Car Data Scraping and Processing Script

# Overview

This Python script scrapes car data from the website [Montate Diuna Nave](https://www.montatediunave.com/), extracts key details such as car name, year, and price, processes the data, and saves the results in compressed format. The script also generates two JSON files that map car models to unique integers and store the minimum car year.

# Features

- Scrapes car data (name, year, price) from the website.  
- Converts car names to unique integers using a mapping.  
- Adjusts the car year relative to the minimum year in the dataset.  
- Rounds car prices to thousands and saves them in a compressed CSV format.  
- Outputs two JSON files:  
 - `model\_mapping.json`: Maps car models to unique integers.  
 - `min\_year.json`: Stores the minimum year in the dataset.

# Requirements

Python 3.x  
- `requests`  
- `beautifulsoup4`  
- `pandas`  
  
You can install the required libraries using:  
```bash  
pip install requests beautifulsoup4 pandas  
```

# Script Breakdown

## 1. Scraping Data

The `scrape\_table()` function extracts car information from the website:  
- \*\*Name\*\*: The car's model name.  
- \*\*Year\*\*: The car's year of manufacture.  
- \*\*Price\*\*: The car's price, with currency symbols and commas removed.  
  
The script uses the `requests` library to fetch web pages and `BeautifulSoup` to parse the HTML content.

## 2. Iterating Over Multiple Pages

The script loops through multiple pages of the car listing website by appending a page number to the URL and calls the `scrape\_table()` function on each page.

## 3. Data Processing

- \*\*Model to Integer Mapping\*\*: A dictionary is created to map car models (names) to unique integers. This mapping is saved to `model\_mapping.json`.  
- \*\*Year Adjustment\*\*: The car's year is adjusted relative to the minimum year found in the dataset. The minimum year is saved to `min\_year.json`.  
- \*\*Price Rounding\*\*: Car prices are divided by 1,000 and rounded to integers.

## 4. Saving Data

The processed car data is saved as a compressed CSV file: `car\_data\_compressed.csv`.

# Running the Script

1. Clone or download the script:  
```bash  
git clone https://github.com/your-repo/car-data-scraper.git  
cd car-data-scraper  
```  
2. Run the script:  
```bash  
python scrape\_car\_data.py  
```  
The script will scrape car data from the website, process it, and save the following files:  
- `model\_mapping.json`: A mapping of car models to integers.  
- `min\_year.json`: The minimum year in the dataset.  
- `car\_data\_compressed.csv`: The compressed car data.

# Example Output

## JSON File Example (`model\_mapping.json`)

{  
 "Toyota Corolla": 0,  
 "Honda Civic": 1,  
 "Nissan Sentra": 2  
}

## CSV File Example (`car\_data\_compressed.csv`)

Name,Year,Price  
0,3,23  
1,2,21  
2,5,30

# Customization

You can modify the page range in the for loop based on the number of pages on the website:  
```python  
for i in range(1, 100): # Adjust the page range as necessary  
```

# Contributing

Feel free to fork the repository and submit pull requests for new features or bug fixes.

# License

This project is licensed under the MIT License.