Personal Expense Tracker - AGS-Python Basics

In [1]:

```
from datetime import datetime
import pandas as pd
```

In [2]:

```
expenses = []
```

In [3]:

```
def enter_expenses():
    expense_date = input("Enter Date (YYYY-MM-DD):")
    expense_category = input("Expense Category:")
    expense_amount = float(input("Amount:"))
    expense_description = input("Brief Description:")

    expense = {
        "date": expense_date,
        "category": expense_category,
        "amount":expense_amount,
        "description": expense_description
    }
    expenses.append(expense)
    return expenses
```

In [ ]:

```
personal_expenses = enter_expenses()
print(personal_expenses)
```

In [5]:

```
print(personal_expenses)
```

```
[{'date': '2020-12-12', 'category': 'Food', 'amount': 78.0, 'description': 'Fo
od for thought'}]
```

In [26]:

```
def print_expenses(personal_expenses):
    for expense in personal_expenses:


        if expense["date"] == "" or expense["category"] == "" or expense["amou
            print("Incomplete expense entry skipped.\n")
            continue

        for k,v in expense.items():
            print(k,v)
        print("\n")
```

In [27]:

```
print_expenses(expenses)
```

```
date 2020-12-12
category Food
amount 78.0
description Food for thought


date 2020-12-06
category Clothing
amount 56.0
description Clothing for the poor


date 2020-08-08
category Food
amount 12.0
description Friday night
```

In [7]:

```
def monthly_budget():
    return float(input("What's your monthly budget?"))
```

In [8]:

```
monthly_budget = monthly_budget()
```

In [9]:

```
def calculate_total_expenses(personal_expenses,budget):
    total_expense = 0
    for expense in personal_expenses:


        #if not (expense["date"] == "" or expense["category"] == "" or expense
        total_expense = expense["amount"]+total_expense


        print("\n")
    if total_expense > budget:
        print("You have exceeded your budget",end="\n")
    else:
        print(f'You have {budget-total_expense} left for the month')
```

```
You have 122.0 left for the month
```

In [11]:

```
# Function to save expenses to CSV
def save_expenses_to_csv():
    if not expenses:
        print("No Expense Data to save")
        return

    df = pd.DataFrame(expenses)
    df.to_csv("expenses.csv", index=False)
    print("Expenses saved to expenses.csv")
```

In [13]:

```
save_expenses_to_csv()
```

```
Expenses saved to expenses.csv
```

In [29]:

```
 def menu():
    while True:
        print("1. Add an expense")
        print("2. View expenses")
        print("3. Track budget")
        print("4. Save expenses")
        print("5. Exit")

        choice = input("Enter your choice: ")
        if choice == "1":
            enter_expenses()
        elif choice == "2":
            print_expenses(expenses)
        elif choice == "3":
            calculate_total_expenses(expenses,monthly_budget)
        elif choice == "4":
            save_expenses_to_csv()
            print("Exiting the program")
        elif choice == "5":
            break
        else:
            print("Invalid choice")
```

In [ ]:

```
menu()
```

```
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

Invalid choice
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit
```

```
date 2020-12-12
category Food
amount 78.0
description Food for thought


date 2020-12-06
category Clothing
amount 56.0
description Clothing for the poor


date 2020-08-08
category Food
amount 12.0
description Friday night


date 2027-07-29
category Uber
amount 11.0
description Uber home


1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit







You have 43.0 left for the month
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

Expenses saved to expenses.csv
Exiting the program
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit
```

In [ ]:

```
menu()
```

```
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit
```

```
Invalid choice
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

date 2020-12-12
category Food
amount 78.0
description Food for thought


date 2020-12-06
category Clothing
amount 56.0
description Clothing for the poor


date 2020-08-08
category Food
amount 12.0
description Friday night


date 2027-07-29
category Uber
amount 11.0
description Uber home


1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit







You have 43.0 left for the month
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit
```

```
Expenses saved to expenses.csv
Exiting the program
1. Add an expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit
```