**1. display\_help() Function**

This function provides a guide to the available commands in the program. When called, it displays:

* NF admin = Create a new file and input its contents.
* TC admin = Perform a "terminal check" to simulate checking for bugs or threats.
* h admin = Show the help menu (this function itself).
* RF admin = Run (view) the contents of an existing file.
* exit = Exit the program.

**2. File Storage**

The files dictionary is used to store files. Each file has:

* **Key**: File name (entered by the user).
* **Value**: File contents (also provided by the user).

**3. The while Loop**

The program runs indefinitely until the user types the exit command. It continuously prompts the user with kathi\_admin: to enter a command.

**4. Commands**

* exit Ends the program, displaying: "Exiting KATHI Admin. Goodbye!".
* h admin Calls the display\_help() function to display the help menu.
* NF admin Lets the user create a new file. The program asks for:
  1. The name of the file.
  2. The contents of the file. Then, it stores this information in the files dictionary.
* RF admin Lets the user view the contents of an existing file.
  1. If the file exists (in the files dictionary), it displays the file name and its content.
  2. If the file doesn't exist, it shows: "File not found."
* TC admin Simulates a "terminal check," showing messages like:
  1. Checking for bugs.
  2. Scanning for threats. This is more of a placeholder to demonstrate an action.
* **Invalid Commands** If the user enters a command that's not recognized, the program responds with: "Invalid command. Type 'h admin' for help."

**Purpose**

This script can be thought of as a basic mock terminal program where a user can create and manage simple "files," as well as simulate diagnostic checks and get help. It's a great example of basic programming concepts like:

* Functions.
* Dictionaries.
* Loops.
* Conditional statements