

# Advanced Worksheet: Monthly Expenses with openpyxl Functions

## Sheet Content:

	A	B	C	D	E
1	Item	Date	Category	Amount	Discounted Price
2	Rent	01/01/2024	Housing	1200	
3	Groceries	01/05/2024	Food	150	
4	Electricity	01/08/2024	Utilities	80	
5	Water Bill	01/10/2024	Utilities	30	
6	Internet	01/12/2024	Utilities	60	
7	Dining Out	01/15/2024	Entertainment	45	
8	Gas	01/17/2024	Transportation	70	
9	Movie Tickets	01/20/2024	Entertainment	25	
10	Gym Membership	01/22/2024	Health	50	
11	Car Insurance	01/25/2024	Insurance	100	
12	Pharmacy	01/27/2024	Health	30	
13	Coffee Shop	01/28/2024	Entertainment	10	
14	Groceries	01/30/2024	Food	120	

## Instructions:

### Task 1: Write Functions to Process Data

- Function 1:** `def calculate_discounted_price(sheet, discount=0.10)`
  - Description:** This function should apply a discount to each item in the "Amount" column and write the discounted value in the "Discounted Price" column.
  - Parameters:**
    - `sheet`: The active sheet of the workbook.
    - `discount`: The discount percentage to apply (default is 10%).
  - Implementation:** Loop through each row, calculate the discounted price (`Amount * (1 - discount)`) and store it in the "Discounted Price" column.
- Function 2:** `def calculate_category_totals(sheet)`
  - Description:** This function should calculate the total amount spent for each category (e.g., "Food", "Utilities") and print the results.
  - Parameters:** `sheet`
  - Implementation:** Use a dictionary to sum values for each category, then print or save these totals.
- Function 3:** `def add_grand_total(sheet)`
  - Description:** This function should calculate the grand total of all expenses in the "Amount" column and write it at the bottom of the column.
  - Parameters:** `sheet`
  - Implementation:** Sum all values in the "Amount" column and add the result below the last entry.

4. **Function 4:** `def highlight_expenses_above_threshold(sheet, threshold)`
- **Description:** This function should highlight rows where the "Amount" exceeds a specified threshold.
  - **Parameters:**
    - `sheet`: The active sheet.
    - `threshold`: The amount above which expenses should be highlighted.
  - **Implementation:** For each row, check if the "Amount" is greater than `threshold`. If true, apply a fill color (e.g., yellow) to the row.
- 

## Practice Exercise: Putting it All Together

- **Step 1:** Load the `Monthly_Expenses.xlsx` sheet using `openpyxl`.
- **Step 2:** Run each function with appropriate parameters, ensuring each task is completed within the sheet.
- **Step 3:** Save the modified workbook with a new name (`Monthly_Expenses_Processed.xlsx`).

## Example Code Template

```
from openpyxl import load_workbook
from openpyxl.styles import PatternFill

# Load workbook and active sheet
workbook = load_workbook("Monthly_Expenses.xlsx")
sheet = workbook.active

# Define each function as described

def calculate_discounted_price(sheet, discount=0.10):
    for row in range(2, sheet.max_row + 1):
        amount = sheet[f"D{row}"].value
        if amount:
            discounted_price = amount * (1 - discount)
            sheet[f"E{row}"] = discounted_price

def calculate_category_totals(sheet):
    category_totals = {}
    for row in range(2, sheet.max_row + 1):
        category = sheet[f"C{row}"].value
        amount = sheet[f"D{row}"].value
        if category and amount:
            category_totals[category] = category_totals.get(category, 0) + amount
    print(category_totals)

def add_grand_total(sheet):
    total = sum(sheet[f"D{row}"].value for row in range(2, sheet.max_row + 1))
    if sheet[f"D{row}"].value:
        sheet[f"D{sheet.max_row + 1}"] = total
```

```

def highlight_expenses_above_threshold(sheet, threshold):
    fill = PatternFill(start_color="FFFF00", end_color="FFFF00",
fill_type="solid")
    for row in range(2, sheet.max_row + 1):
        amount = sheet[f"D{row}"].value
        if amount and amount > threshold:
            for col in "ABCDE": # Apply fill across the row
                sheet[f"{col}{row}"].fill = fill

# Call functions
calculate_discounted_price(sheet)
calculate_category_totals(sheet)
add_grand_total(sheet)
highlight_expenses_above_threshold(sheet, threshold=100)

# Save the modified file
workbook.save("Monthly_Expenses_Processed.xlsx")

```