

Summary

The aim of this project is to create a Python program that compares two specified ranges of cells in a spreadsheet, and identifies differences between them.

Your script should first prompt users to specify two ranges of identical size, on different worksheets of a spreadsheet. It should then compare the two ranges cell by cell, and create a new worksheet where it lists all differences between corresponding cells. Optionally, if the differing values are *numerical*, the script should also write formulas in another column on the new worksheet which calculate the percentage difference between the two numbers.

You will obviously need to write your script differently depending on whether you're targeting an Excel workbook or a Google Sheets spreadsheet (or better yet, both!), but from an end-user perspective, the programs should work the same.

The example Excel file you can test your solution against ("Employee Sales.xlsx") can be found in the course resources for this section, while a link to copy the example Google Sheets file (also "Employee Sales") can be found in a PDF in this section's course resources.

Below are detailed requirements for exactly what the program should accomplish:

1. Load the Spreadsheet:

Load the "Employee Sales" workbook into a Python program. This workbook consists of two worksheets, each containing datasets with identical numbers of rows and columns.

2. Take User Input to Specify Worksheets and Ranges:

The program should interactively ask the user to specify two worksheets and corresponding cell ranges within the loaded workbook that they want to compare. The user should provide:

- a. **Worksheet Names:** The user should be prompted to enter the name of the first worksheet and then the name of the second worksheet. This should be the exact name of the worksheets as they appear in the workbook.
- b. **Cell Ranges:** The user should be prompted to specify the cell range on each worksheet they want to compare. They need to input this range in the form commonly used in Excel or Google Sheets, like "A1:C9".

Optionally, after the user provides the names of the two worksheets and their corresponding cell ranges, the program should check if these ranges have the same dimensions (i.e., the same number of rows and columns).

If they don't match, the program should display an error message and prompt the user to input the worksheet names and ranges again. This process should continue in a loop until the user specifies two cell ranges with identical dimensions. This ensures that every cell in the first range has a corresponding cell in the second range, enabling a meaningful comparison.

3. Perform Comparison Between Specified Cell Ranges:

Once the user has specified two cell ranges of identical dimensions on different worksheets, the program should compare each cell in the first range with the corresponding cell in the second range.

The process can be thought of like this:

- a. Iterating Over Rows and Columns: The program should go row by row, and within each row, cell by cell (i.e., column by column) to access each individual cell in both ranges. For example, it would first compare the cell in the first row and first column of the first range with the cell in the first row and first column of the second range, then move on to the cell in the first row and second column, and so on.
- b. Comparing Cell Values: For each pair of corresponding cells, the program should check whether the value in the cell from the first range is the same as the value in the cell from the second range. This involves a direct comparison of the two values, whether they're numbers, strings, or some other data type.
- c. Recording Differences: If a pair of corresponding cells have different values, this is a "difference" that should be recorded programmatically (I recommend a list of lists for this). The program should store the following information about each difference:
 - i. Which row it occurred in
 - ii. Which column (using the column header)
 - iii. The differing value from the first range
 - iv. The differing value from the second range

In the end, the program will have examined every pair of corresponding cells in the two ranges and noted down any differences between their values. These differences will be used in the next steps of the project.

4. Create New Worksheet and Write Differences:

- a. Programmatically create a new worksheet named "Diffs".
- b. Programmatically write the following column headers to cells A1:E1 of the "Diffs" worksheet:
 - i. "Row"
 - ii. "Column"

- iii. "Value 1"
- iv. "Value 2"
- v. "Delta" (Only needed if you're completing the optional step 5 below)

- c. Then, for each difference you recorded in step 3, write a row in this new worksheet. The first cell in the row should contain the row number (of the original range), the second should contain the column header, the third should contain the value from the first range, and the fourth should contain the value from the second range.

For example, the (hypothetical) first difference your program finds could be represented in cells **A2:D2** as 2, "Employee Name", "John Smith", "Jon Smith", where "2" is the row number, "Employee Name" is the column, and "John Smith" and "Jon Smith" are the differing cell values.

5. OPTIONAL: Calculate Percentage Differences for Numeric Values:

Write **formulas** to the "Delta" column of the "Diffs" worksheet which calculate the *absolute difference* between "Value 1" and "Value 2" for each row, *if applicable*.

This operation should **only** be performed if the values in "Value 1" and "Value 2" are both numbers; in other words, if either "Value 1" OR "Value 2" is non-numeric, the corresponding cell in the "Delta" column should be left blank.

The formula (in spreadsheet terms) for the absolute difference between two numbers is **=ABS(value1, value2) / AVERAGE(value2, value2)**.

6. OPTIONAL: Format "Delta" Column:

Format the cells in the "Delta" column as follows:

- The text color should be **red**
- The values (which are percent differences between two numbers) should also be *formatted* as percentages.

We haven't explicitly covered how to programmatically apply *number* formats so far, so here's how to specify a percentage number format with openpyxl and gspread, respectively:

Openpyxl:

- Import "numbers" from openpyxl.styles: **from openpyxl.styles import Font, numbers**
- Given a variable "cell" that contains a cell object, format the cell with a percentage number format as follows: **cell.number_format = numbers.FORMAT_PERCENTAGE**

Gspread:

- Inside of your whatever dictionary you use to define your style for the “Delta” cells, add the following key-value pair:

```
"numberFormat": {  
    "type": "PERCENT"  
}
```

7. Finally, test your program against either the “Employee Sales.xlsx” Excel spreadsheet, or the “Employee Sales” Google Spreadsheet. The results should look like the following:

Row	Column	Value 1	Value 2	Delta
3	Employee Name	Kadyn Madden	Kadyn Maden	
3	Q1 Sales	328166	309434	5.875784191%
5	Employee Name	Jazmine Liu	Jasmine Liu	
7	Q4 Sales	179600	229123	24.23303802%
8	Employee Name	Remington Maynard	Remington Maynerd	
9	Q4 Sales	110824	89163	21.66240806%
11	Q2 Sales	225087	176103	24.41935243%
11	Q3 Sales	248226	215604	14.06636052%
17	Employee Name	Ivan Brennan	Ivan Brennen	
18	Q4 Sales	341707	361728	5.692352527%