Intro in RPA

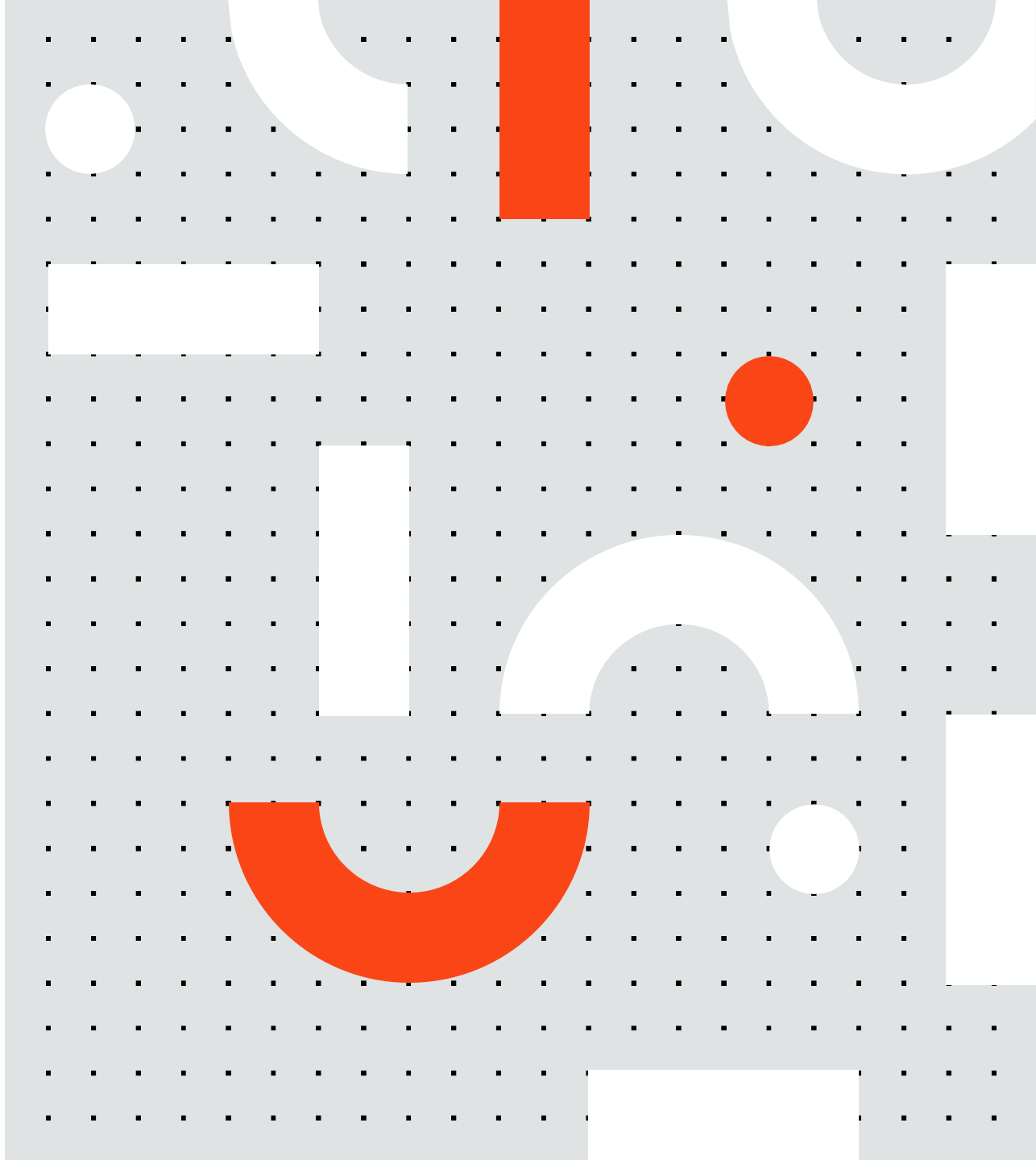
2nd Lab – 18/10/2023

Staicu Adelina Nicoleta
nicoleta.staicu@unibuc.ro

UiPath™

Get current time/weather in Bucharest (optional homework from Lab 1)

Objective: Get the current time/weather in Bucharest (or another region of your choice) using Google search (“Bucharest current time”)



Number order sorting

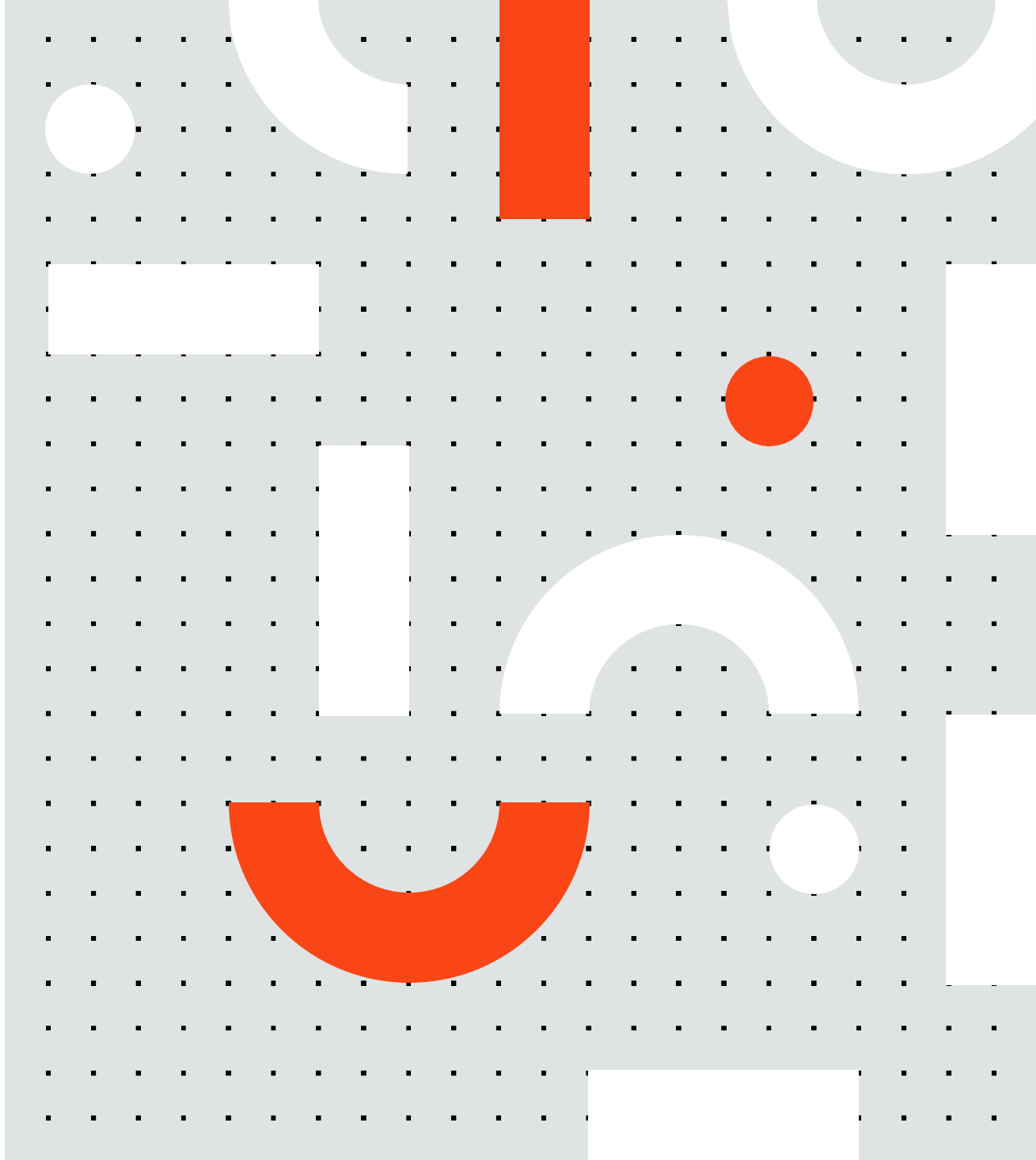
1. Number order sorting

Objective: To code a Robot in UiPath Studio to sort numbers in Ascending and Descending order in an excel file

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** and **"Assign"** activity.
✓ **"Comment"** and "**Annotation".**
✓ **"Excel Application Scope"** activity.
✓ **"Read Range"** and **"Write Range"** activity.
✓ **"Sort Data Table"** activity.

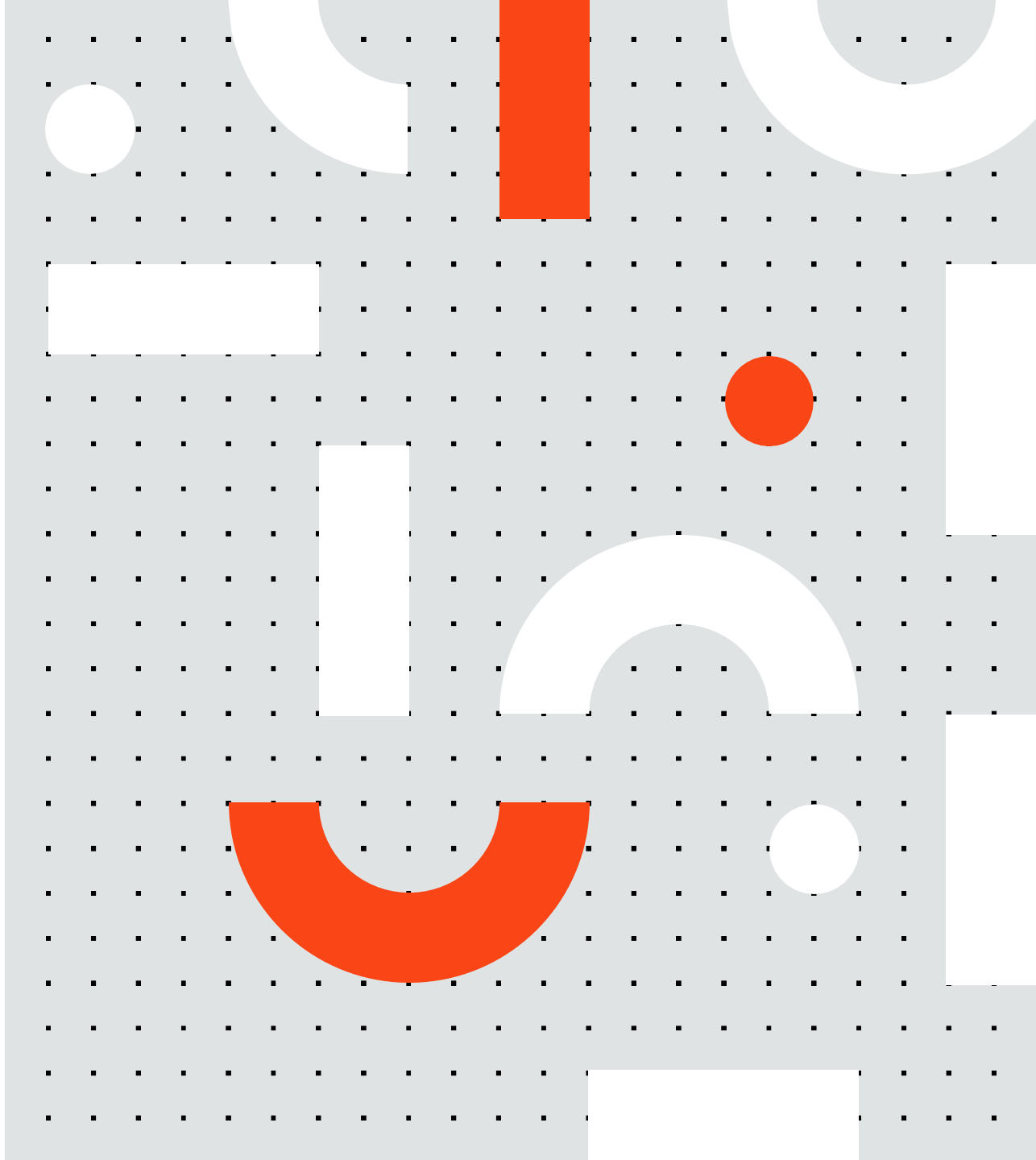


Number order sorting

1. Number order sorting

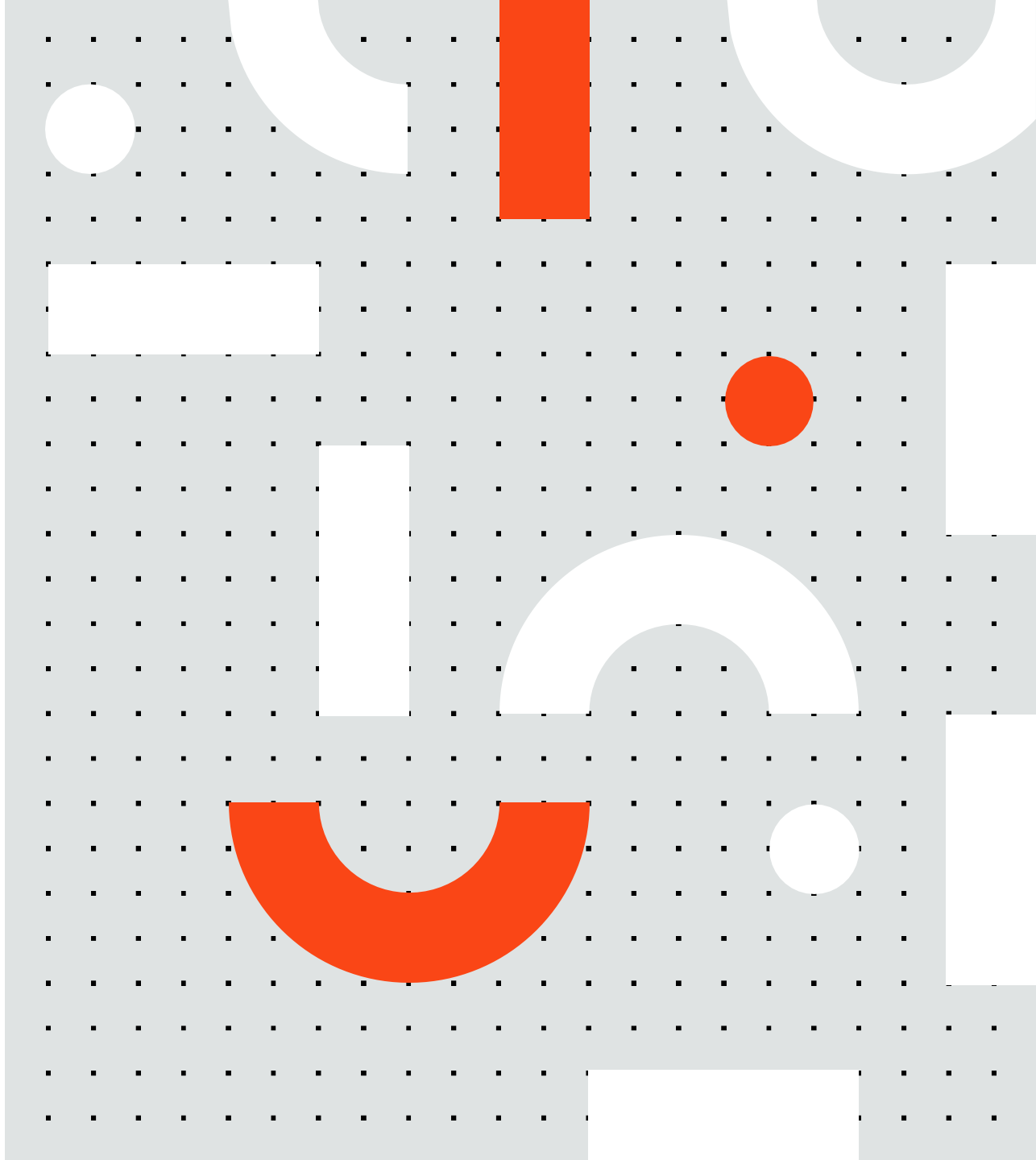
Algorithm:

Step 1: START
Step 2: Declare variables as ‘DataTable1’, ‘DateTable2’
Step 3: Add Excel Application scope activity to read the excel file
Step 4: Add Read Range Activity to Read the Sheet
Step 5: Add Sort Data Table activity and change order as Ascending
Step 6: Add Sort Data Table activity and change order as Descending



Step by Step process:

Step 1: Open UiPath Studio.
Step 2: Create the process and name it.
Step 3: Drag the **“Sequence”** activity from the activity panel and drop it in the workflow.
Step 4: Name the **“Sequence”** activity as **Sequence - ‘Sort numbers’.**
Step 5: Drag and Drop the “**Excel Application Scope”** activity in the **“Sequence”** activity.
Step 6: In the “**Excel Application Scope**” activity text field specify the name of the excel file to be used in the code. In this code, the name of the file used is "With_DataTable.xlsx,” so we will write this name in the text field.
Step 7: Change the name of “**Do**” sequence from the “**Excel Application Scope**” activity as **Do – ‘To read, write, and sort the data in the excel file’.**
Step 8: Right-click on the “**Do**” sequence and select **Annotation > Add Annotation ‘Read, write and sort the data in ascending and descending order by using Sort Data Table activity.**’
Step 9: Select the **ExcelReadRange** activity from the activities panel and drop in the sequence activity workflow.



Step by Step process:

Step 10: Change the name of "**ExcelReadRange**" activity as **ExcelReadRange – 'This activity is used to read an excel sheet'** and write the name of excel sheet as mentioned in the excel file used. In this case, the sheet name used is '**Sheet 1**'.

Step 11: Create two variables from the "Variables" panel in the "**ExcelReadRange**" activity as under:

Name	Variable Type	Scope	Default
DataTable1	DataTable	Do – 'To read, write, and sort the data in the excel file'	
DataTable2	DataTable	Do – 'To read, write, and sort the data in the excel file'	

Step 12: Declare the variable **DataTable1** in the "**Output**" Property of "**Read Range**" activity.

Step 13: Drag and drop the "**SortData Table**" activity and change the name it as – **SortData Table** – '**sort the data of the excel sheet in ascending order**'

UiPath™

Step by Step process:

Step 14: To sort the data in ascending order, write the following values in the **Properties panel** of **SortData Table** activity as mentioned below.

Input (Data Table)	Output (Data Table)	Name	Order
DataTable1	DataTable2	"S.no"	Ascending

NOTE:

- DataTable1 declare in the "Input Property" of "SortData Table" activity.
- DataTable2 declare in the "Output Variable" of "SortData Table" activity.
- S.no declare for the name of sorting column in the "SortData Table" activity

Step 15. Select **ExcelWriteRange** activity from the activities panel and drop it in the sequence workflow. Rename it as **ExcelWriteRange – 'Write the sorted data in the specified column of excel sheet'.**

Step 16: In the "**ExcelWriteRange**" property, write the following values in **ExcelWriteRange properties panel**:

Sheet Name	*Cell*	*Data Table*
"Sheet1"	"B2"	DataTable2

Note:

- Sheet1 declare in the "Sheet name" of the ExcelWriteRange activity.
- B2 declare in the "StartingCell" of the ExcelWriteRange activity
- DataTable2 declare in the "Input Property" of "Write Range activity"

UiPath™

Step by Step process:

Step 17: Drag and drop the "**SortData Table**" activity and name it as – **SortData Table - 'This activity will sort the data of the excel sheet in descending order'.**

Step 18: To sort the data in descending order, write the following values in the Properties of **SortData Table activity** as mentioned below.

Input (Data Table)	Output (Data Table)	Name	Order
DataTable1	DataTable2	"S.no"	Descending

Note:

- DataTable1 declare in "Input Property" of "Sort DataTable activity".
- DataTable2 declare in "Output Property" of "Sort DataTable" activity.
- S.no declare in the Sorting column of SortData Table activity.

Step 19: Select **ExcelWriteRange** activity from the activities panel and drop it in the sequence workflow. Change the name it as **ExcelWriteRange** – '**Write the sorted data in the specified column of excel sheet**'.

Step 20: In the "**ExcelWriteRange**" activity properties panel, write the following values in the **ExcelWriteRange property panel** as mentioned values:

Sheet Name	*Cell*	*Data Table*
"Sheet1"	"C2"	DataTable2

Note:

- Sheet1 declare in the "Sheet name" of the ExcelWriteRange activity.
- C2 declare in the "StartingCell" of the ExcelWriteRange activity.

DataTable2 declare in the "Input Property" of "ExcelWriteRange" activity.

UiPath™

Compare two columns of a spreadsheet

1. Compare two columns of a spreadsheet

Objective: To code a Robot in UiPath Studio to compare two columns in a spreadsheet and display the result in third column if it is a match or not.

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** and **"Assign"** activity.
✓ **"Comment"** and "**Annotation".**
✓ **"Excel Application Scope"** activity.
✓ **"Read Range"** and **"Write Cell"** activity.
✓ **"For Each Row"** activity.
✓ **"If"** activity and how to set conditions.

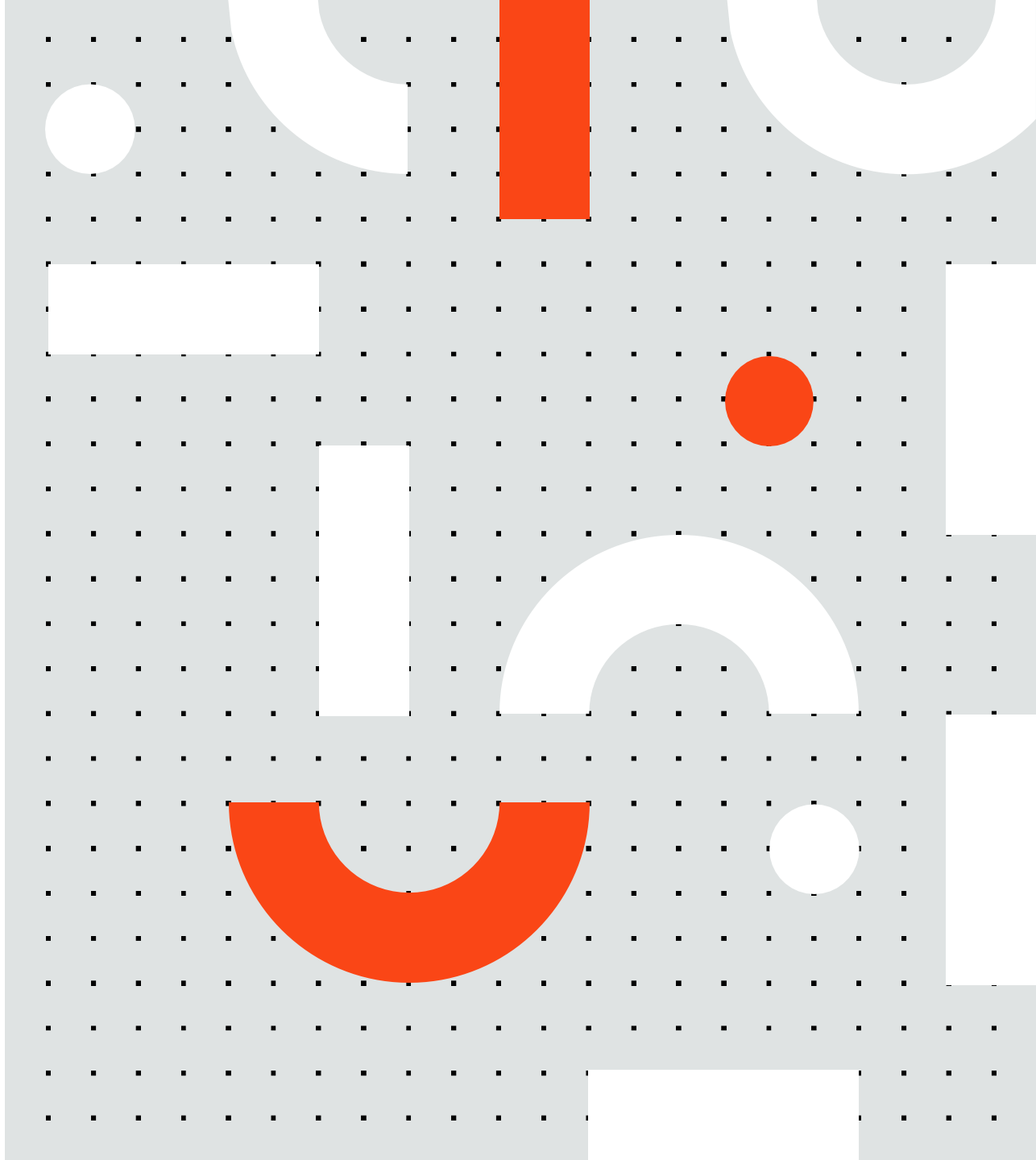
UiPath™

Compare two columns of a spreadsheet

1. Compare two columns of a spreadsheet

Algorithm:

Step 1: START
Step 2: Read all the rows in Column1 and Column2 of Excel
Step 3: Declare a variable RowNumber
Step 4: If Column1 = Column2 then it is a match else it is not
Step 5: RowNumber = RowNumber + 1
Step 6: STOP



Step by Step process:

Step 1: Open UiPath Studio.
Step 2: Create the process and name it.
Step 3: Drag the **"Sequence"** activity from the activity panel and drop it in the workflow.
Step 4: Name the **"Sequence"** activity as **Sequence - 'Compare two columns'.**
Step 5: Drag and Drop the "**Excel Application Scope"** activity in the **"Sequence"** activity.
Step 6: In the "**Excel Application Scope**" activity text field specify the name of the excel file to be used in the code. In this code, the name of the file used is "Compare Columns.xlsx," so we will write this name in the text field.
Step 7: Change the name of "**Do**" sequence from the "**Excel Application Scope**" activity as **Do – 'To read, compare, and write the data in the excel file'.**
Step 8: Right-click on the "**Do**" sequence and select **Annotation > Add Annotation 'Read an excel file for each row, check if column 1 and column 2 are equal and write result'**
Step 9: Select the **ExcelReadRange** activity from the activities panel and drop in the sequence activity workflow.

UiPath™

Step by Step process:

Step 10: Change the name of "**ExcelReadRange**" activity as **ExcelReadRange – 'This activity is used to read an excel sheet'** and write the name of excel sheet as mentioned in the excel file used. In this case, the sheet name used is '**Sheet 1**'.

Step 11: Create two variables from the "Variables" panel in the "**ExcelReadRange**" activity as under:

Name	Variable Type	Scope	Default
DataTable1	DataTable	Do – 'To read, compare and write the data in the excel file'	
RowNumber	Int32	Do – 'To read, compare and write the data in the excel file'	

Step 12: Declare the variable **DataTable1** in the "**Output" Property of "Read Range"** activity.

Step 13: Drag and drop the "**For Each Row**" activity

Step 14: Declare **Row** in **ForEach** and **DataTable** in **in** section.

Step 15: Drag the "**IF**" activity from the activity panel and drop it in the workflow.

Step 16: Inside the **"If"** activity write the condition **"cint(row(0)) = cint(row(1))"**.

Step 17: In the **Then** section of "**If**" activity add a "**Write Cell**" Excel activity and change the name from "**Write Cell" as Write Cell - 'Write "Match" in each cell if condition is met'.**

UiPath™

Step by Step process:

Step 18: In the "**Write Cell**" activity properties panel, write the following values in the property panel as mentioned below:

Sheet Name	*Range*	*Value*
"Sheet1"	"C"+RowNumber.ToString	"Match"

Step 19: In the **Else** section of **"If"** activity add a "**Write Cell**" Excel activity and change the name from "Write Cell" as Write Cell - 'Write "Not a Match" in each cell if condition is not met'

Step 20: In the "**Write Cell**" activity properties panel, write the following values in the property panel as mentioned below:

Sheet Name	*Range*	*Value*
"Sheet1"	"C"+RowNumber.ToString	"Not a Match"

Step 21: Drag and drop the "**Assign**" activity below the "If" activity.

Step 22: Change the "**Assign" activity name Assign - 'Increment the RowNumber variable by 1**'

Step 23: Declare the **'RowNumber'** variable in the **To** box and '**RowNumber+1**" in the value box.

UiPath™

Extracting data from a website

1. Extract data from a website

Objective: To code a Robot in UiPath Studio to scrape data from a website and store it in .CSV File.

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"**
✓ **"Comment"** and "**Annotation".**
✓ **"Open Browser"** activity.
✓ **"Type into"** activity.
✓ **"Browser scope"** activity.
✓ **"Extract data"** activity.
✓ **"Write CSV"** activity.

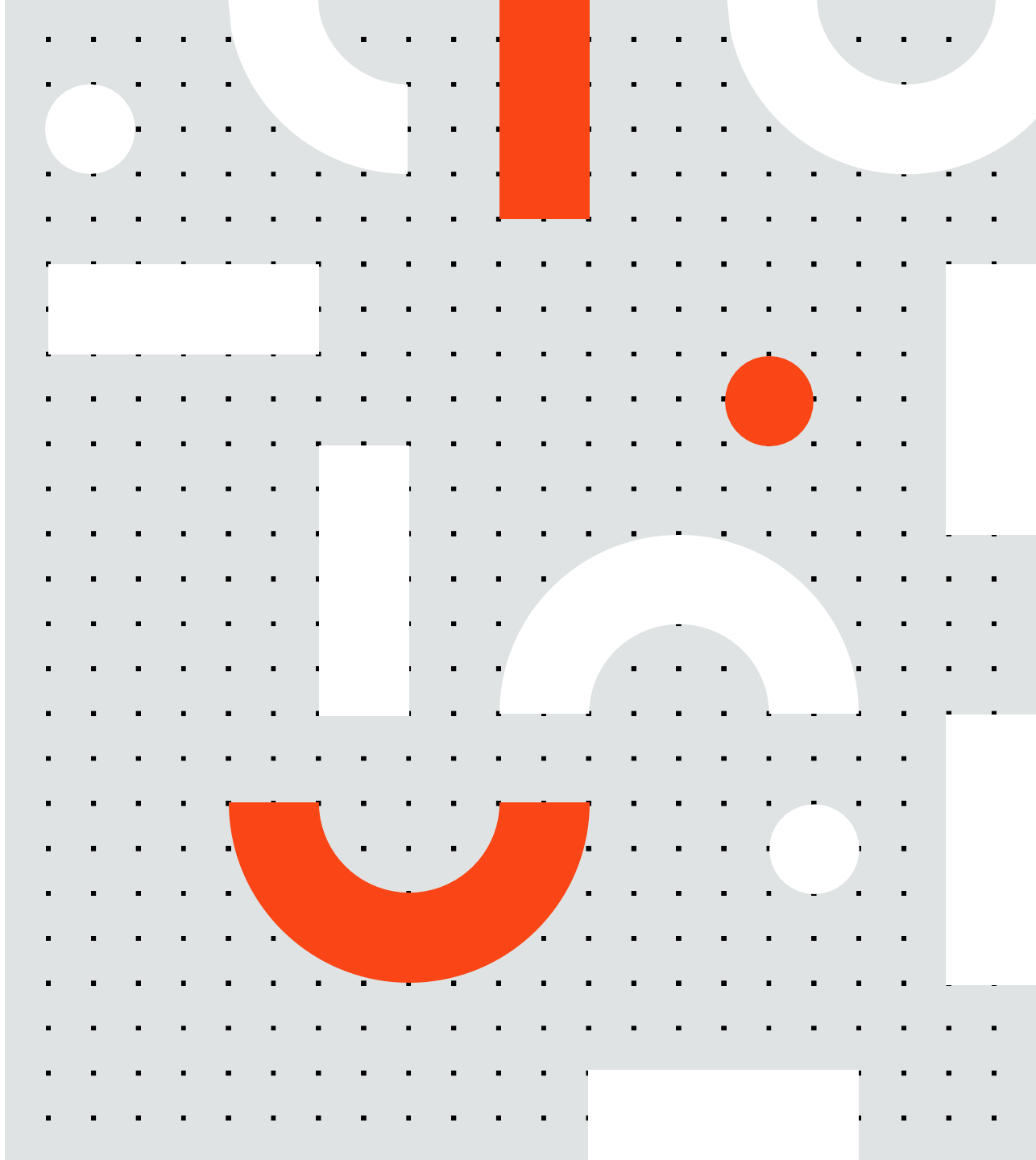
UiPath™

Extracting data from a website

1. Extracting data from a website

Algorithm:

Step 1: START
Step 2: Open the URL using Open Browser Activity
Step 3: Declare variables as ‘CSVFile’, ‘ExtractedDT’, ‘SearchItem’, ‘URL’.
Step 4: Use the variables in the different activity blocks to search, find the given item
Step 5: Output the result in the write csv file activity
Step 6: STOP



Step by Step process:

Step 1: Open UiPath Studio.
Step 2: Create the process and name it.
Step 3: Create new xaml file as **Sequence**.
Step 4: Name the new file **"Extract data from website".**
Step 5: Drag the '**Open browser'** activity and drop it in the workflow
Step 6: Create an **in argument** corresponding to the URL (e.g. in_URL). Value set in Main: www.amazon.com. This argument will be the input for the '**Open browser'** activity.
Step 7: Drag the **Type Into** activity and drop it in the workflow
Step 8: Change the properties accordingly
Step 9: Create an **in argument** e.g. **in_SearchItem** – contains the name of the product we are searching for (e.g. iPhone). This argument will represent the **input for Type into activity.**
Step 10: Add "Enter" key in the Type Into activity. Check **"Simulate Type"** property deactivated.
Step 11: Drag and drop "Attach Browser" activity.
Step 12: Drag and drop "Extract Data" activity into the previous added "Attach Browser"
Step 13: Create the **output argument** of the Extract Data activity e.g. **out_ExtractedInfoDT**. Type DataTable.

UiPath™

Step by Step process:

Step 14: Create a new xaml file as **Sequence**.
Step 15: Name the new file **"Write extracted data into csv".**
Step 16: Create the **input argument** containing the data table extracted previously e.g. in_ExtractedInfoDT.
Step 17: Drag the "**Write CSV**" activity from the Activity panel.
Step 18: Create an in argument that will store the path to the .csv file e.g. CSVFilePath. Provide the argument to "Write CSV" activity.
Step 19: The input argument containing the data table will represent the Input for "Write CSV" activity.