

# Number Processing Terminal: Code Documentation

## Overview

This Python program is an interactive Number Processing Terminal. It allows users to input numbers, analyze them, and view a session history. The program emphasizes user-friendly design with features like delayed outputs, clear headers, and a clean console interface.

## Features

1. Number Analysis:
  - Input five numbers.
  - Analyze totals, even/odd counts, and sums.
2. Session History:
  - Stores analysis results from each session.
  - Displays historical data in a structured manner.
3. User Experience Enhancements:
  - Console clearing and headers for clean outputs.
  - Digital loading effects for better interaction.
  - Delayed prints for smoother transitions.

## Key Functions

1. `clear_screen()`
  - Clears the console screen for better readability.
  - Usage: Supports cross-platform environments (Windows and Unix).
2. `digital_loader(message="Loading")`
  - Simulates a loading effect by cycling through dots.
  - Purpose: Enhances user experience by mimicking real-time processing.
3. `print_header(title)`
  - Displays a formatted header with a title.
  - Purpose: Visually organize sections of the terminal.
4. `delayed_print(label, value)`
  - Prints results with a delay for a smoother display.

**Parameters:**

label (str): Description of the result.

value (str/int/float): The result value.

**5. process\_numbers()**

Main processing function for analyzing user-input numbers.

**Steps:**

1. Prompts the user to input 5 numbers.
2. Analyzes the input:
  - Total sum of numbers.
  - Count and sum of even/odd numbers.
3. Displays results using `delayed_print`.
4. Saves results to a global history list.

**Key Variables:**

`total_sum`, `even_count`, `odd_count`, `even_sum`, `odd_sum`.

**6. display\_history()**

Displays the analysis history for all sessions.

**Functionality:**

Iterates over history to print results of past sessions.

Formats data using headers and delayed outputs.

**7. Main Menu Loop**

Controls the flow of the program.

**Options:**

1. Analyze Numbers (`process_numbers`).
2. View History (`display_history`).
3. Exit the Program.

## Program Flow

1. Startup:  
Displays the main menu.
2. User Interaction:  
Prompts user for an action: number analysis, history viewing, or exit.
3. Session Analysis: Accepts number inputs.  
Processes and displays results
4. History Management:  
Maintains a global list of session results for review.

## User Experience Highlights

### Interactive Design:

Headers and loaders for a professional feel.  
Error handling for invalid inputs.

## History Accessibility:

Easily review previous session data.

## Code Snippet: Example Analysis Output

When analyzing numbers, the terminal produces the following:

```
📄 Results Summary:
➡ Numbers Entered:    12, 15, 7, 8, 20
➡ Total Sum:          62
➡ Even Count:         3
➡ Sum of Evens:       40
➡ Odd Count:          2
➡ Sum of Odds:        22
```

---

## Improvements

Add support for floating-point numbers.  
Provide data visualization (e.g., bar charts for even/odd counts).  
Include an option to save/export session history to a file.