: ")
```

```
    value = input("Enter the value: ")
```

```
    my_dict_manager.add_entry(key, value)
```

```
elif choice == '2':
```

```
    key = input("Enter the key to update: ")
```

```
    value = input("Enter the new value: ")
```

```
    my_dict_manager.update_entry(key, value)
```

```
elif choice == '3':
```

```
    key = input("Enter the key to delete: ")
```

```
    my_dict_manager.delete_entry(key)
```

```
elif choice == '4':
```

```
    key = input("Enter the key to search: ")
```

```
    my_dict_manager.search_entry(key)
```

```
elif choice == '5':
```

```
    my_dict_manager.display_all_entries()
```

```
elif choice == '6':
```

```
    key = input("Enter the key to check: ")
```

```
    if my_dict_manager.key_exists(key):
```

```
        print(f"The key '{key}' exists in the dictionary.")
```

```
    else:
```

```
        print(f"The key '{key}' does not exist.")
```

```
elif choice == '7':
```

```
key = input("Enter the key to get the value: ")
value = my_dict_manager.get_value(key)
if value is not None:
    print(f"The value for key '{key}' is: {value}")
else:
    print(f"The key '{key}' does not exist.")
```

```
elif choice == '8':
    print("Exiting the program.")
    break
```

```
else:
    print("Invalid choice. Please try again.")
```

Its output

Choose an operation:

1. Add entry
2. Update entry
3. Delete entry
4. Search entry
5. Display all entries
6. Check if a key exists
7. Get value for a key
8. Exit

Enter your choice (1-8): 1

Enter the key: Name

Enter the value: Alice

Entry added: Name -> Alice

Choose an operation:

1. Add entry
2. Update entry

3. Delete entry
4. Search entry
5. Display all entries
6. Check if a key exists
7. Get value for a key
8. Exit

Enter your choice (1-8): 5

All entries in the dictionary:

Name -> Alice

Choose an operation:

1. Add entry
2. Update entry
3. Delete entry
4. Search entry
5. Display all entries
6. Check if a key exists
7. Get value for a key
8. Exit

Enter your choice (1-8): 2

Enter the key to update: Name

Enter the new value: abcd

Entry updated: Name -> Alice changed to abcd

Choose an operation:

1. Add entry
2. Update entry
3. Delete entry
4. Search entry
5. Display all entries
6. Check if a key exists

7. Get value for a key

8. Exit

Enter your choice (1-8): 5

All entries in the dictionary:

Name -> abcd

hoose an operation:

1. Add entry

2. Update entry

3. Delete entry

4. Search entry

5. Display all entries

6. Check if a key exists

7. Get value for a key

8. Exit

Enter your choice (1-8):

Its explanation

What is DictionaryManager?

DictionaryManager is a Python class that helps you manage a dictionary of key-value pairs. You can add, update, delete, search, and display entries in the dictionary using this class.

How does it work?

1. Initialization:

- When you create an object of DictionaryManager, it initializes an empty dictionary to store the data.

2. Adding Entries:

- Use the add_entry method to add a new key-value pair to the dictionary. If the key already exists, it will notify you to use the update method instead.

3. Updating Entries:

- Use the update_entry method to change the value of an existing key. If the key does not exist, it will notify you to use the add method instead.

4. Deleting Entries:

- Use the `delete_entry` method to remove a key-value pair from the dictionary. If the key does not exist, it will notify you.
- 5. **Searching Entries:**
 - Use the `search_entry` method to find and display the value associated with a key. If the key does not exist, it will notify you.
- 6. **Displaying All Entries:**
 - Use the `display_all_entries` method to print all key-value pairs in the dictionary. If the dictionary is empty, it will notify you.
- 7. **Checking if a Key Exists:**
 - Use the `key_exists` method to check if a key is present in the dictionary. It returns `True` if the key exists, otherwise `False`.
- 8. **Getting the Value for a Key:**
 - Use the `get_value` method to get the value associated with a key. If the key does not exist, it returns `None`.

Interactive Menu

The interactive menu allows you to choose different operations to perform on the dictionary by entering a number (1-8). It keeps running in a loop until you choose to exit.

Example Usage

When you run the script, it will continuously prompt you to choose an operation. You can add, update, delete, search, display, check, or get values from the dictionary based on your input.

Overall, `DictionaryManager` makes it easy to handle a dictionary of key-value pairs with a user-friendly menu interface.