

Reading from CSV File

Introduction:

CSV (*Comma-Separated Values*) files are widely used for storing tabular data in a plain text format. Each line in a CSV file represents a row of data, and the values within each row are separated by commas.

Reading from CSV file:

We have the following data of patients in a hospital:

Patient Name	Temperature (Celsius)	Blood Pressure (mmHg)	Heart Rate (bpm)
Ali Abbas	37.2	120/80	72
Sarah Jawad	36.8	118/75	68
Ola Mohammed	37.5	122/78	75
Mohammed Saif	36.9	115/70	70
Fatima Ahmed	37.1	119/76	74
Ahmad Saleh	36.7	121/79	70
Nour Hassan	37.4	118/72	68
Youssef Kamal	36.6	123/81	76
Layla Ibrahim	37	120/77	72
Zainab Ali	36.5	124/82	78
Mona Khalid	37.3	117/74	71
Samiya Mustafa	36.8	120/78	69
Hassan Karim	37.2	122/79	73
Aisha Omar	36.9	118/75	70
Khaled Hadi	37.5	125/80	77
Rawan Mahmoud	36.7	119/76	68
Tariq Ahmed	37.4	120/77	75
Noor Ali	36.6	123/80	71
Salma Nasser	37.1	117/73	72

This code snippet demonstrates how to read data from a the CSV file using the csv module and printing each patient information in the console:

```
import csv
```

```
file_path = 'data.csv'
```

```
with open(file_path, 'r') as file:
```

```
    reader = csv.reader(file) # Create a CSV reader object
```

```
    next(reader)# Skip the header row
```

```
    # Iterate over each row in the CSV file
```

```
    for row in reader:
```

```
        name = row[0]
```

```
        temperature = float(row[1])
```

```
        blood_pressure = row[2]
```

```
        heart_rate = int(row[3])
```

```
        print(f"Patient Name: {name}")
```

```
        print(f"Temperature: {temperature} °C")
```

```
        print(f"Blood Pressure: {blood_pressure}")
```

```
        print(f"Heart Rate: {heart_rate} bpm")
```

- **Explanation:**

1. `import csv`: This line imports the CSV module, which provides functionality for reading and writing CSV files.
2. `file_path = 'data.csv'`: This line specifies the path to the CSV file that contains the patient data.
3. `with open(file_path, 'r') as file::` This line opens the CSV file in read mode ('r') using a context manager. It ensures that the file is properly closed after it's done being used. The file is opened using the variable `file`.
4. `reader = csv.reader(file)`: This line creates a CSV reader object called `reader` using the `csv.reader()` function. The CSV reader is used to iterate over the rows in the CSV file.
5. `next(reader)`: This line skips the header row of the CSV file using the `next()` function. The header row typically contains the column names and is not part of the data to be processed.
6. `for row in reader::` This line starts a loop that iterates over each row in the CSV file.
 - `name = row[0]`: This line extracts the patient's name from the first column (index 0) of the current row.
 - `temperature = float(row[1])`: This line extracts the temperature from the second column (index 1) of the current row and converts it to a floating-point number using the `float()` function.
 - `blood_pressure = row[2]`: This line extracts the blood pressure from the third column (index 2) of the current row.
 - `heart_rate = int(row[3])`: This line extracts the heart rate from the fourth column (index 3) of the current row and converts it to an integer using the `int()` function.
 - `print(f"Patient Name: {name}")`: This line prints the patient's name.
 - `print(f"Temperature: {temperature} °C")`: This line prints the patient's temperature.
 - `print(f"Blood Pressure: {blood_pressure}")`: This line prints the patient's blood pressure.
 - `print(f"Heart Rate: {heart_rate} bpm")`: This line prints the patient's heart rate.