Python for Beginners: Portfolio Project

[Nwichi Fredrick Chiemezie]

Project Title:

(Feline Fact Finder)

Project Code:

(https://replit.com/@fredlumberg/program-cat2#main.py)

Description:

(This Python code uses the Cat API to retrieve information about a specific cat breed based on user input. Overall it provides a simple and interactive way for cat lovers to explore and learn about different cat breeds using data sourced from the Cat API)

How to Use:

1. Visit the Replit link provided above.
2. Click on the "Run" button.
3. Once prompted, enter the cat breed you want information on.
4. The program will display the name, description, temperament, origin and life span of the cat breed.
5. To exit the program, simply close the Replit window.

Technical Details:

- The get_cat_info function accepts a cat breed as input and constructs a URL to query the Cat API for relevant data.
- Upon receiving a response from the API, the code parses the JSON data and extracts details such as the breed's name, description, temperament, origin, and life span.
- If no information is found for the specified breed, an appropriate message is displayed.

- The code handles potential errors gracefully, including connection errors, timeouts, and other request exceptions, by providing informative error messages to the user.

- When executed as the main program, the script prompts the user to input the name of a cat breed and calls the get_cat_info function to fetch and display information about the requested breed.

Future Plans:

- Caching Mechanism: Implement a caching mechanism to store previously retrieved cat breed information locally. This reduces the number of API requests and improves performance, especially for frequently accessed breeds.

- Unit Testing: Develop unit tests to verify the correctness and reliability of individual functions within the codebase. Testing ensures that each component behaves as expected and facilitates code maintenance and refactoring.

- User Interface Enhancement: Create a graphical user interface (GUI) or command-line interface (CLI) to provide a more user-friendly and interactive experience. This allows users to interact with the application more intuitively and efficiently.

- Expand Data Retrieval: Consider retrieving additional information about cat breeds, such as images, average weight, popularity rankings, and health considerations. This enriches the user experience and provides more comprehensive insights into different cat breeds.

- Integration with External Services: Integrate with other external services or APIs to enhance the functionality and data sources available to users. For example, integrate with image recognition APIs to identify cat breeds from uploaded images.

Python Code:

```
# Import the requests module for making HTTP requests
import requests


# Define a function to fetch and display information about a cat breed
def get_cat_info(cat_type):
    # Construct the URL for querying the Cat API with the provided cat breed
    url = f"https://api.thecatapi.com/v1/breeds/search?q={cat_type}"

    try:
        # Send a GET request to the Cat API and store the response
        response = requests.get(url)

        # Parse the JSON response into a Python dictionary
        data = response.json()

        # Check if data is empty (no information found for the given cat breed)
        if not data:
            print("No information found for that cat breed.")
            return

        # Extract relevant information about the cat breed from the response
        information = data[0]
        name = information.get("name", "N/A")
        description = information.get("description", "N/A")
        temperament = information.get("temperament", "N/A")
        origin = information.get("origin", "N/A")
        life_span = information.get("life_span", "N/A")

        # Print out the retrieved information about the cat breed
        print(f"Name: {name}")
        print(f"Description: {description}")
        print(f"Temperament: {temperament}")
        print(f"Origin: {origin}")
        print(f"Life Span: {life_span}")
        print("---------------")

    # Handle connection errors (e.g., internet connection issue)
    except requests.ConnectionError:
        print("Error: Failed to connect to the API. Please check your internet connection.")

    # Handle request timeout errors
    except requests.Timeout:
        print("Error: The request to the API timed out. Please try again later.")

    # Handle other request-related exceptions
    except requests.RequestException as e:
        print(f"An error occurred: {e}")

# Main entry point of the program
if __name__ == "__main__":
    # Prompt the user to enter the name of a cat breed
    cat_type = input("Enter the name of a cat breed: ")

    # Call the get_cat_info function to retrieve information about the entered cat breed
    get_cat_info(cat_type)
```