

Robotic Process Automation

UiPath Hands-On Guide

for

RPA EP0705

(Elective Module)



Versions of UiPath

- 1. UiPathCommunity Edition is free** for individual developers:
 - open source projects,
 - academic research,
 - education,
 - small professional teams

- 2. Enterprise Version 60-day trial Version with full functions**

UiPath Free Trial

INDIVIDUALS & SMALL TEAMS

Community Cloud**Try it**

For individual RPA developers and small teams. Upgrade to enterprise versions at any time.

ENTERPRISE USE

Studio**Try it**

For individual enterprise developers wanting to try out UiPath Studio.

ENTERPRISE USE

Enterprise Cloud**Try it**

Cloud-based version of the entire RPA enterprise platform (Studio, Robots, Orchestrator). Currently in Preview 1

ENTERPRISE USE

Enterprise Server**Try it**

On-premises version of the entire enterprise automation platform (Studio, Robots, Orchestrator).

Free trial period

Always free**60 days****60 days****60 days**Studio 1**2** licenses**1** licenses**2** licenses

—

Robots included 2**2** attended; **1** unattended 2**1** attended**2** attended; **1** unattended 2

—

Additional robots

— 1

—

Unlimited; license as you need**Unlimited;** license as you need

AI Computer Vision

30 megapixels /min.

—

240 megapixels /min.

Infrastructure dependent

Automation capacity

3 robots

—

Enterprise scale from UiPath Enterprise Cloud

Enterprise scale with your infrastructure

Availability

99.5%+

—

99.5%+, with UiPath support

High availability architectures available

User management

—

—

Centralized enterprise user management

Centralized enterprise user management

Software Updates

UiPath-managed updates

Self-managed updates

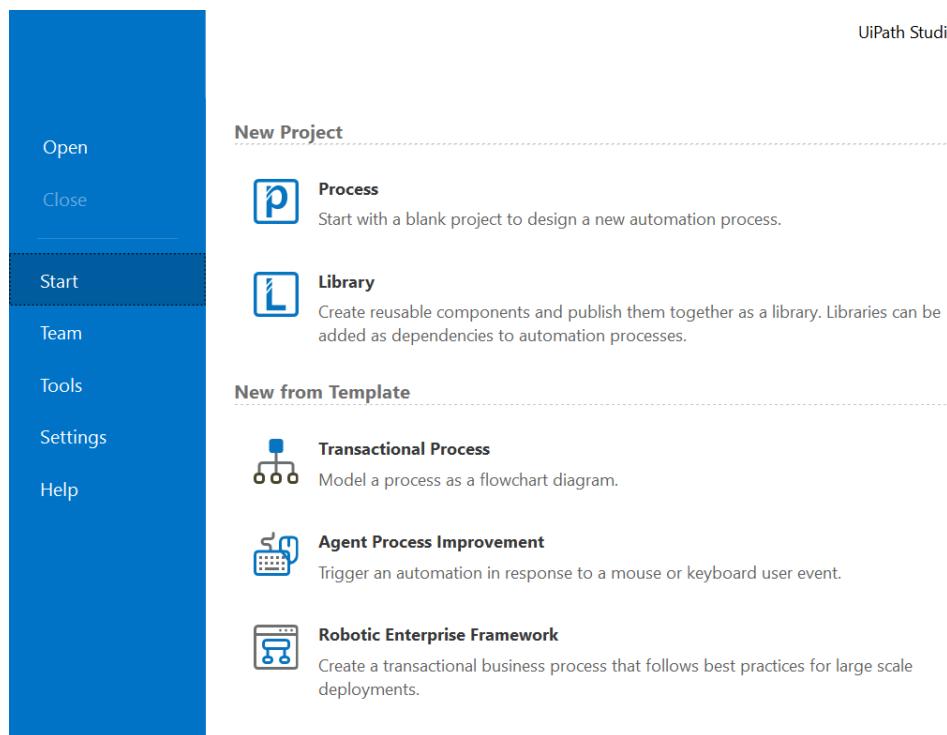
Jointly managed updates 3

Self-managed updates

Starting the UiPath Studio

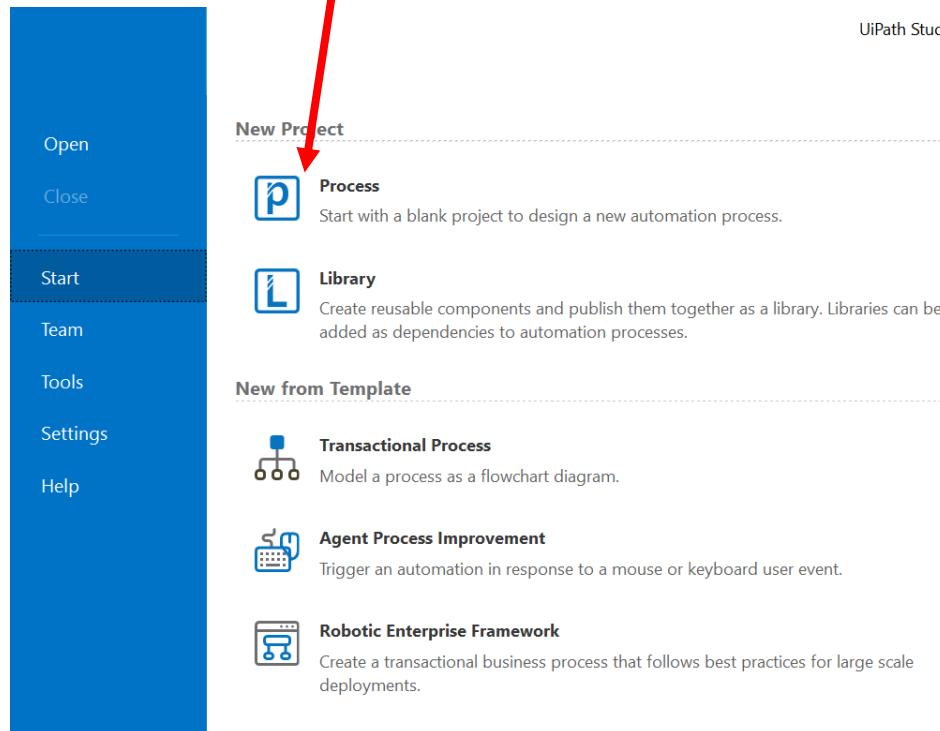


1. Double click on the “UiPathStudio” icon on your desktop to launch the RPA.
2. If the shortcut icon is not on your desktop, you can do a window search and pin it to the desktop.

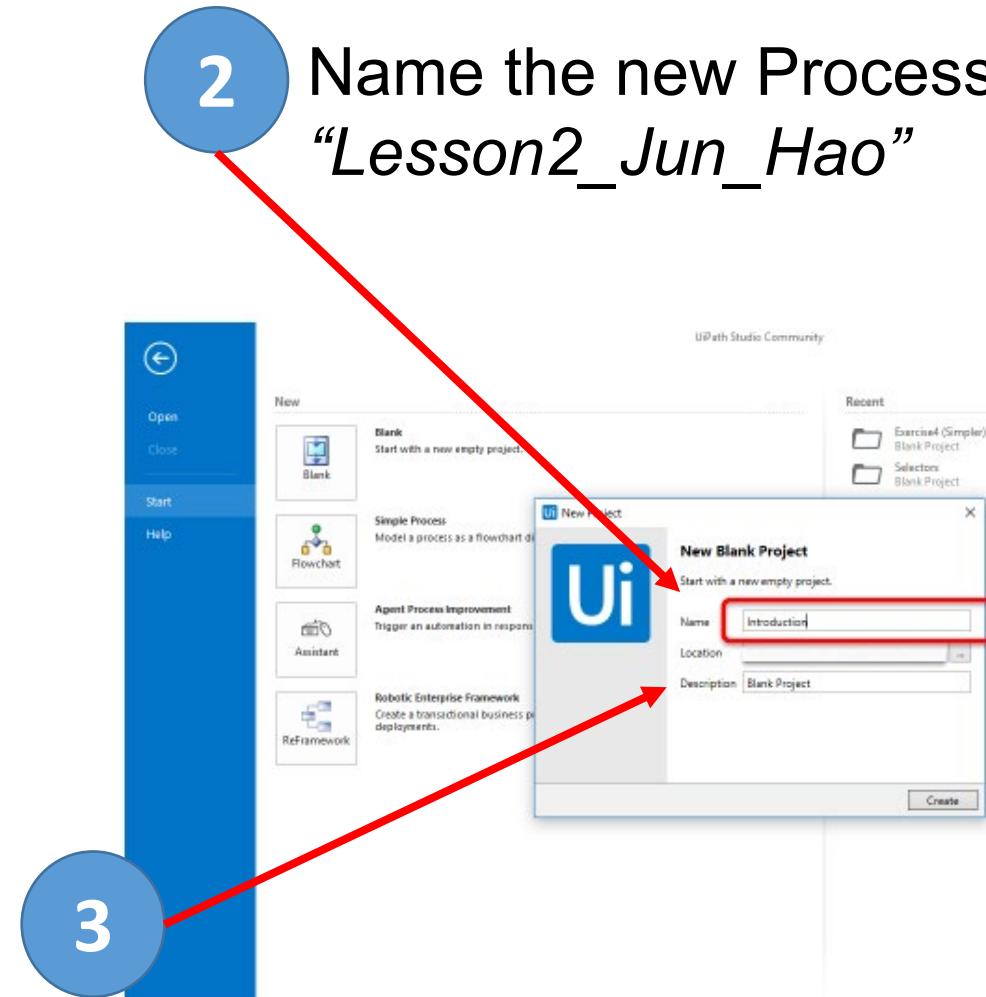


Creating a New Process/Project

1 Select
“Process”

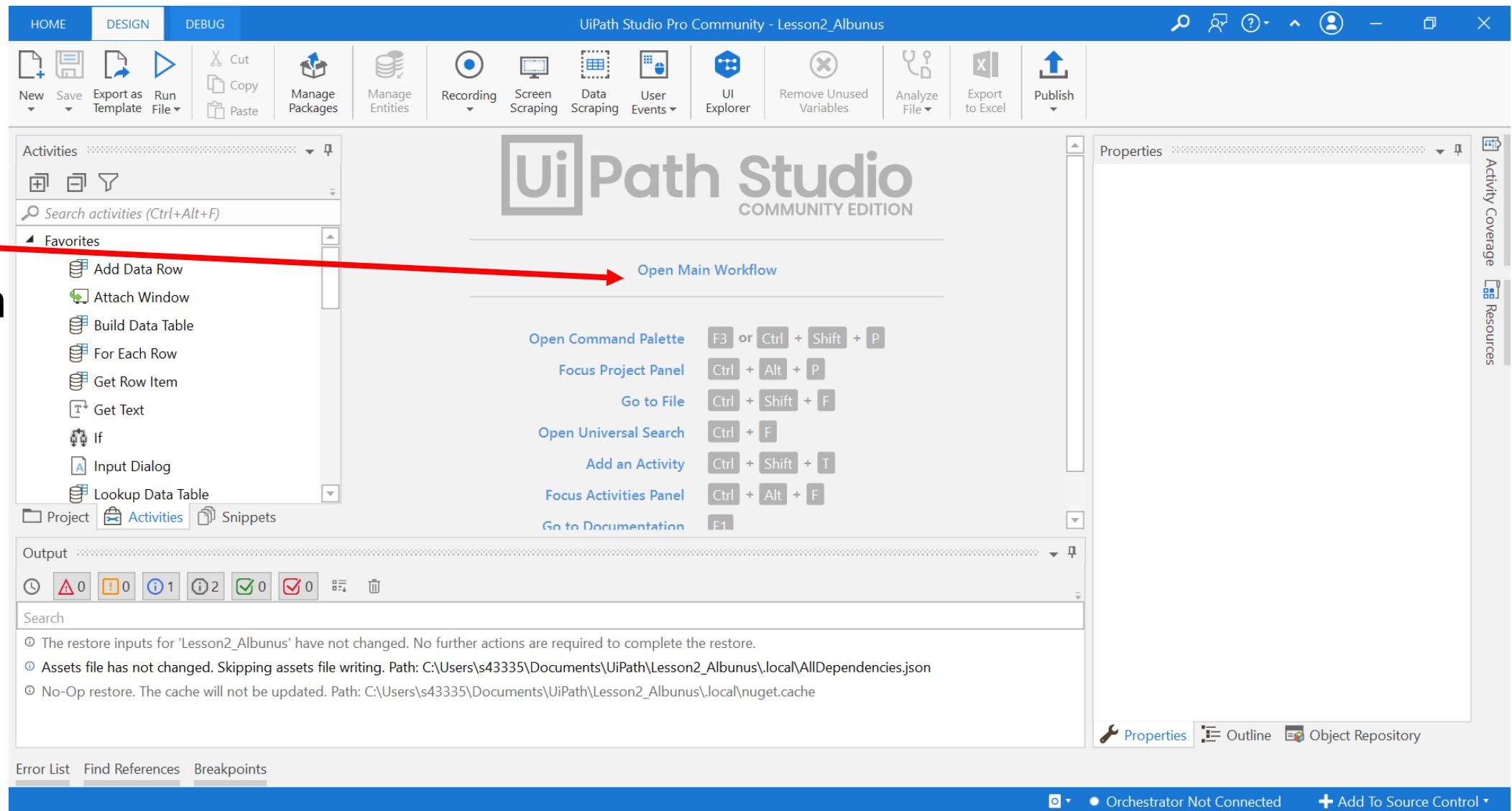


2 Name the new Process e.g.
“Lesson2_Jun_Hao”

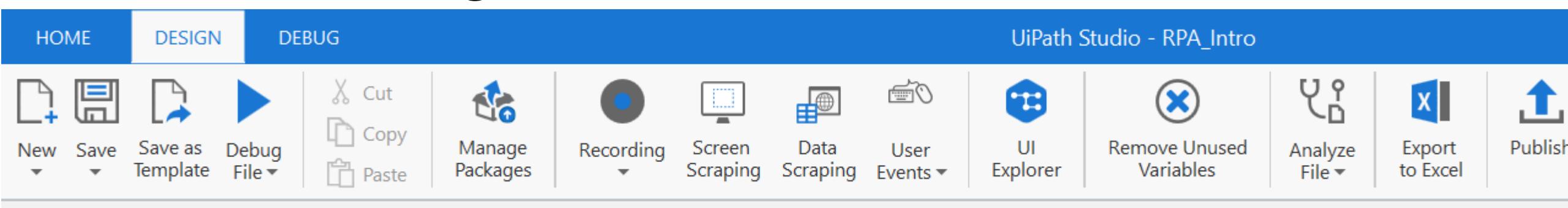


3 Provide description e.g.
“RPA Lesson 2, Ex 1.0 to Ex 1.2”

UiPath Studio Screen after launch



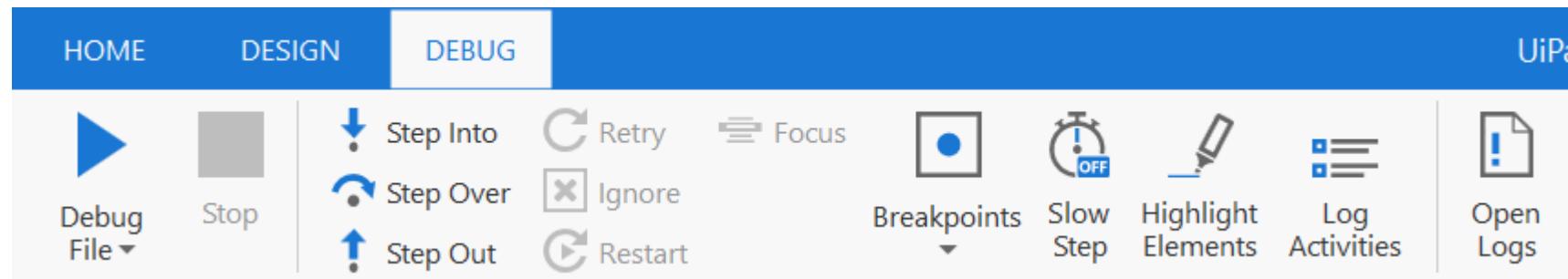
“Design” Toolbar



Design

- Create or launch sequences, flowcharts or state machine diagrams
- Manage Packages
- Access wizards
- Create and Manage variables
- Inspect user interface elements from third-party apps
- Export scripts to Excel

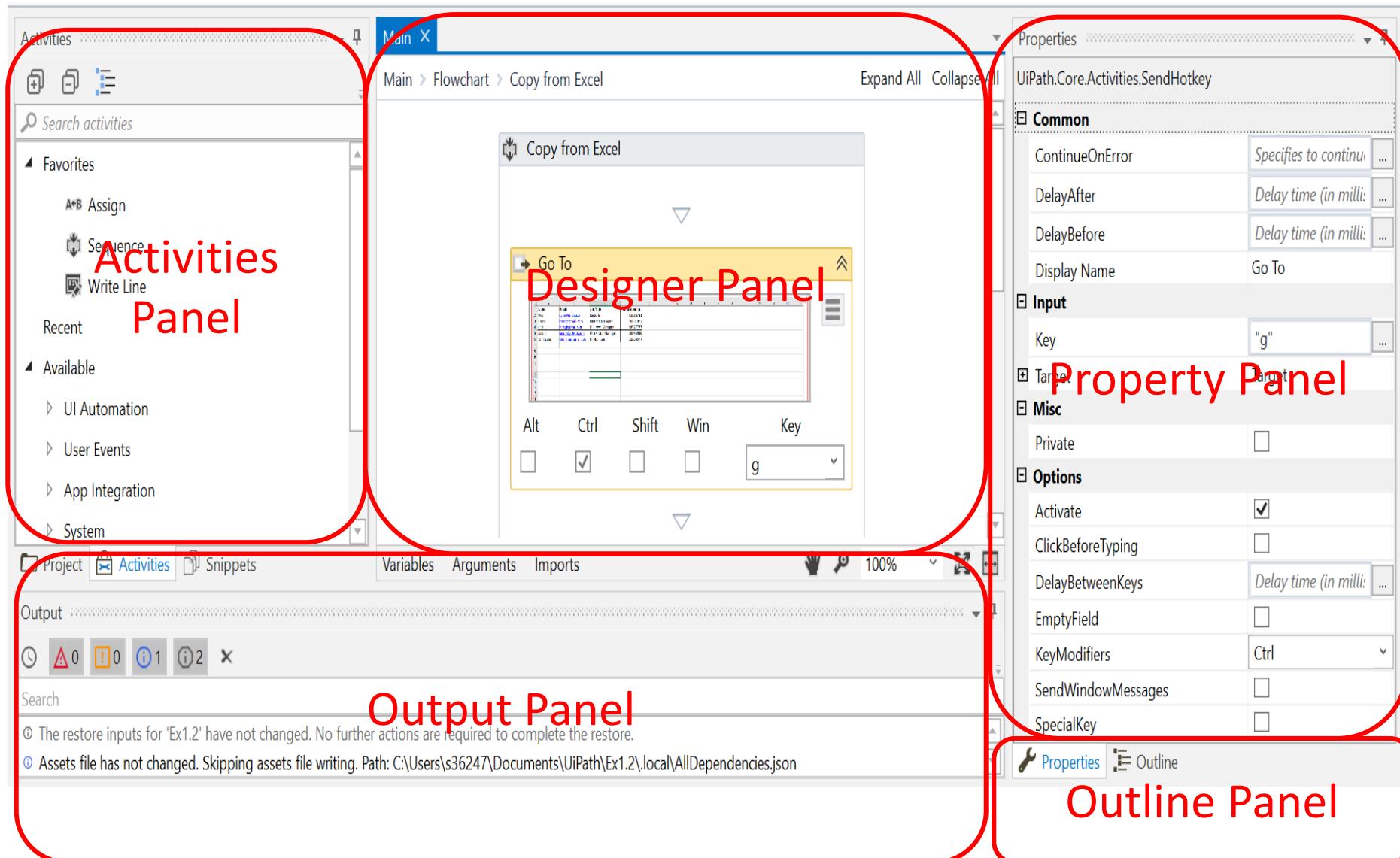
“Debug” Toolbar



Execute

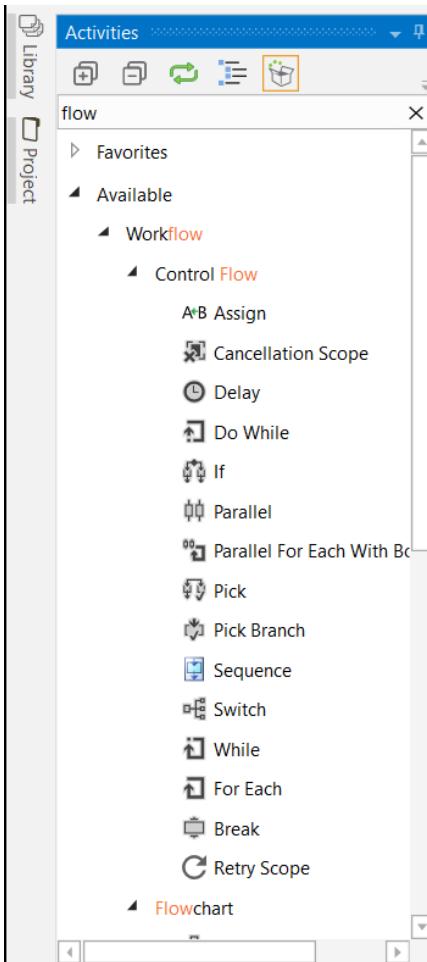
- Run or stop projects
- Start the debug process
- Slow down steps
- Open logs

UiPath Studio Window



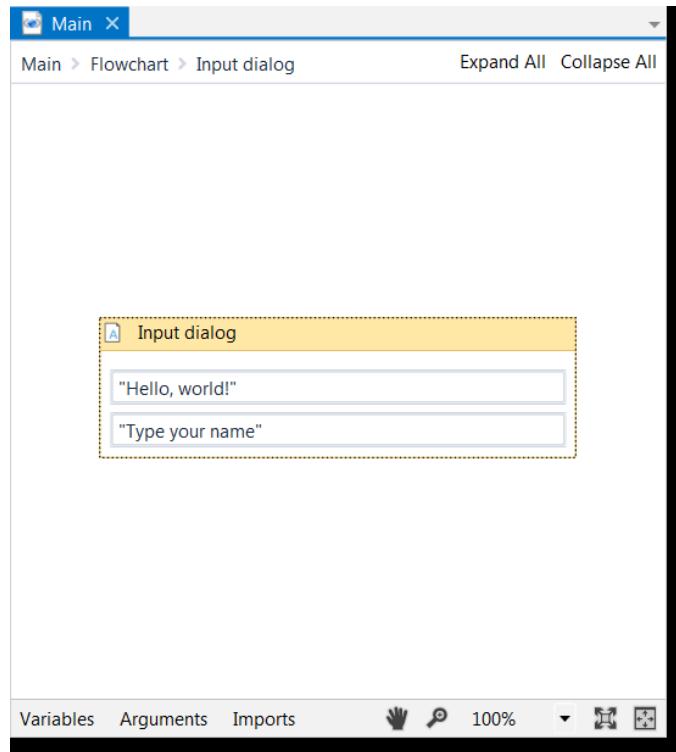
Activities Panel

The **Activities** Panel contains the different key actions required to perform different tasks in UiPath.



- Quick access to all available activities that can be dragged to the current project.
- Search box feature.
- **Show Activities** - hide or show the **Favourites**, **Recent**, and **Available** folders of activities.
- **Manage Projects** – View and edit different projects

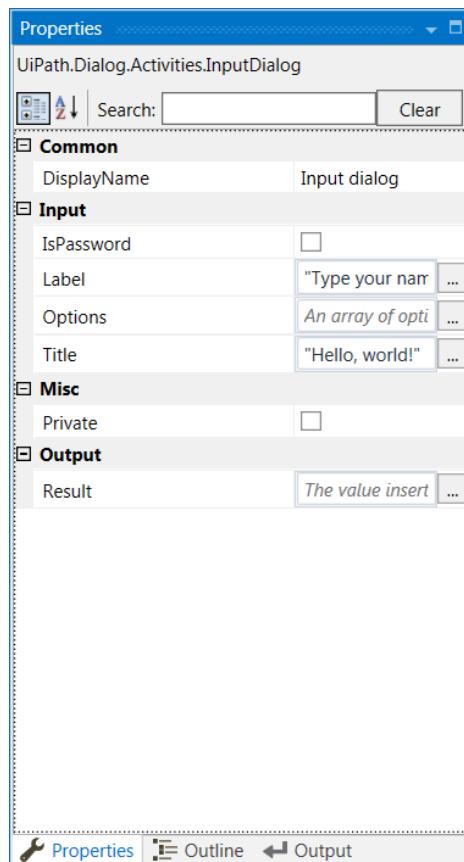
Designer Panel



The **Designer** panel

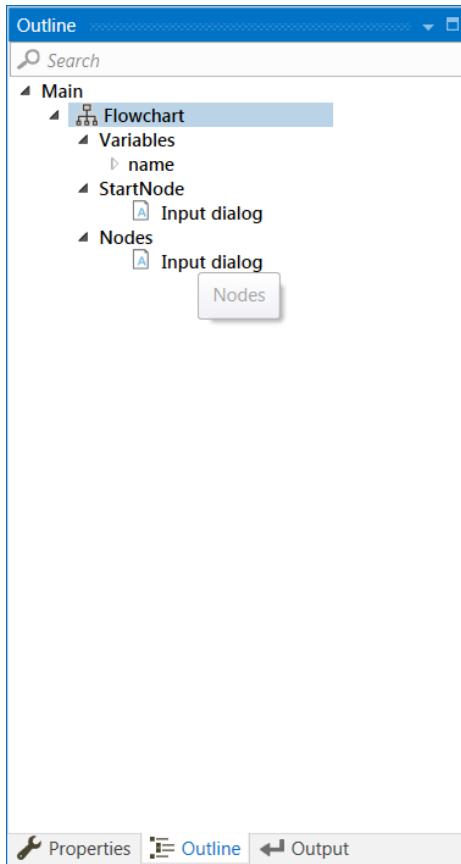
- Displays your current automation project.
- Enables you to make changes to it.
- Provides quick access to variables, arguments and imports.

Properties Panel



The **Properties** panel provides detailed properties about each activity and enables you to view and change the properties of a selected activity.

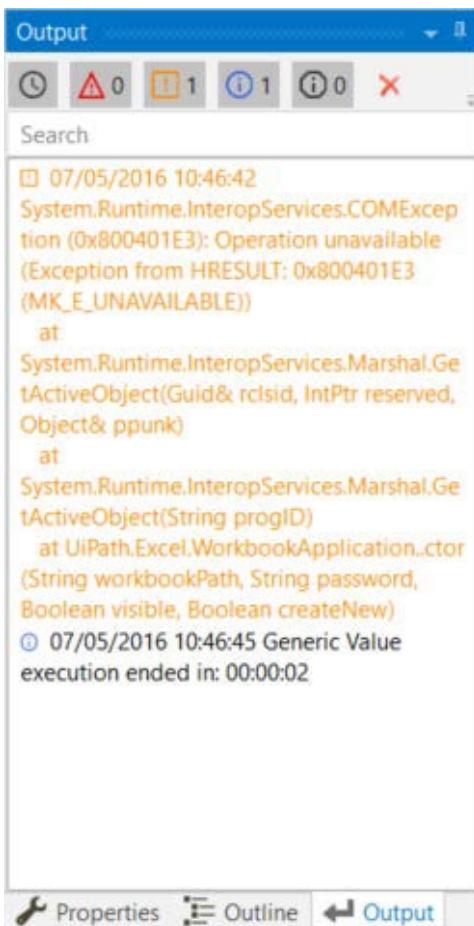
Outline Panel



The **Outline** panel

- Displays the project hierarchy, all available variables and nodes.
- Includes a search box. This enables you to easily navigate through large automations.

Output Panel

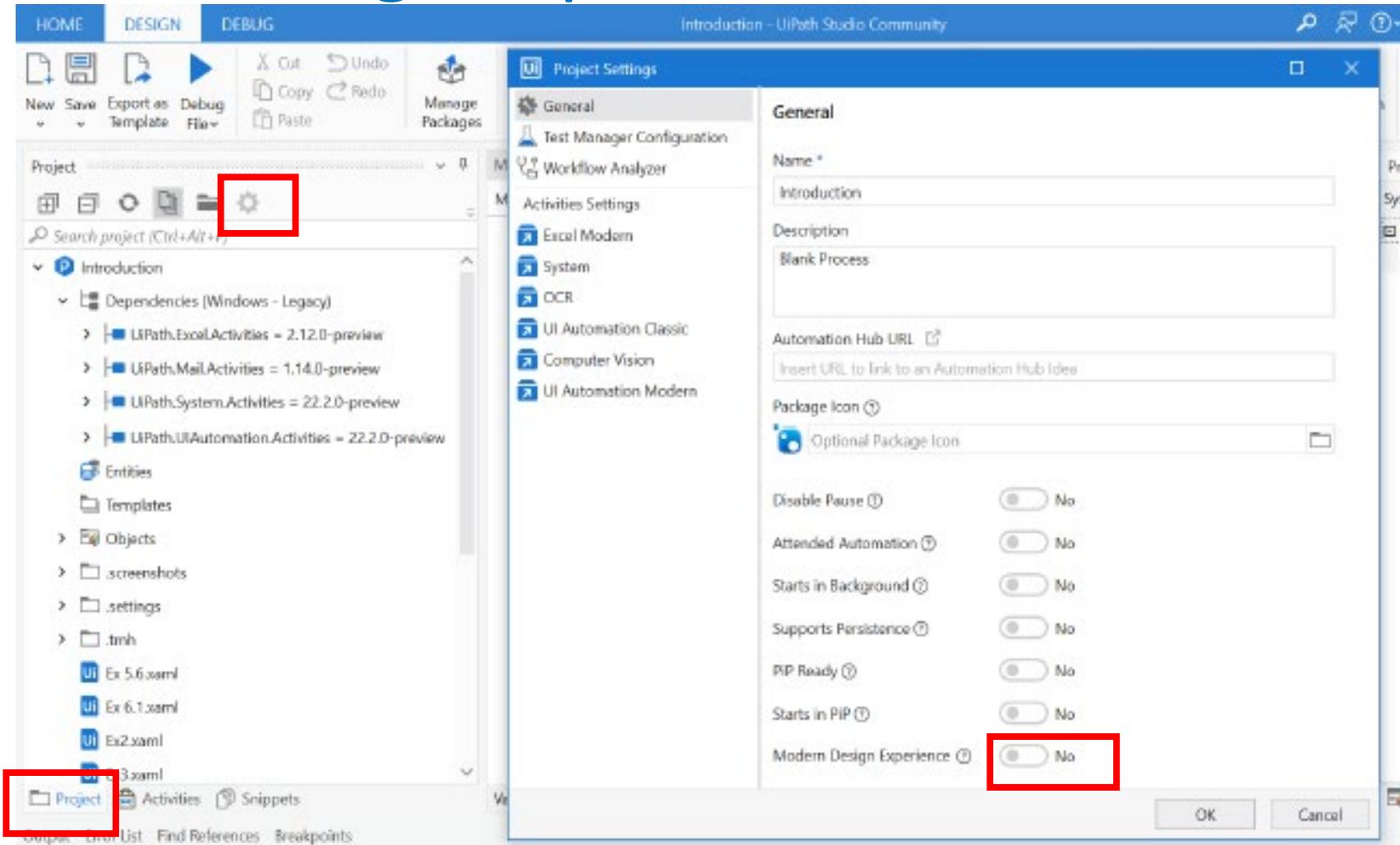


The **Output** panel

- Displays the output of the **Log Message** or **Write Line** activities.
- Displays logs when the debug mode is activated.

How to change from modern to classic design experience in UiPath

Go to the project settings and disable the modern design experience here.

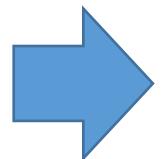




Ex1.0 "Hello World"



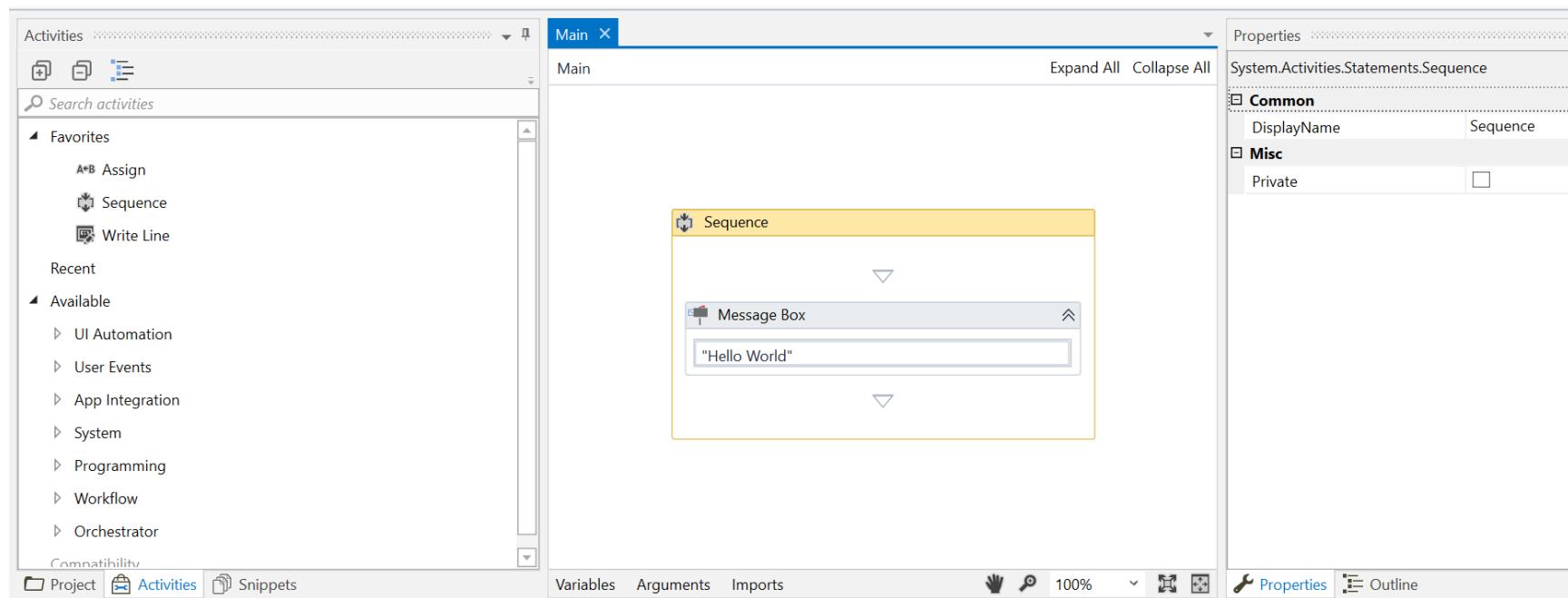
1. In UiPath Studio, create a new Process and name it “*Robot1_XXXXXX*” with description as “*RPA Lesson2 from Ex1.0 to Ex4.0*”.
XXXXXX is your first name, e.g. Jun_Hao or Jeremy etc
2. Open Main Workflow
3. Create a Workflow to display **Message Box** when “*Run File*” is clicked.
4. Refer to Ex1 in “RPA UiPath Hands-On Guide” for step-by-step instruction



Exercise 1.0

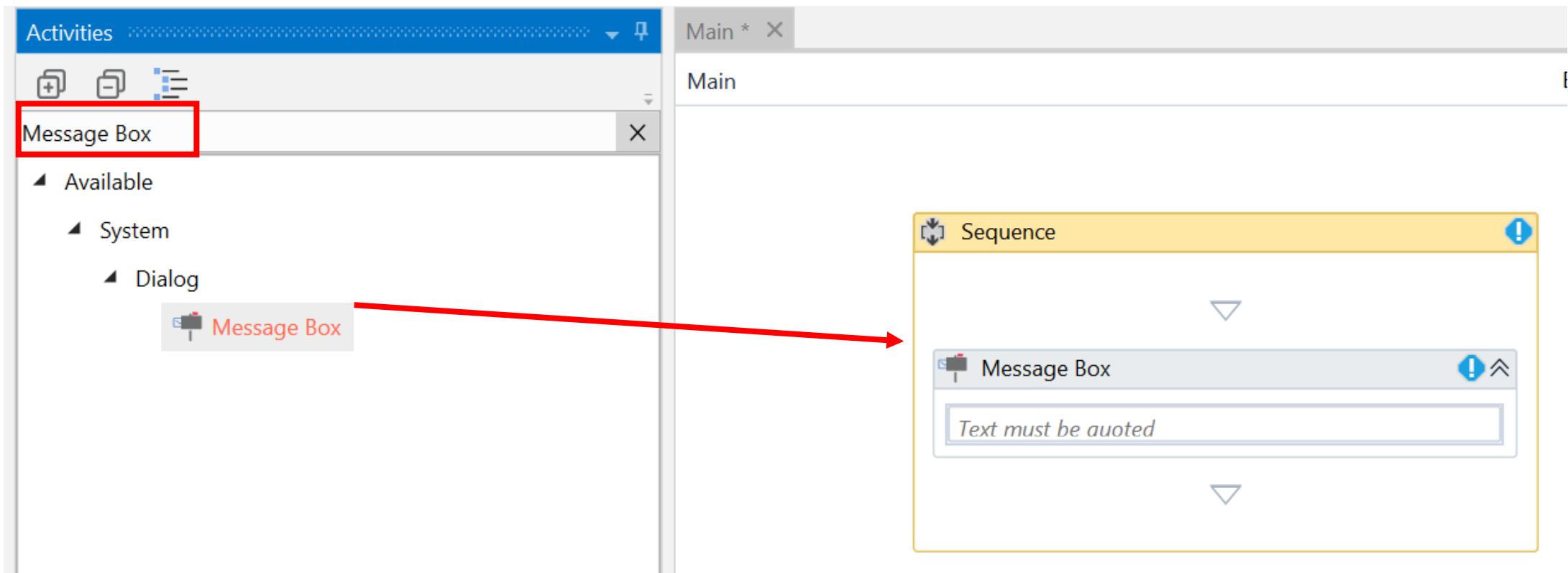
Exercise 1.0

In this exercise, we are going to get the RPA to output a message, “Hello World!” using a Message Box



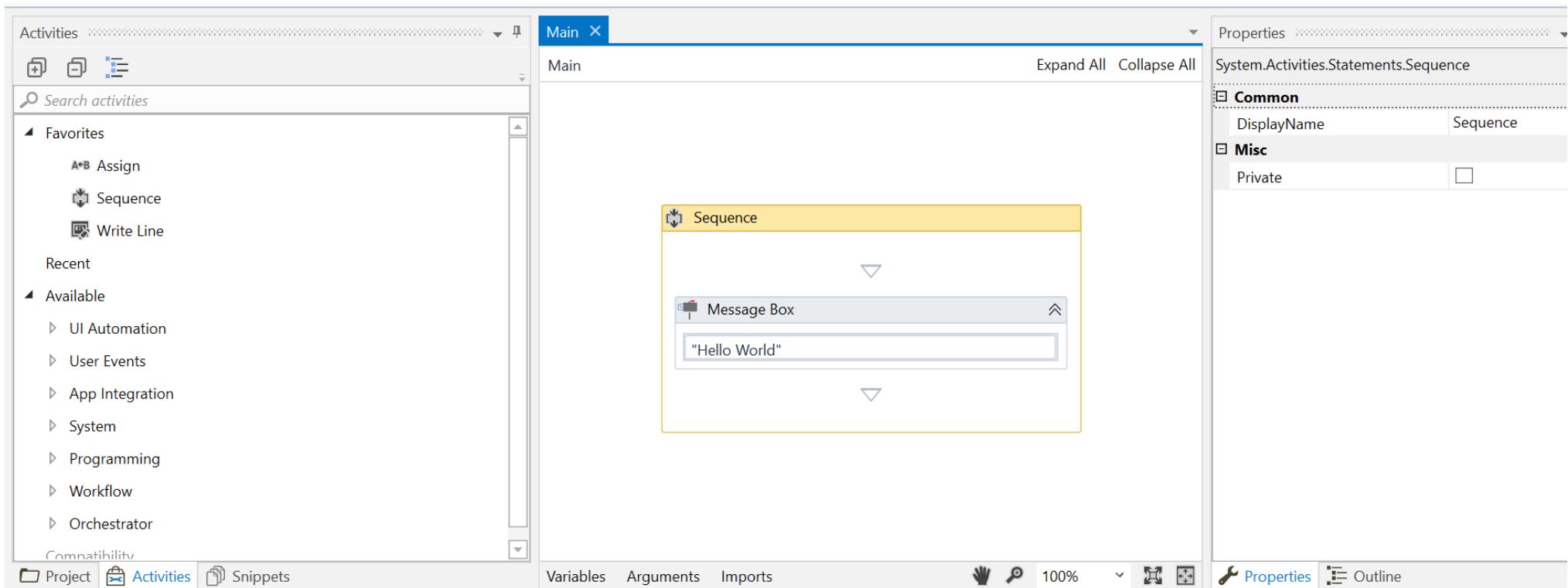
Exercise 1.0 (Step 1)

In the process “Introduction”, search for **Message Box** in the **Activities** panel and drag (or double-click) it into the **Designer Panel**.



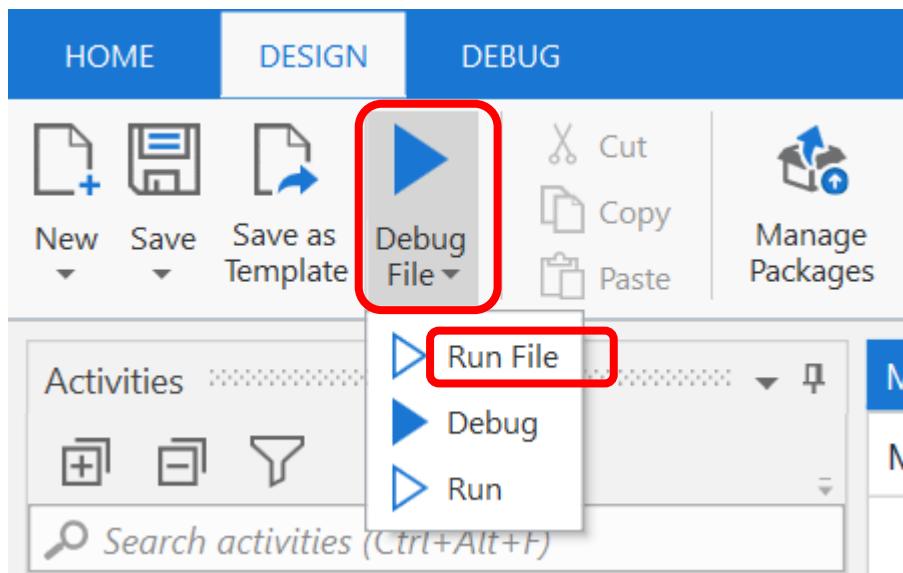
Exercise 1.0 (Step 2)

Type “Hello World!” into the content field of the **Message box** window. Remember to enclose the text with inverted commas.

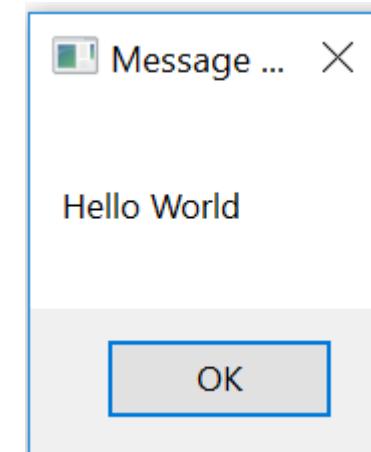


Exercise 1.0 (Step 3)

Click the **Start** button and select “Run File”



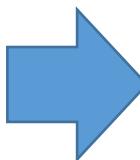
A **Message box** will appear with the message we had just input earlier





Ex 2 "Hello World" (cont'd)

5. Use the [Write Line](#) activity to display "Hello World" in the Output panel, per diagram
6. Refer to Ex2 in "[RPA UiPath Hands-On Guide](#)" for step-by-step instruction



The screenshot shows the 'Output' panel in the UiPath Studio interface. At the top, there is a header with the word 'Output' followed by a long, dashed grey line. Below the header are several status indicators: a clock icon, an exclamation mark icon with '0', an exclamation mark icon with '0', an information icon with '2', an information icon with '2', a green checkmark icon with '0', a red checkmark icon with '0', and a grid icon. To the right of these icons are two small buttons: a magnifying glass and a trash can.

Below the status indicators is a search bar labeled 'Search' with a magnifying glass icon. Underneath the search bar is a list of log entries:

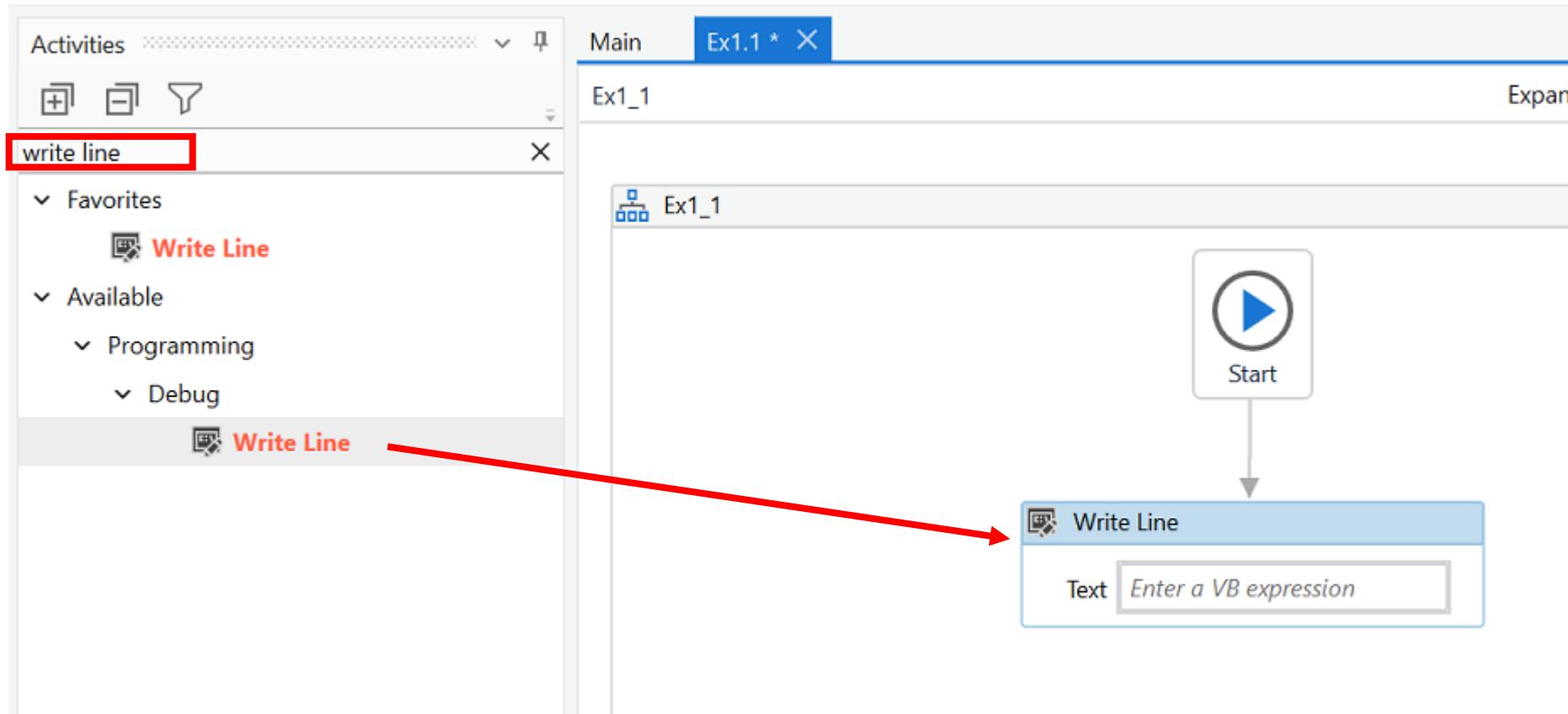
- ① Execution started for file: Main
- ① Lesson2_Albusnus execution started
- ① Hello World
- ① Lesson2_Albusnus execution ended in: 00:00:05



Exercise 2

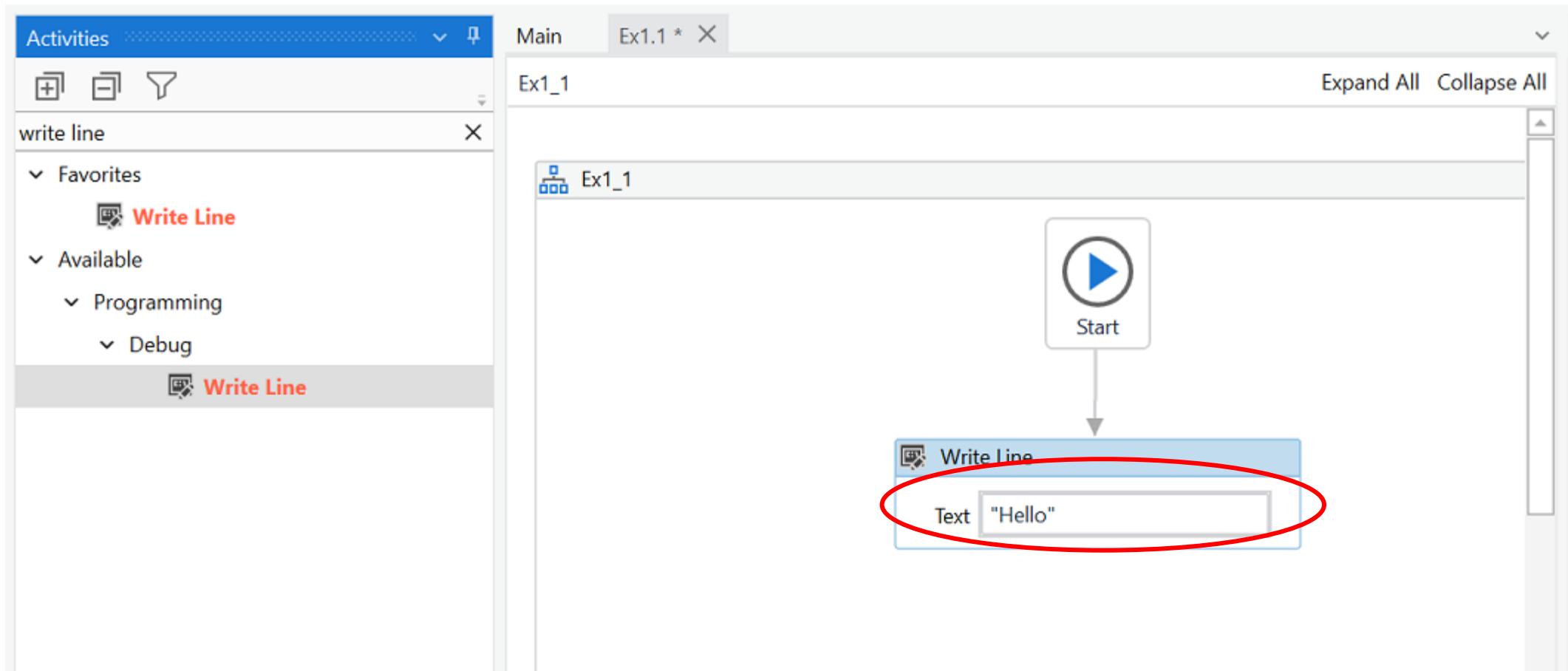
Exercise 2 (Step 1)

In the process “Introduction”, search for **Write Line** in the **Activities** panel and drag (or double-click) it into the **Designer Panel**.



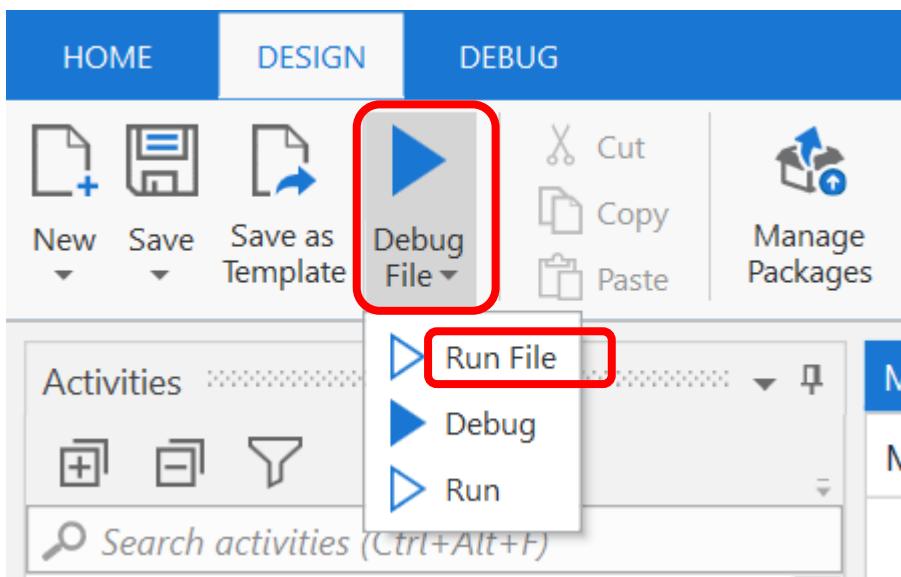
Exercise 2 (Step 2)

Type "Hello World!" into the content field of the **Write Line** window.
Remember to enclose the text with inverted commas.

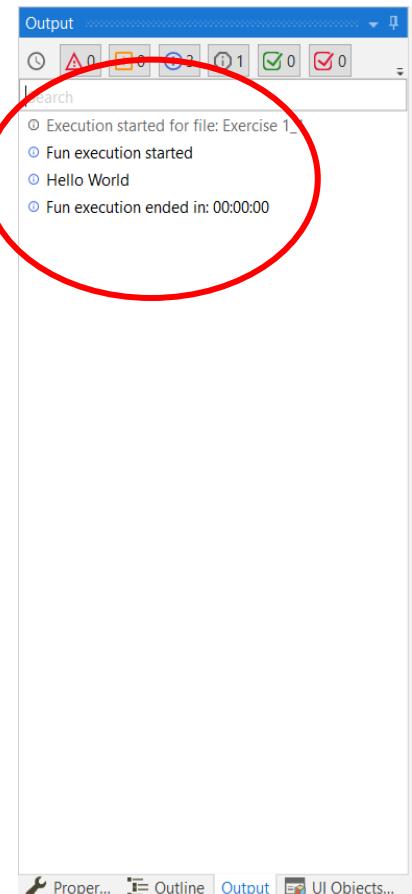


Exercise 2 (Step 3)

Click the **Start** button and select “Run File”



The message will appear under the Output



What are the differences in using
Message Box activity and Write Line activity?

Message Box Vs Write Line

Message Box

A **Message box** will appear with the message we had just input earlier

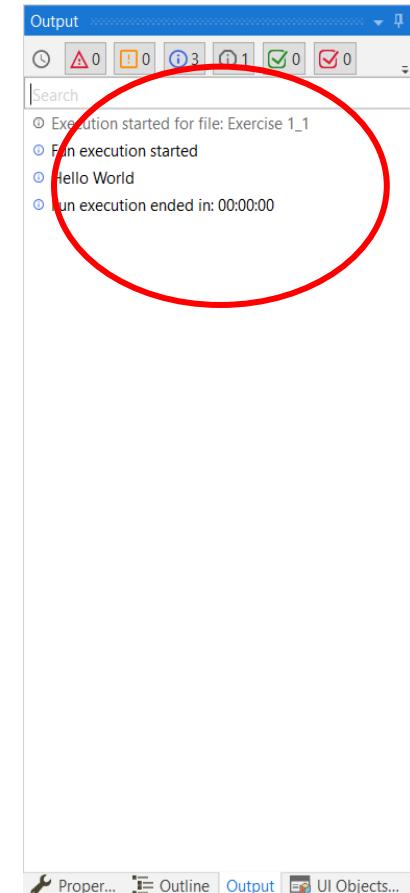
User **must click OK** for the robot to continue and complete the workflow execution



Write Line

The message will appear under the Output

User **does not need to interact** or click anything when robot executes this activity.

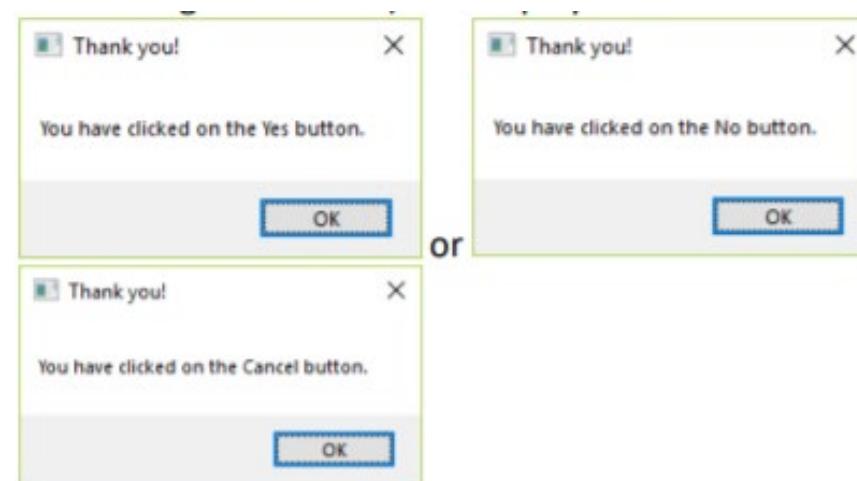
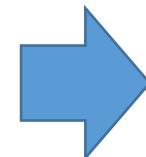




Ex 3 "Hello World" (Challenge)

1. Use the [Message Box](#) activity to display "Hello World" with "Yes", "No" and "Cancel" buttons
2. Hint: explore, use the Property Panel of the activity.

3. Use the [Message Box](#) activity to display "Thank you" with a customized message based on user's click in Step 1.
4. Hint: use **Variable**, [click](#) & watch [video](#)



Recording Function

Exercise 4

Exercise 4 – Web Recording

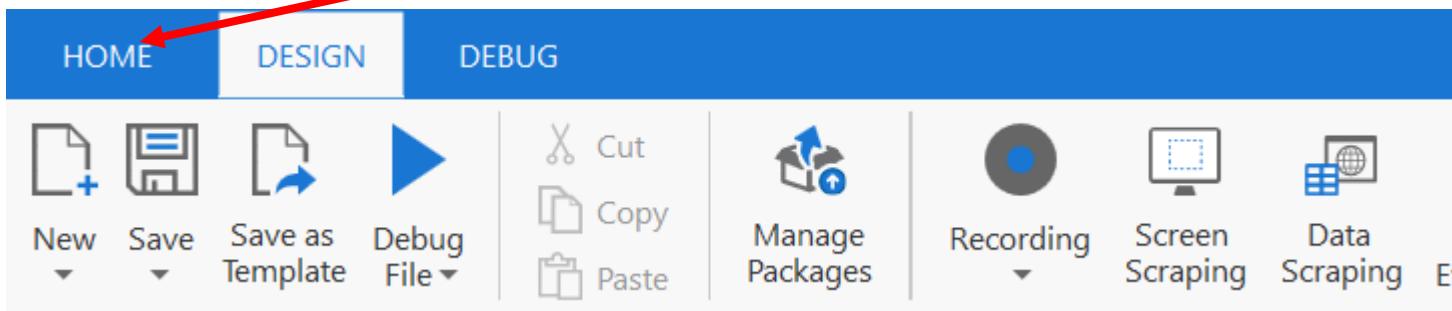
- In this exercise we will do a search for information in a web page using **Web Recording** in UiPath Studio.

Setup for Google Extension

This set-up is necessary for UiPath to interact with Google Chrome browser

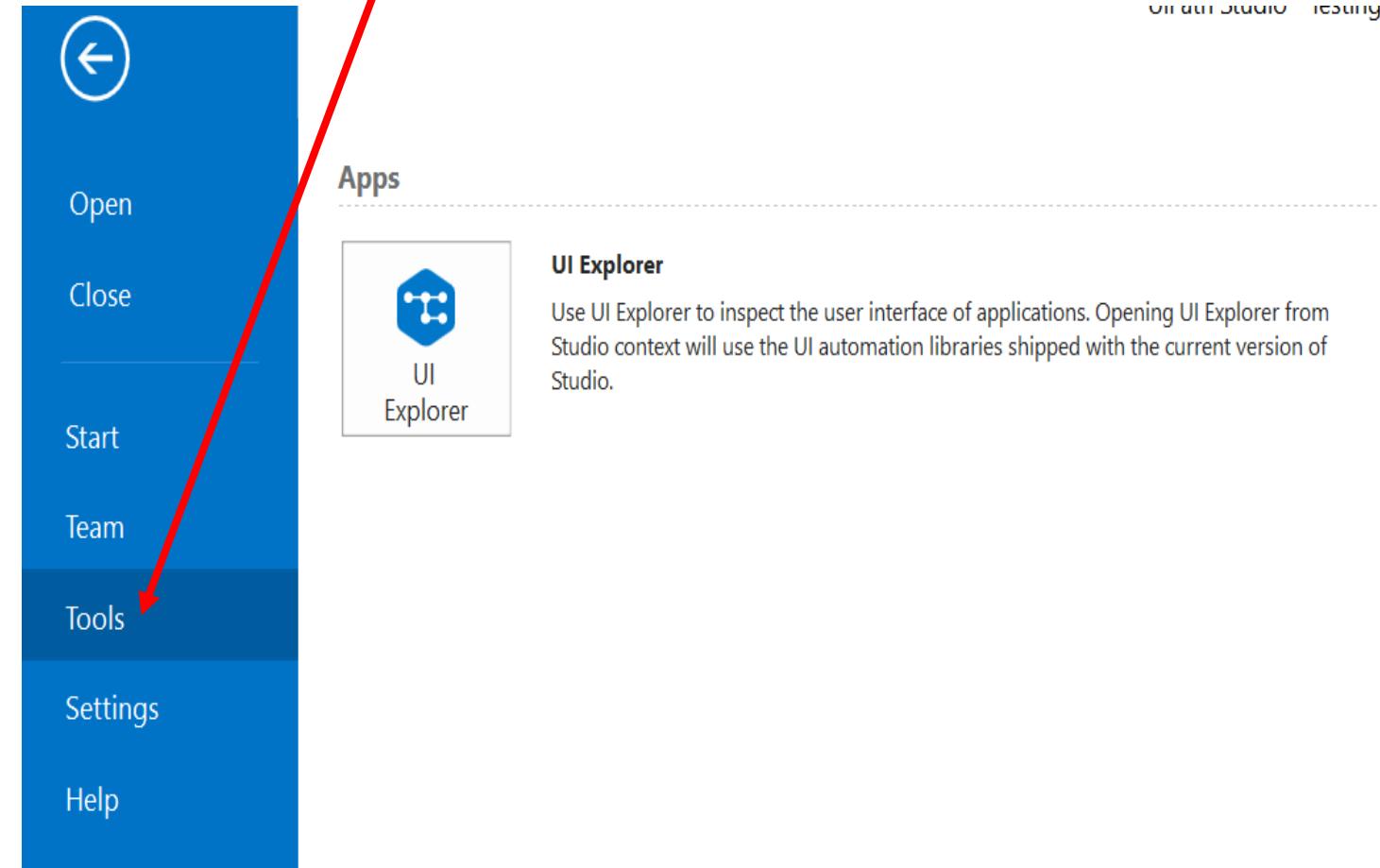
Set-Up for Google Extension

- In the Uipath Studio, click on “HOME”

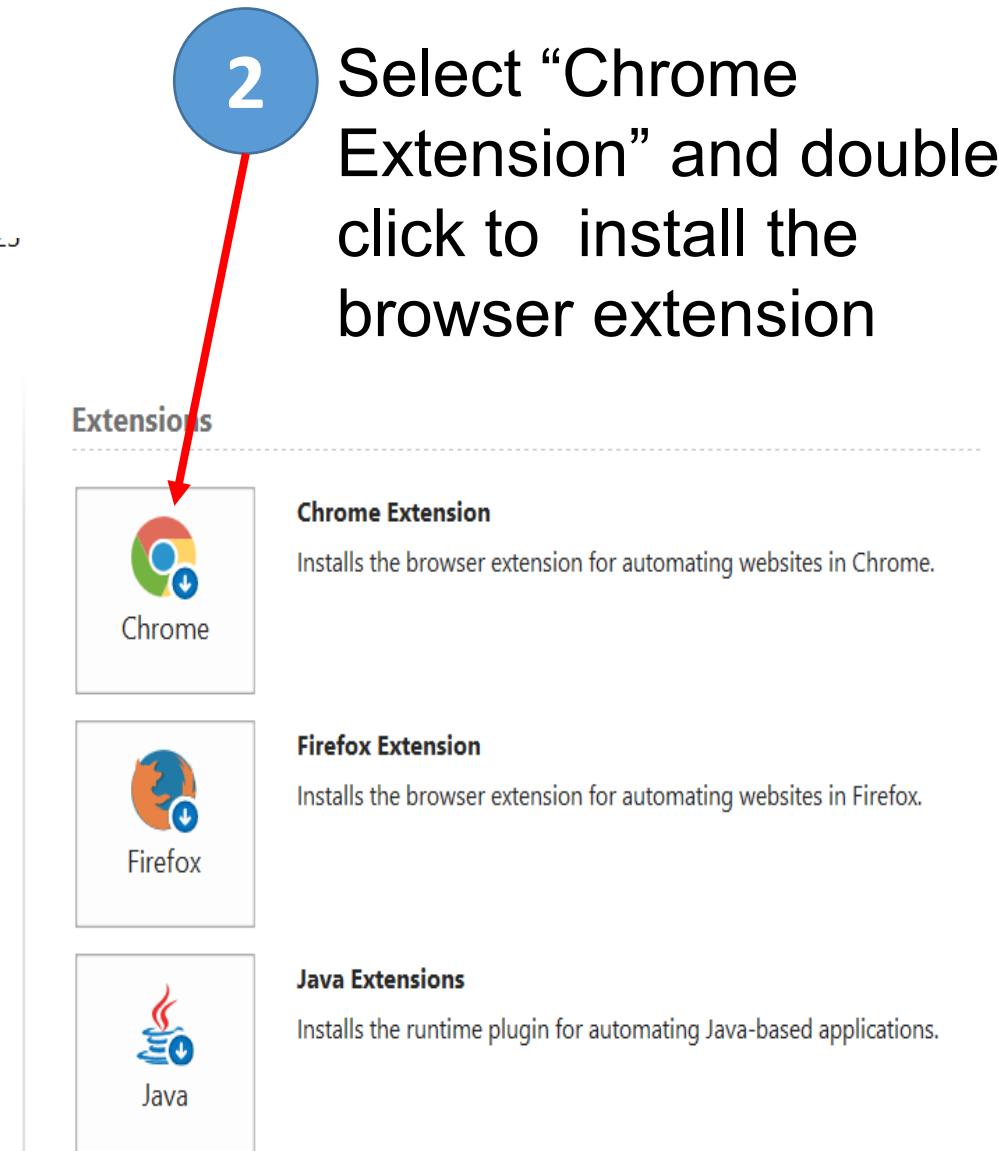


Set-Up for Google Extension

1 Select “Tools”



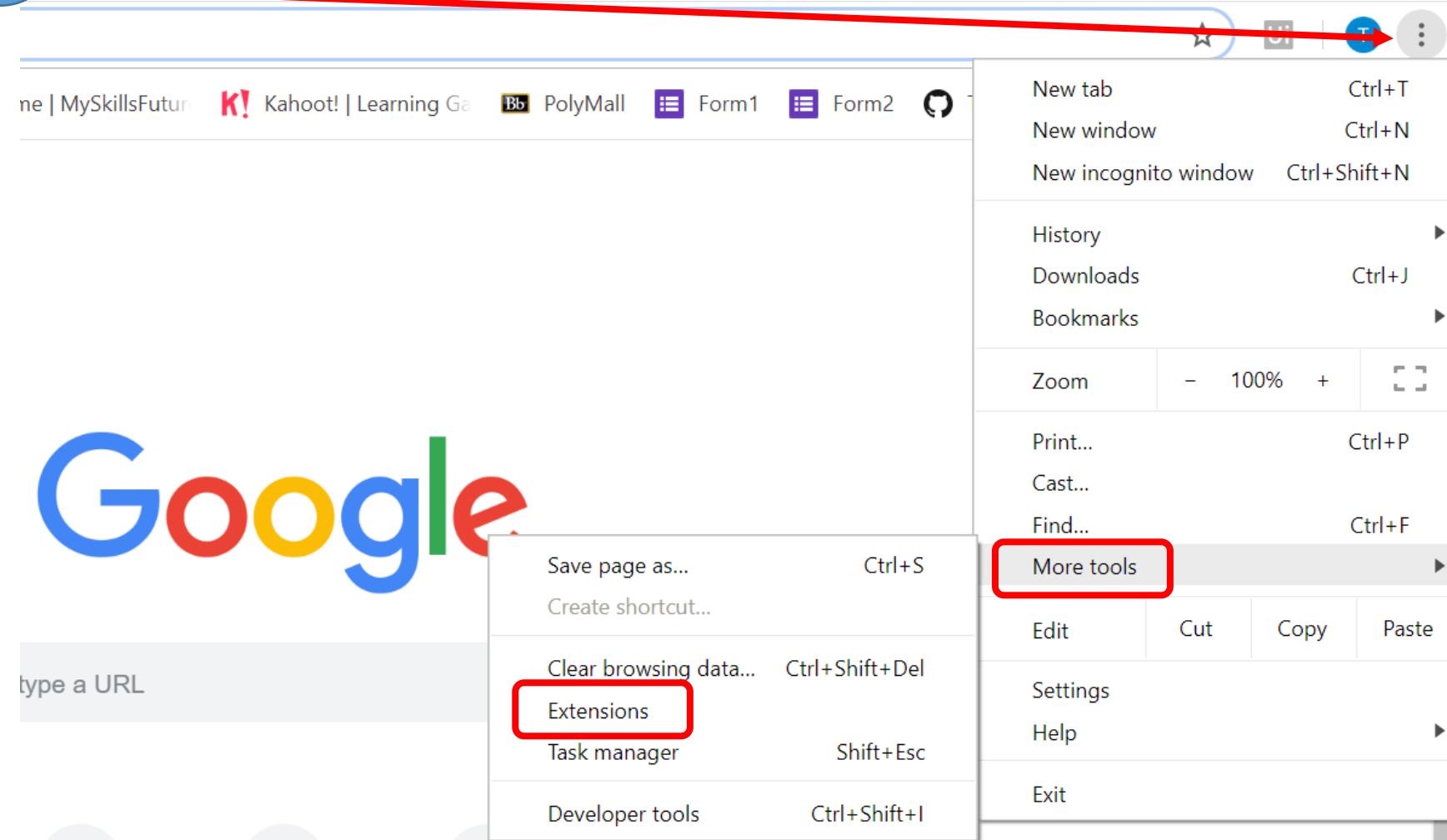
2 Select “Chrome Extension” and double click to install the browser extension



Set-Up for Google Extension (In Chrome)

1

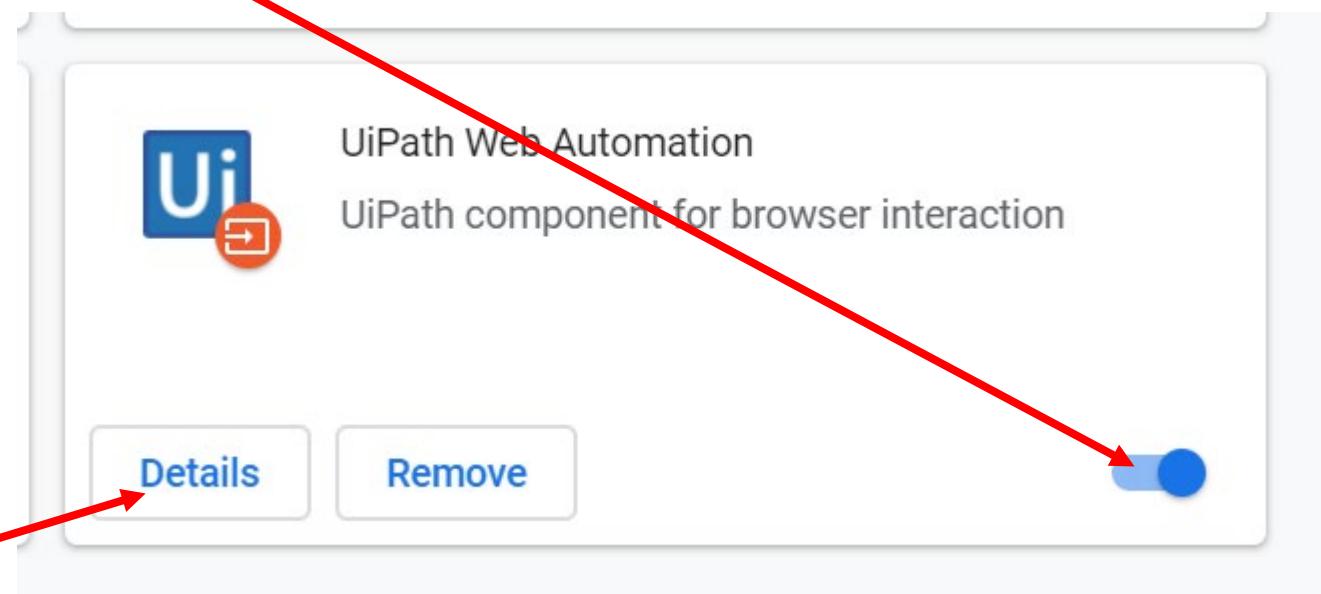
Select “...” → “More tools” → Extensions



Set-Up for Google Extension (In Chrome)

1

Make sure this is turned on.



2

Click on Details and see next slide.

Set-Up for Google Extension (In Chrome)

1

Make sure these are turned on too

Update

- Read your browsing history
- Communicate with cooperating native applications

Allow this extension to read and change all your data on websites you visit:

On all sites ▾

Learn more about [site access](#)

Allow in incognito

Warning: Google Chrome cannot prevent extensions from recording your browsing history. To disable this extension in incognito mode, unselect this option.



Allow access to file URLs

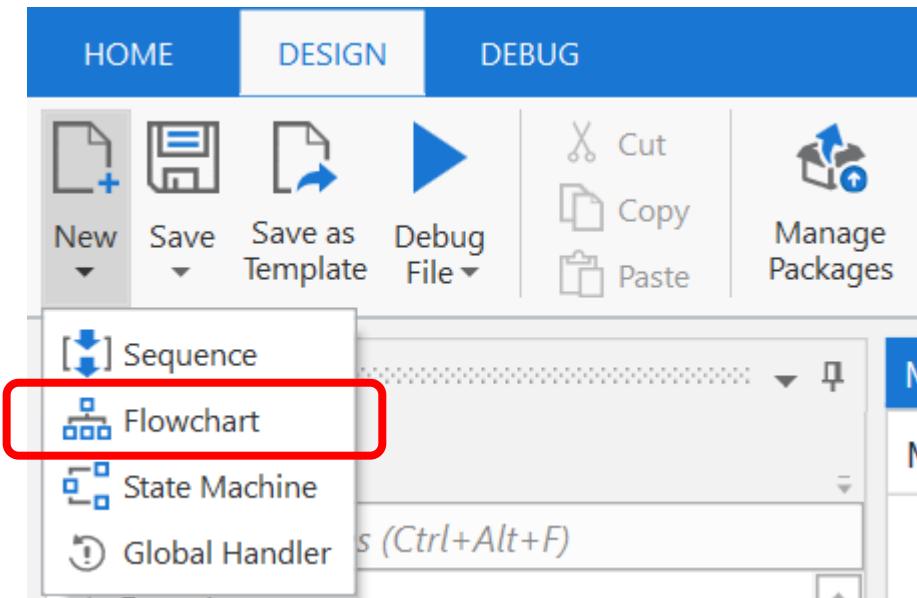


Collect errors



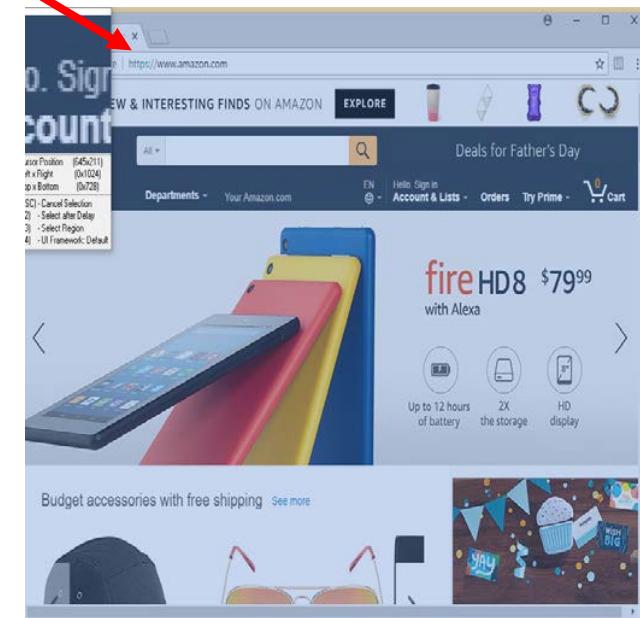
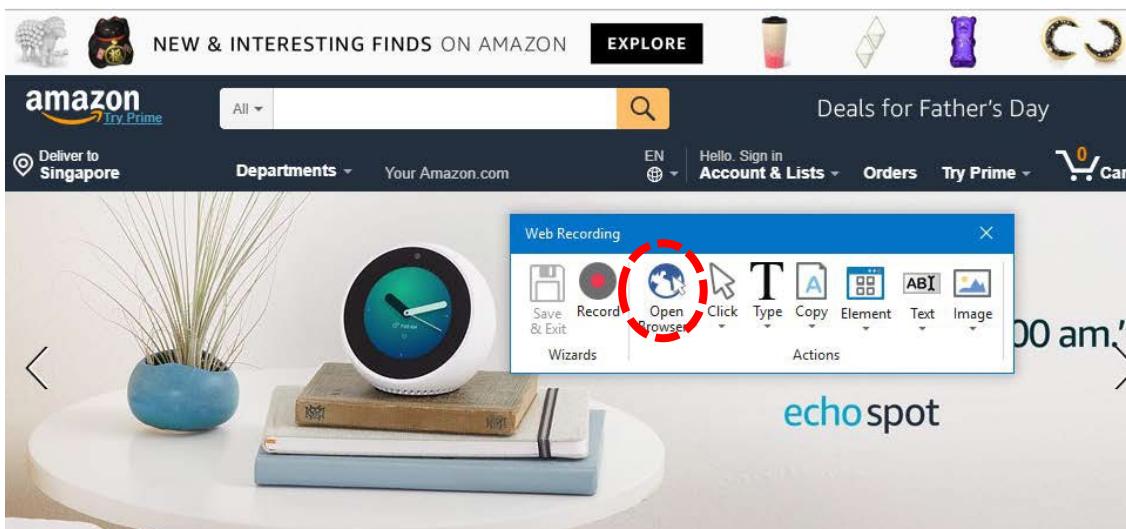
Exercise 4 (Step 1)

In the UiPath Studio, create a new flowchart, and name it “Ex4”



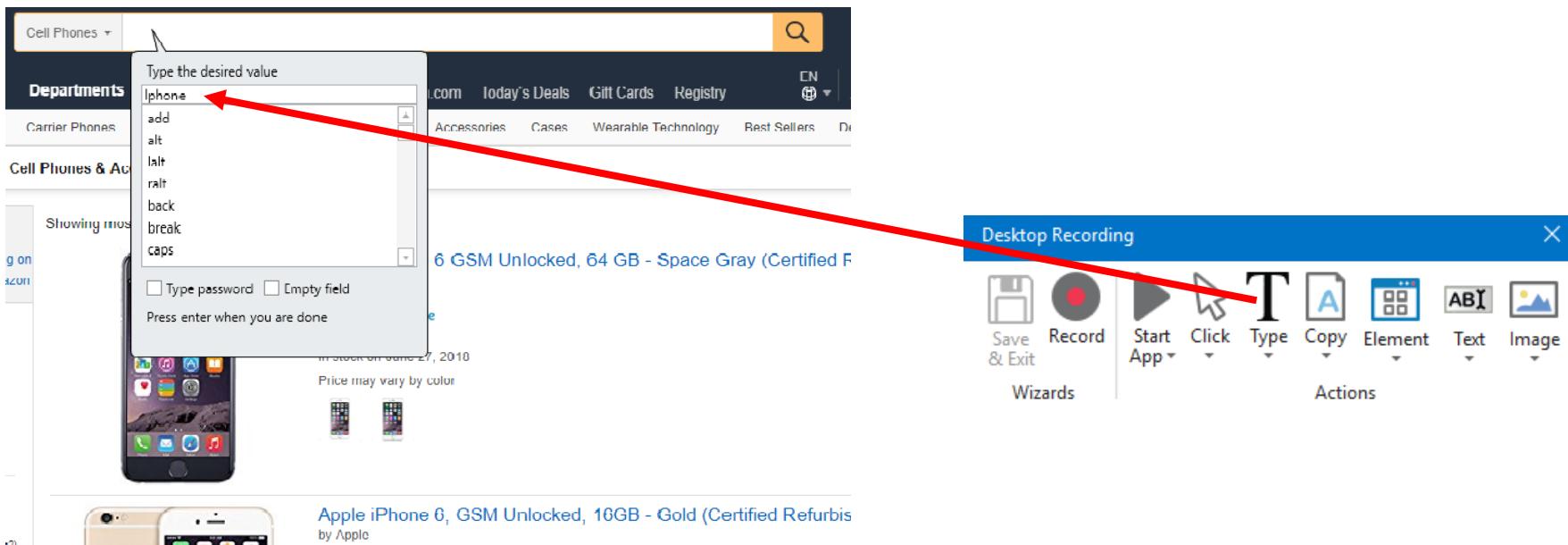
Exercise 4 (Step 2)

1. Launch Google Chrome and navigate to www.amazon.com
2. Click on **Recording** → **Web** → **Open Browser**.
3. Then, click on the centre of the web page. (The URL should be shown as www.amazon.com)



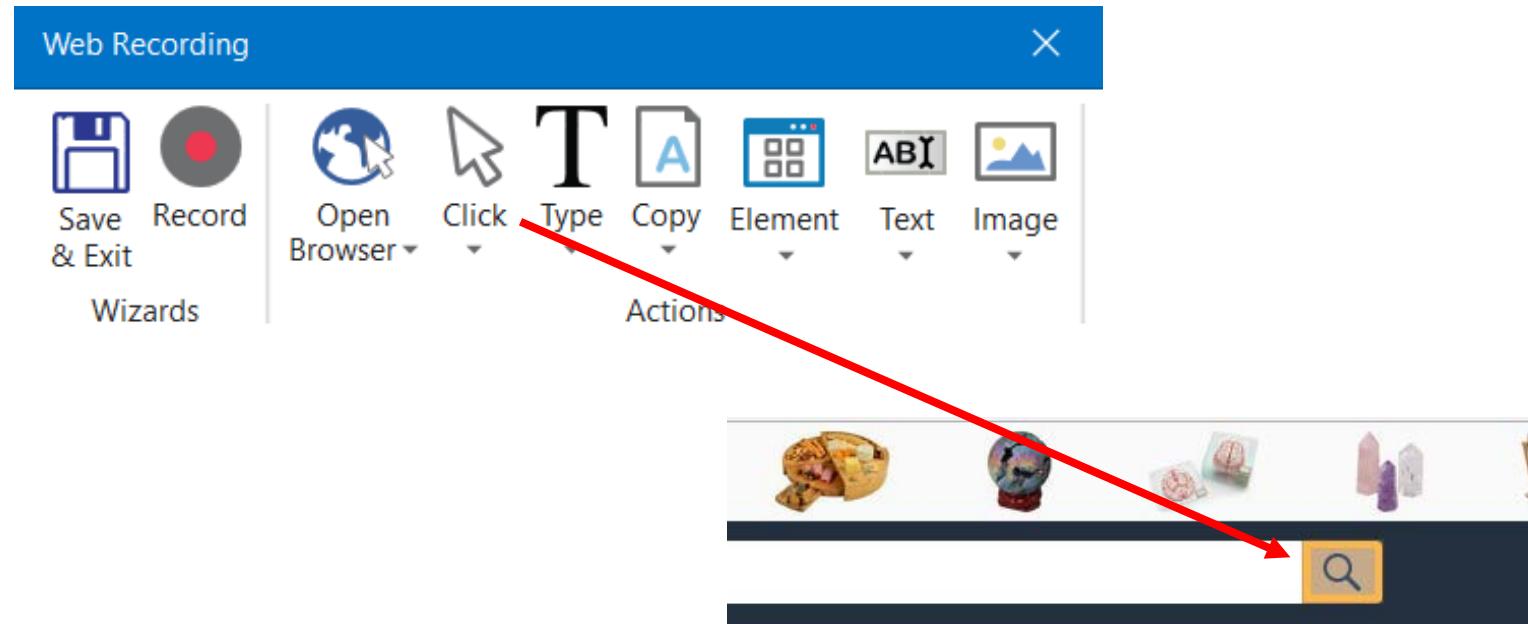
Exercise 4 (Step 3)

- Select **Type** from toolbar
- Click on the Search bar and key-in “iPhone”



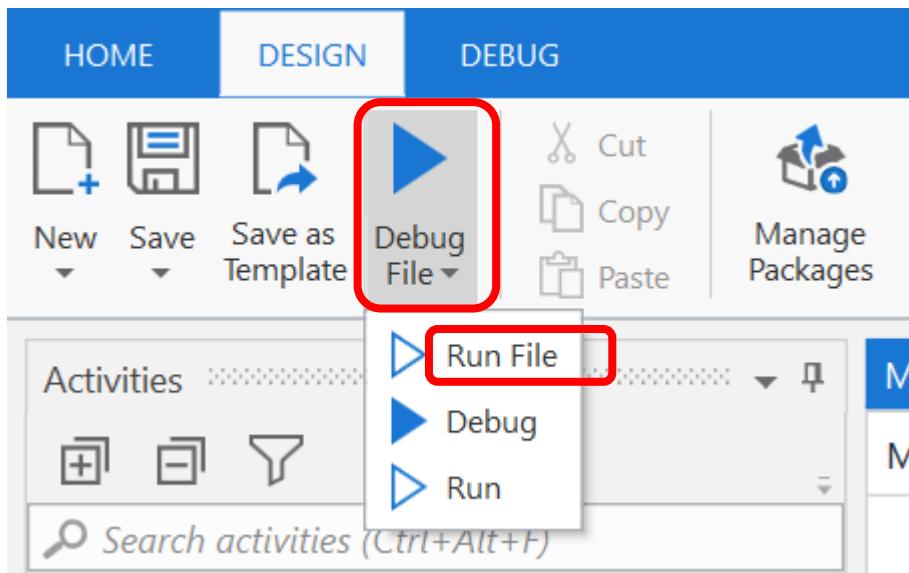
Exercise 4 (Step 4)

- Select **Click** from toolbar
- Click on the Search button, ie the magnifying glass
- Click “Save & Exit”



Exercise 4 (Step 5)

Click the **Debug File** button and select “Run File”
- You should see the whole search action working



Review

- **Web Recording** is designed for recording in web apps and browsers (supported: Internet Explorer, Google Chrome), generates containers and uses the **Simulate Type/Click** input method by default.



Ex5.1 Google Form Fill

1. In UiPath Studio, create a new Process and name it “*Robot2_XXXXXX*” with description as “*RPA Lesson3-4 Ex5.1 to Ex5.6*”.
XXXXXX is your first name, e.g. *Jun_Hao* or *Jeremy* etc
2. Refer to Ex5.1 in “RPA UiPath Hands-On Guide” for step-by-step instruction



Exercise 5.1 – Google Form Fill

RPA Form 1
RPA Training Form 1

Name **1**
Your answer

Email **2**
Your answer

Job Title **3**
Your answer

Phone Number **4**
Your answer

Submit

P14	A	B	C	D
1	Name	Email	Job Title	Phone number
2	Alvin Ng	Alvin@gmail.com	Lecturer	63216781
3	Bee Leng	Beeleng@gmail.com	Finance Manager	90001892
4	David	david@yahoo.com	Business Manager	56777777
5	Jason Chong	jason@yahoo.com	Marketing Manager	33888999
6	Kok Leong	kokleong@gmail.com	IT Manager	22266677
7				

Google Form Fill

Exercise 5.1

Exercise 5.1 – Setup

- What you need:
 - Excel template, “Exercise 5”
 - Google form:

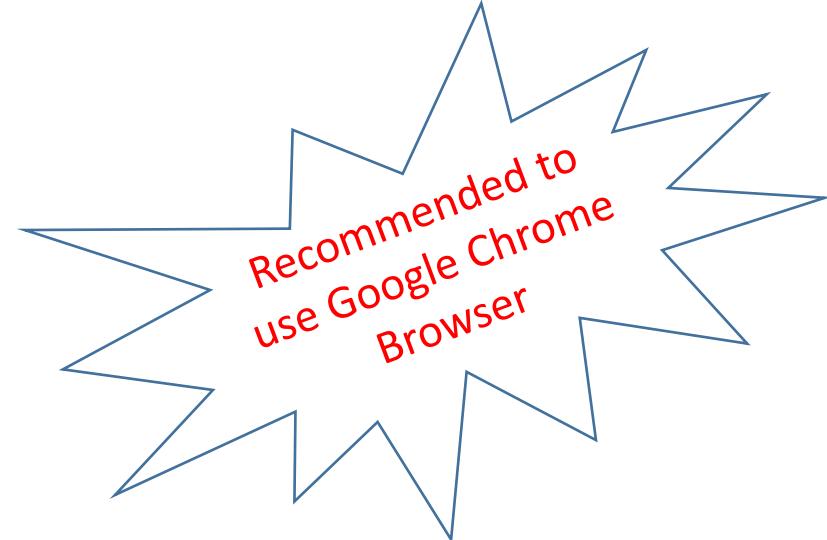
[Form 1](#) (for Ex 5.1, 5.2, 5.3)

https://docs.google.com/forms/d/e/1FAIpQLSdkAsTq117_5lcE02VjoBmhPrWlaWpANnyEDOivnNqnBBymzQ/viewform

[Form 2](#) (for Ex 5.4, 5.5 and 5.6)

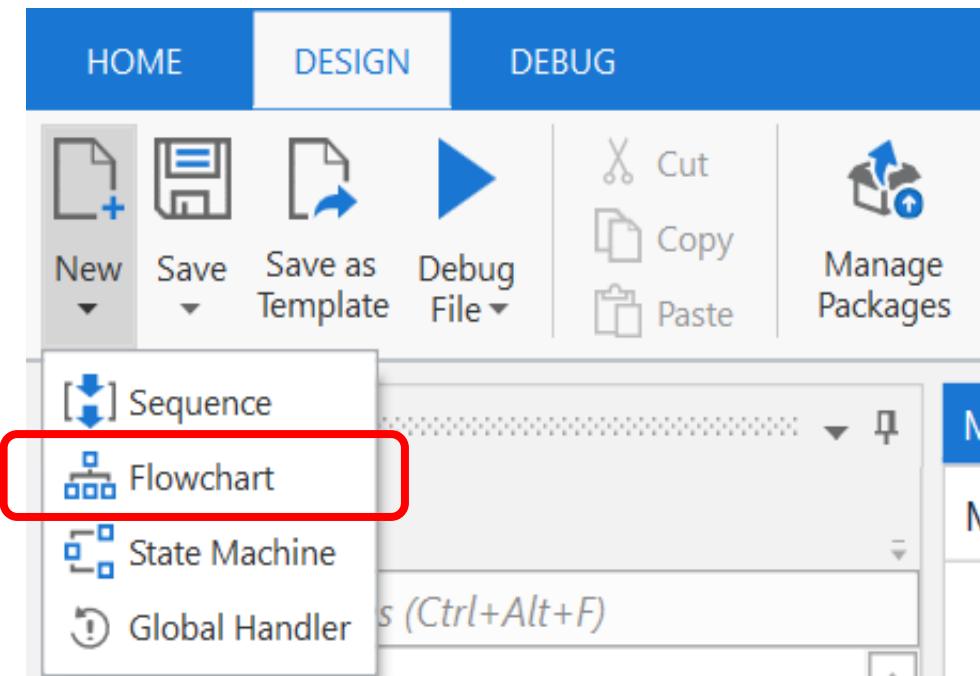
<https://docs.google.com/forms/d/e/1FAIpQLScQEgTYEU9linawWqeHnz6fXMFkzmF5KMCQMSJFAounx262hig/viewform>

- Others:
 - make sure the Google Form is opened (preferably google chrome)
 - Open the excel template, “Exercise 5” and set the cursor at cell “A2”



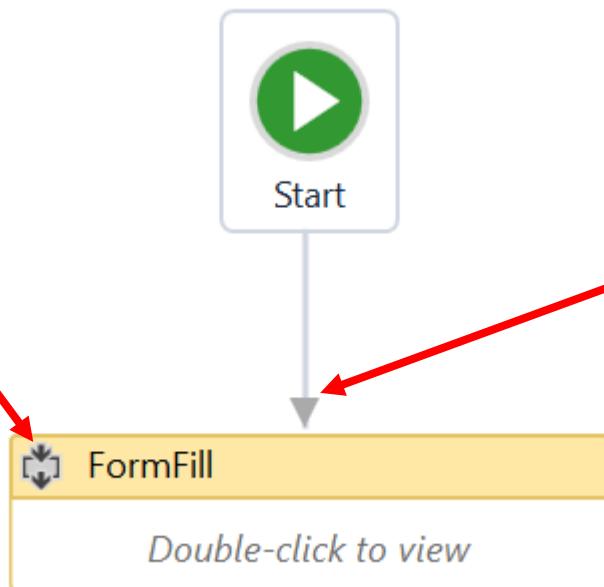
Exercise 5.1 (Step 1)

In the UiPath Studio, create a new flowchart by click on **New → Flowchart** and name it “Exercise 5.1”



Exercise 5.1 (Step 2)

- Insert a “Sequence” and change the header from “Sequence to “FormFill”

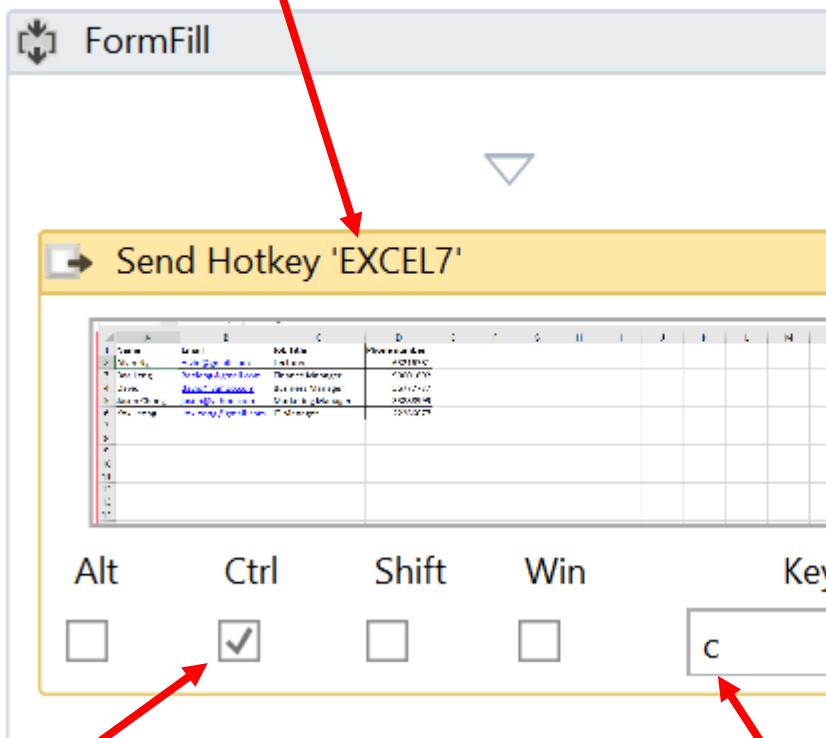


- Right click on the sequence and select “Set as start node” to join the start button with the sequence, “Formfill”

Exercise 5.1 (Step 3)

1

Insert a “Send Hotkey”



2

Click and select “Indicate on screen” and select the excel template

4

Enter “c”

3

Select “Ctrl”

- Indicate on screen
- Edit Selector
- Change Informative
- Remove Informative
- Show Informative S

Exercise 5.1 (Step 4)

1 Insert a “Send Hotkey”

2 Click and select “Indicate on screen” and click on the blank field under “Name” in google form

3 Select “Ctrl”

4 Enter “v”

The screenshot shows the Google Forms interface. At the top, there's a toolbar with 'File', 'FormFill', and 'Exp...' options. Below the toolbar, there are two 'Send Hotkey' triggers. The first trigger has 'Ctrl' checked in its modifier section and 'c' in its key section. The second trigger also has 'Ctrl' checked and 'v' in its key section. Red arrows from the numbered steps point to these specific fields. To the right of the triggers, a Google Form is displayed with a question labeled 'Name *'. A red arrow from step 2 points to the 'Name' field. Another red arrow from step 4 points to the letter 'v' in the 'Your answer' field.

Exercise 5.1 (Step 5)

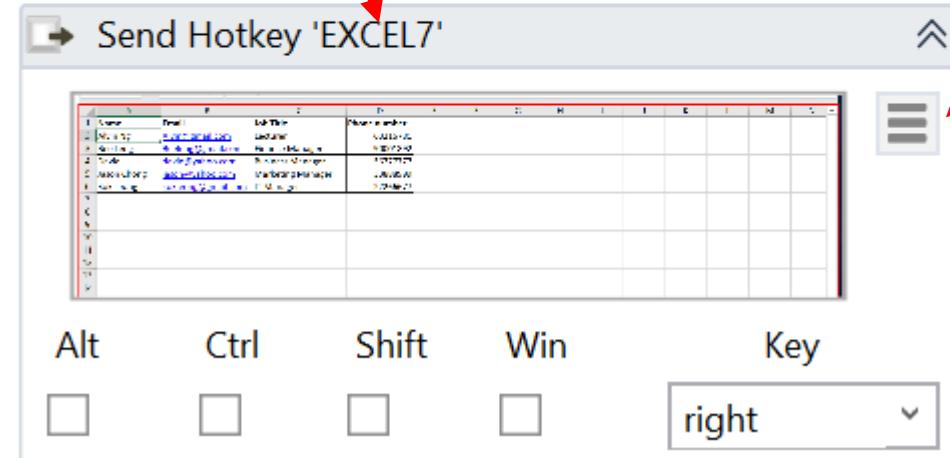
1

Insert a “Send Hotkey”



2

Click and select “Indicate on screen” and click on the excel template



3

Select “right”

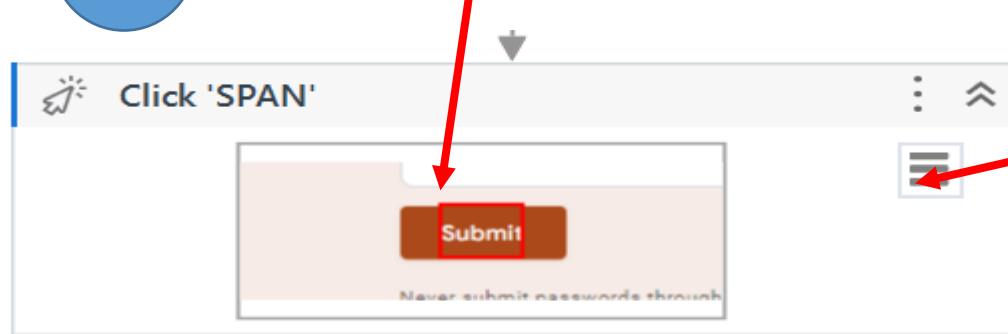
Exercise 5.1 (Step 6)

Repeat Step 3 to 5 until you have completed filling in the form.

Exercise 5.1 (Step 7)

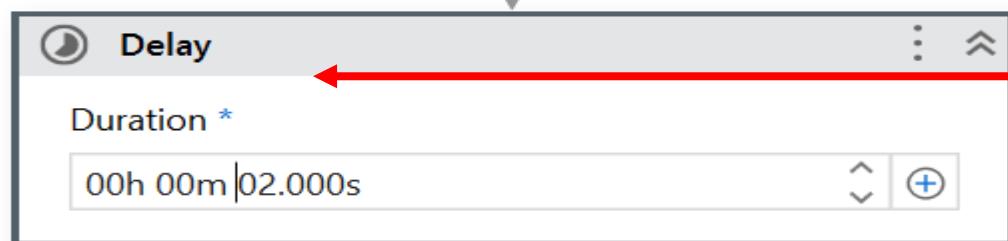
1

Insert a “Click”



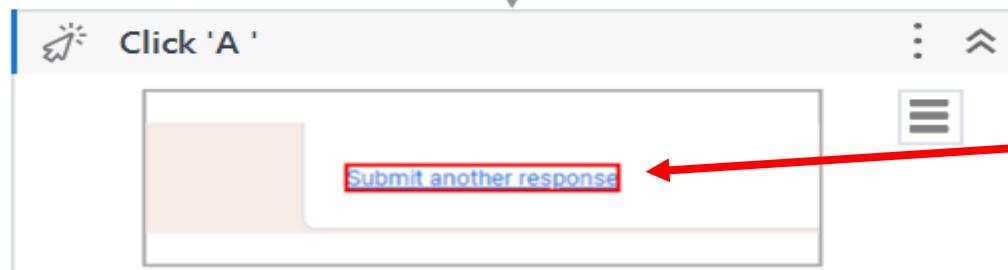
2

Click and select “Indicate on screen” and click on the “submit” button in the google form



3

Insert a “Delay” of 2 seconds

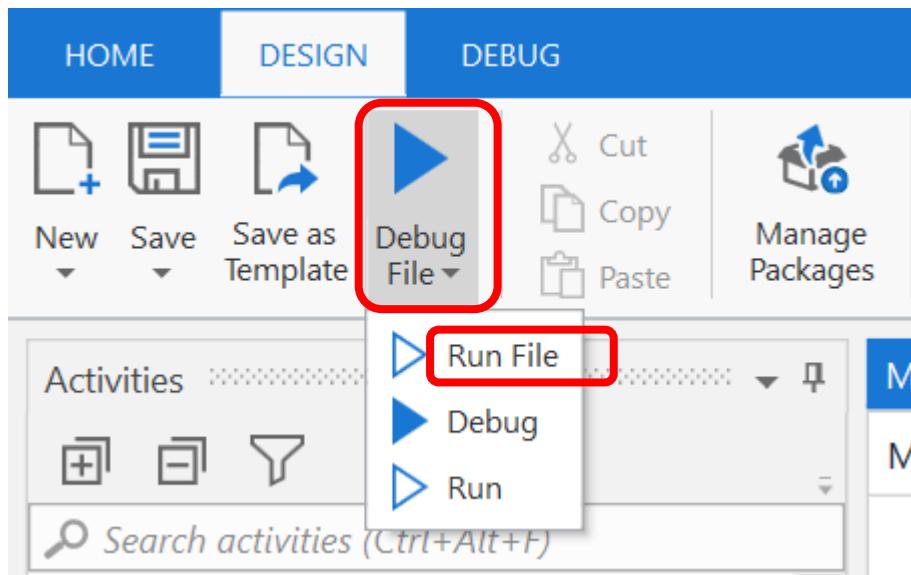


4

Repeat the above steps for this to allow submission of another record.

Exercise 5.1 (Step 8)

Click the **Debug File** button and select “Run File”



Exercise 5.1 – Review

How is creating scripts **manually to automate data entry?**

How would you **repeat data entry for the other records?**



Ex5.2 Google Form Fill

1. Use the created “*Robot2_XXXXXX*” process created in the previous lesson.
Create a new Workflow and name it Ex5.2
2. Script Robot to fill in and submit a Google Form from
an Excel table using [Variables](#)
3. Refer to Ex5.2 in “RPA UiPath Hands-On Guide” for
step-by-step instruction



RPA Form 1
RPA Training Form 1

Name
Your answer

Email
Your answer

Job Title
Your answer

Phone Number
Your answer

Submit

Google Form Fill

Exercise 5.2

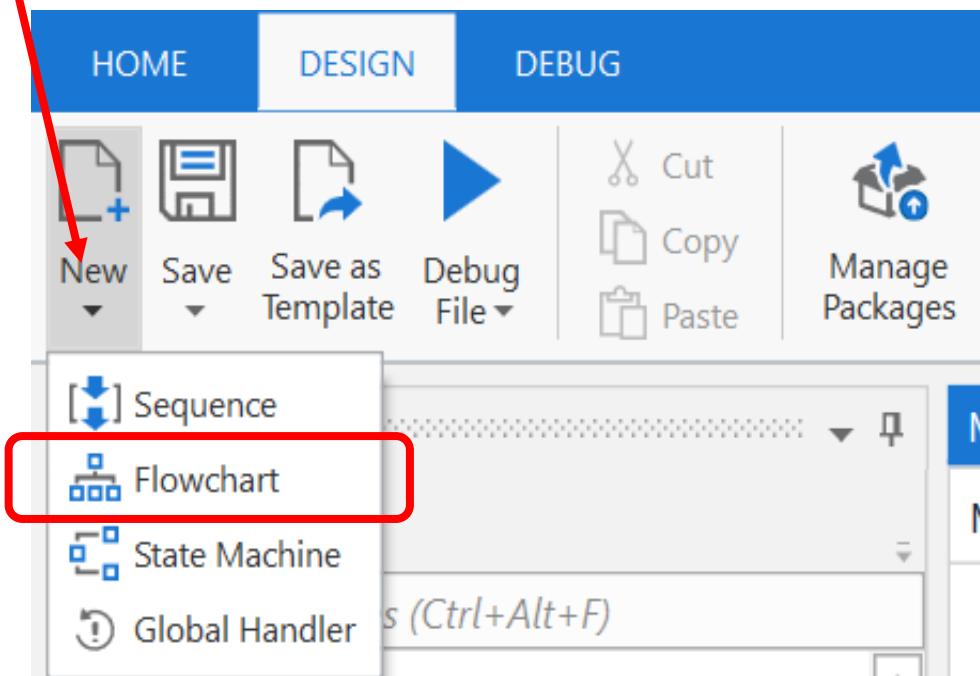
Exercise 5.2 – Excel and Data Entry using Variable

- In this exercise we will make use of the **variable** function to execute Exercise 5.1
- **Variable** is used to store data temporarily.

Exercise 5.2 (Step 1)

1

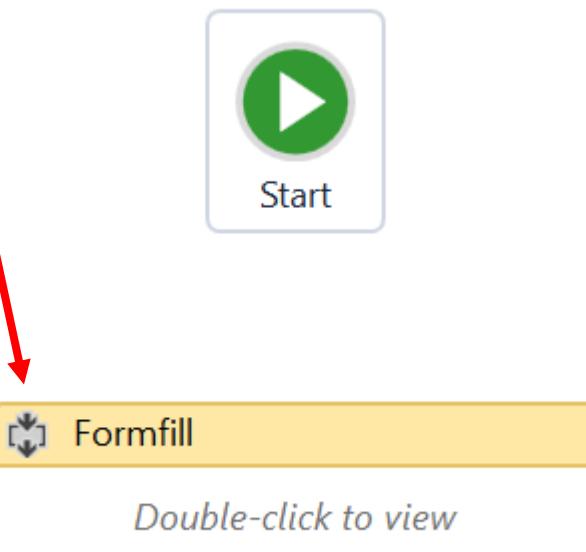
Click on **New** and select “Flowchart” and name it “Exercise 5.2”



Exercise 5.2 (Step 2)

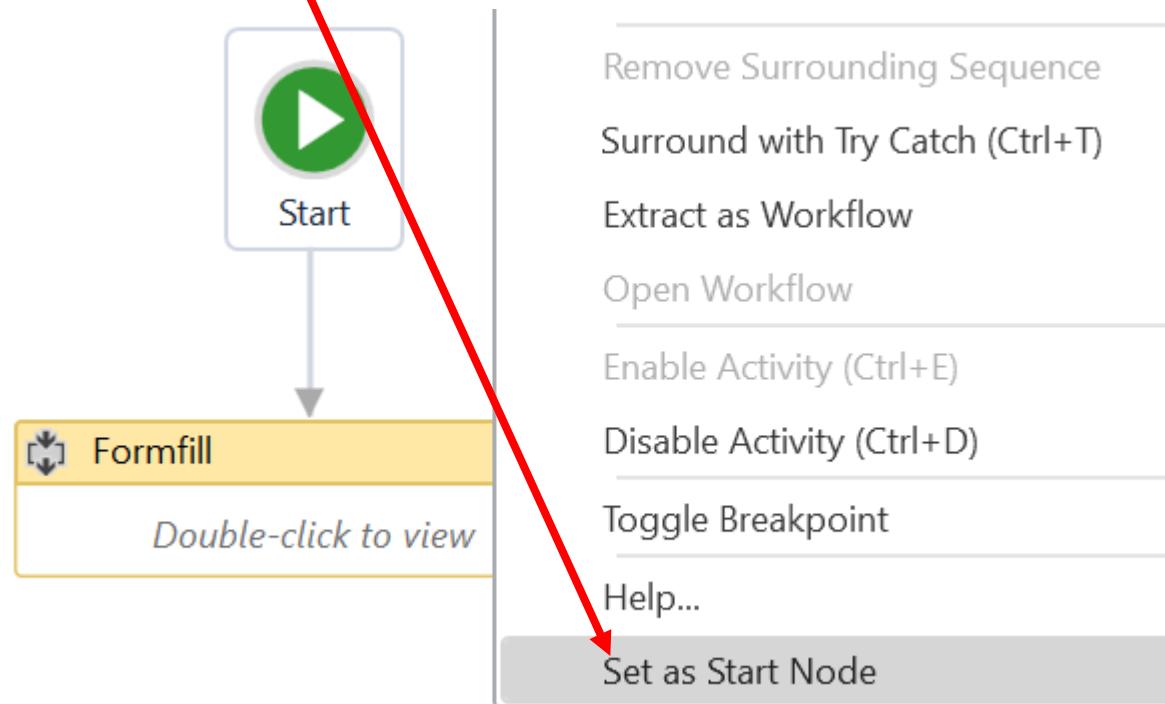
1

Insert a “Sequence” and change the header name to “FormFill”



2

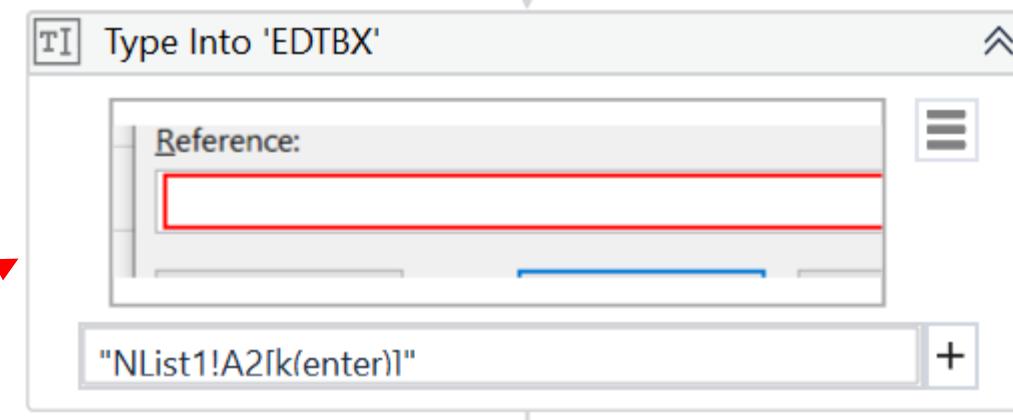
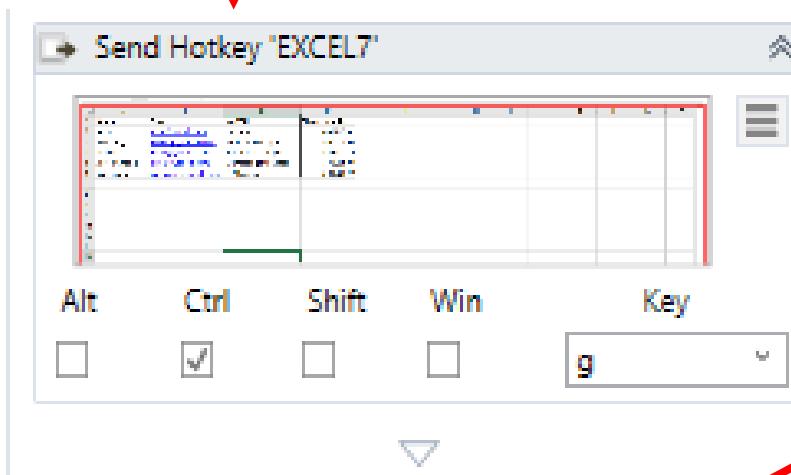
Right click on the sequence and select “Set as Start Node” to join the start button with the sequence, “Formfill”



Exercise 5.2 (Step 3)

1

Insert a “Send Hotkey” and register “Ctrl + g” for the excel screen
“Ctrl + g” is Go to Function in Excel



2

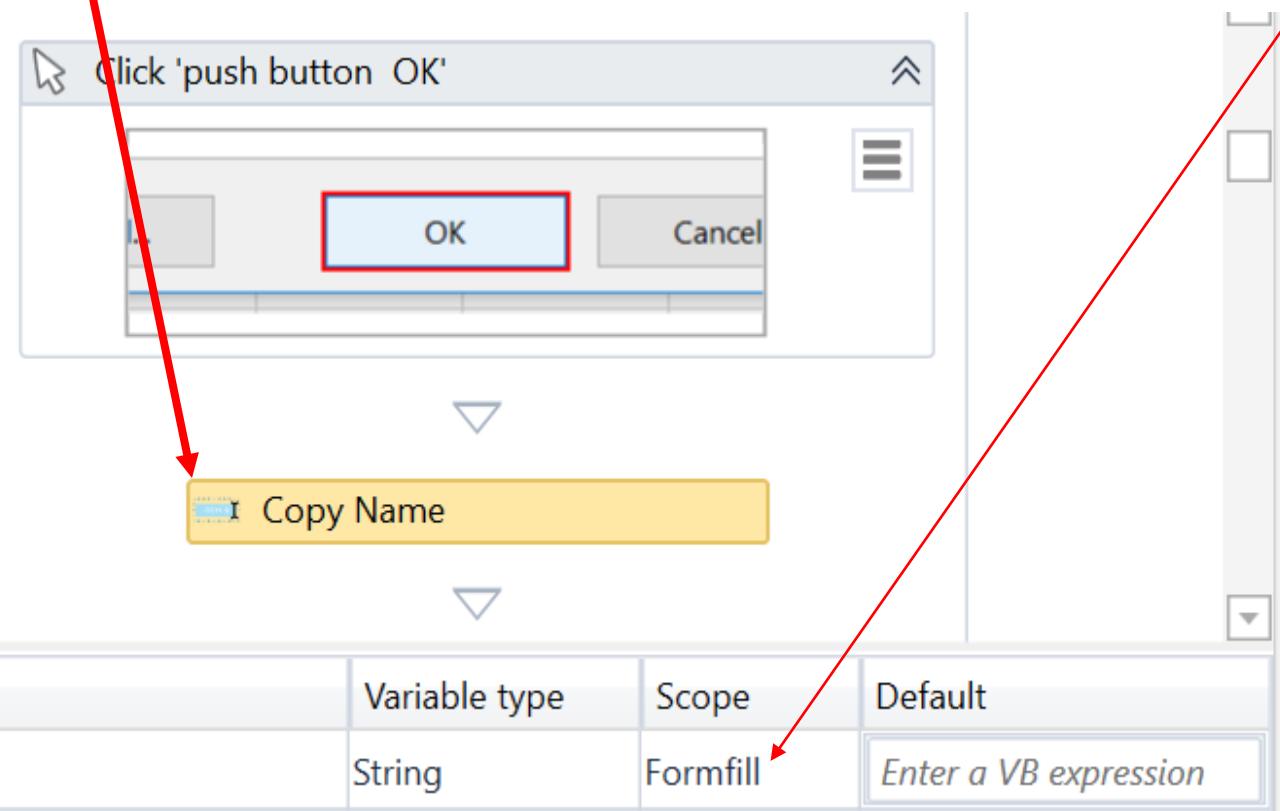
Insert a “Type Into”, capture the screen to enter the cell reference and enter text “NList1!A2”, click on + and select Enter

Exercise 5.2 (Step 4)

1

Insert a “Copy Selected Text” and rename it to “Copy Name”

NB: The Scope of the variable refers to the area/level in which a variable is available, such as a specific activity.

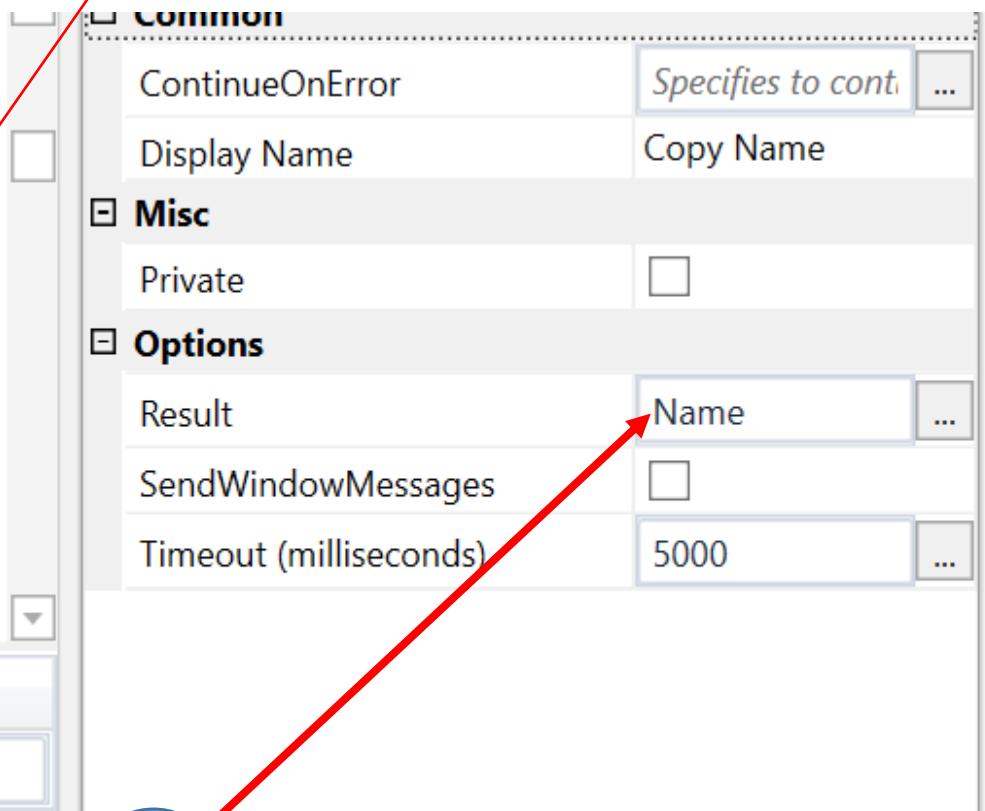


2

Click on create variable and create a variable “Name” with Variable type “String”

3

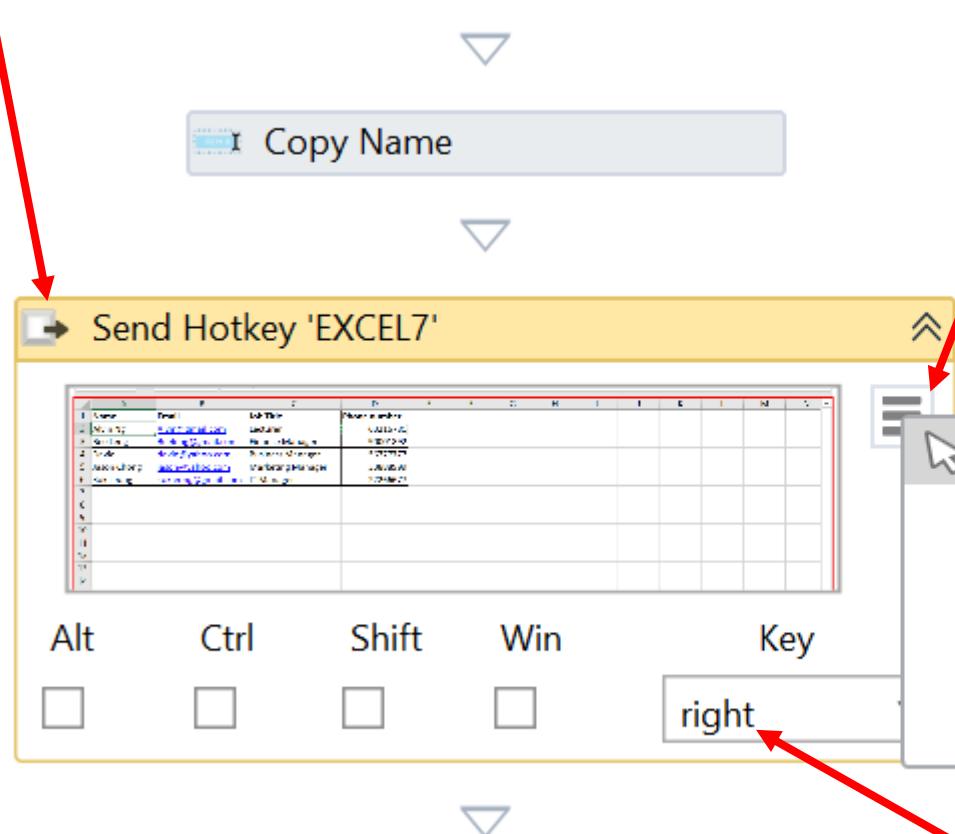
Type “N or n” and select the variable “Name”



Exercise 5.2 (Step 5)

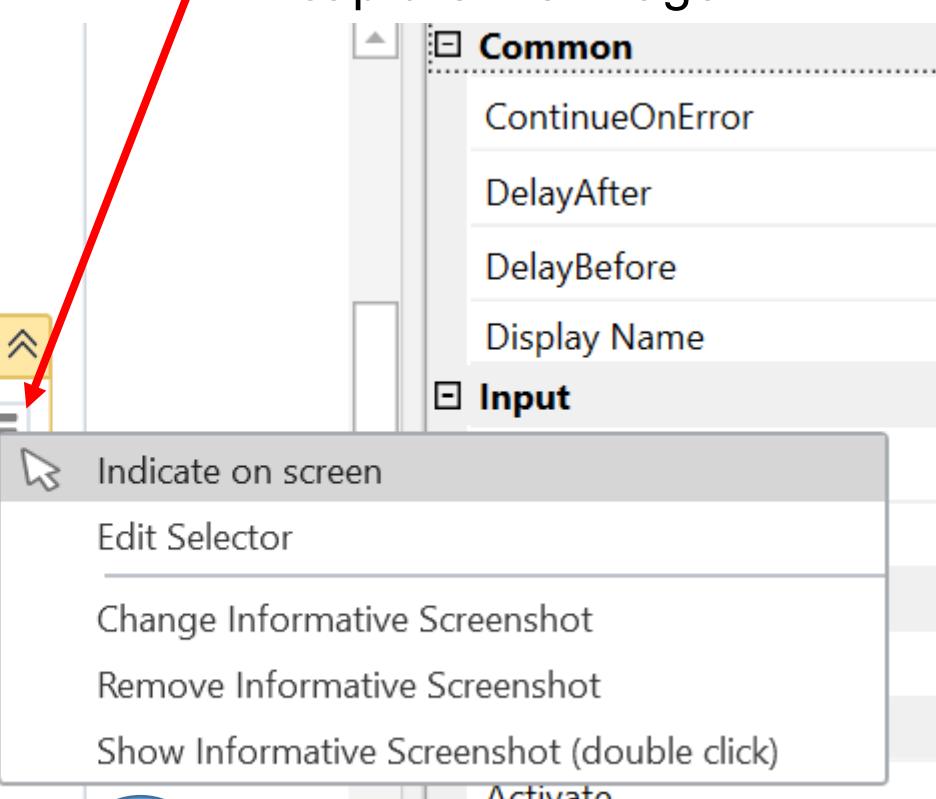
1

Insert a “Send Hotkey”



2

Click here and select “Indicate on screen” and click on the excel file to capture the image



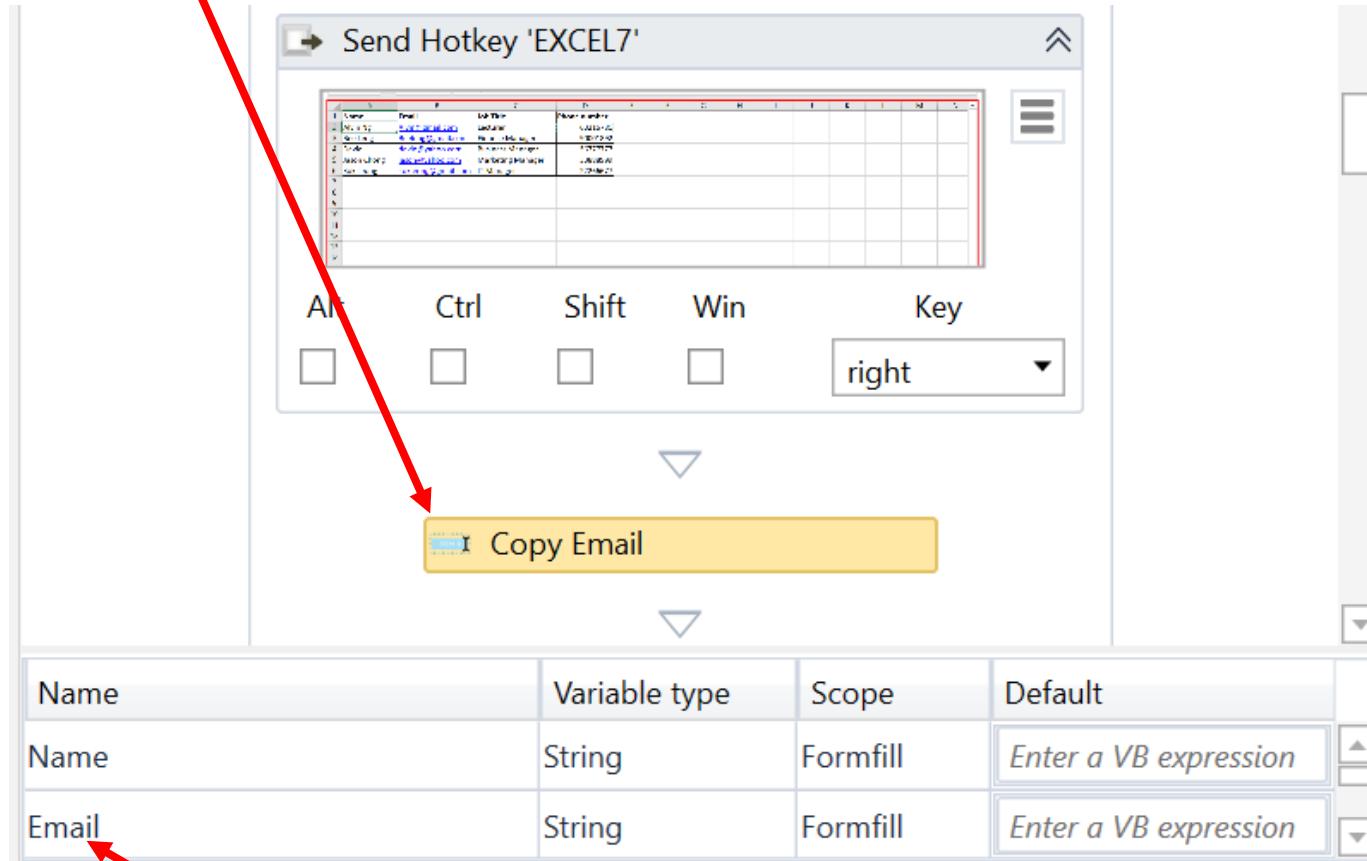
3

Select the “Right” Key to move to the cell on the right in excel

Exercise 5.2 (Step 6)

1

Insert a “Copy Selected Text” and name it “Copy Email”

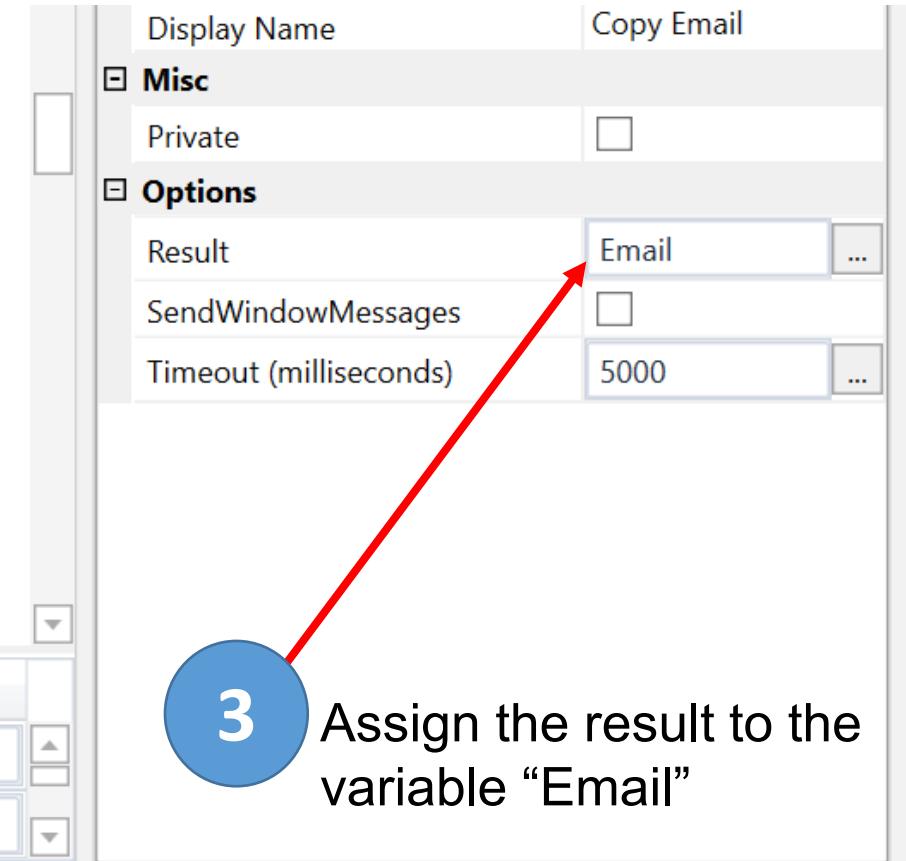


2

Click on create variable and create a variable “Email” of type ‘String’

3

Assign the result to the variable “Email”



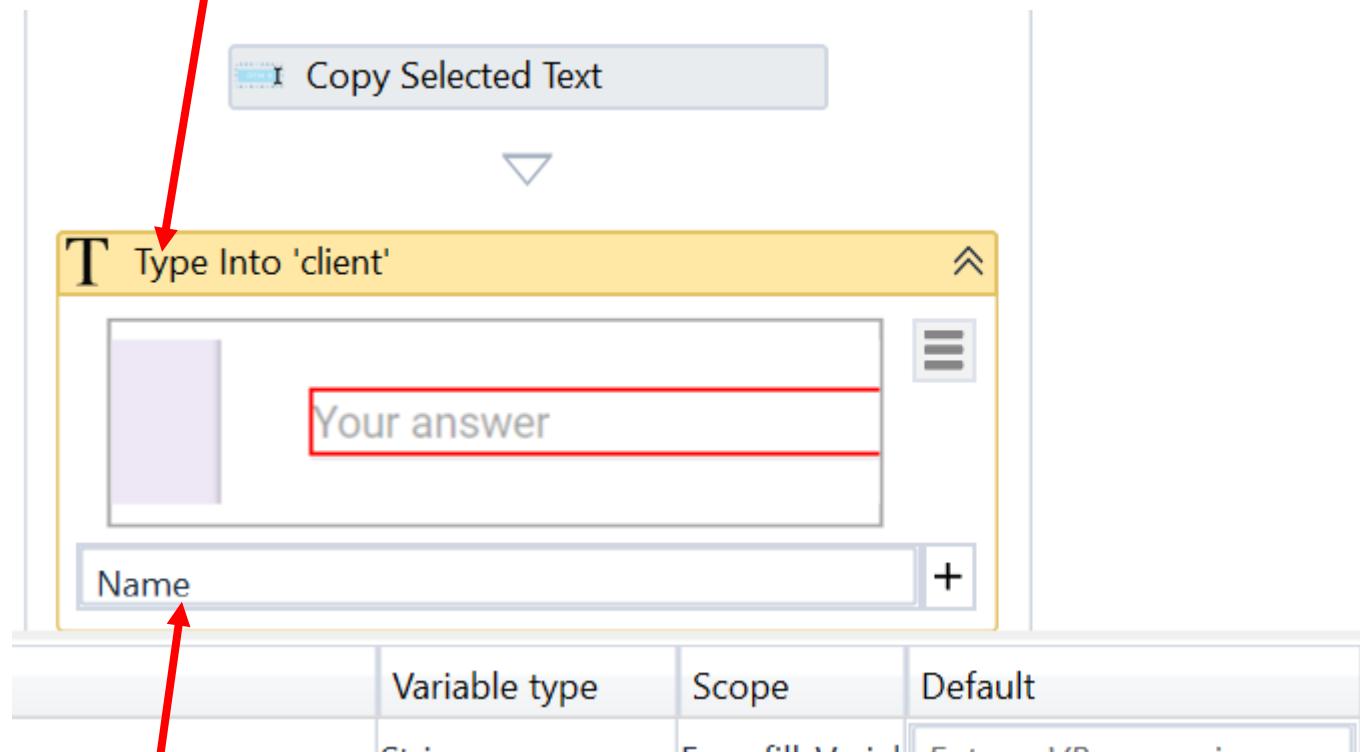
Exercise 5.2 (Step 7)

**Repeat Step 5 and 6 to complete
copying for the next 2 variables, ie
Job title and Phone number**

Exercise 5.2 (Step 8)

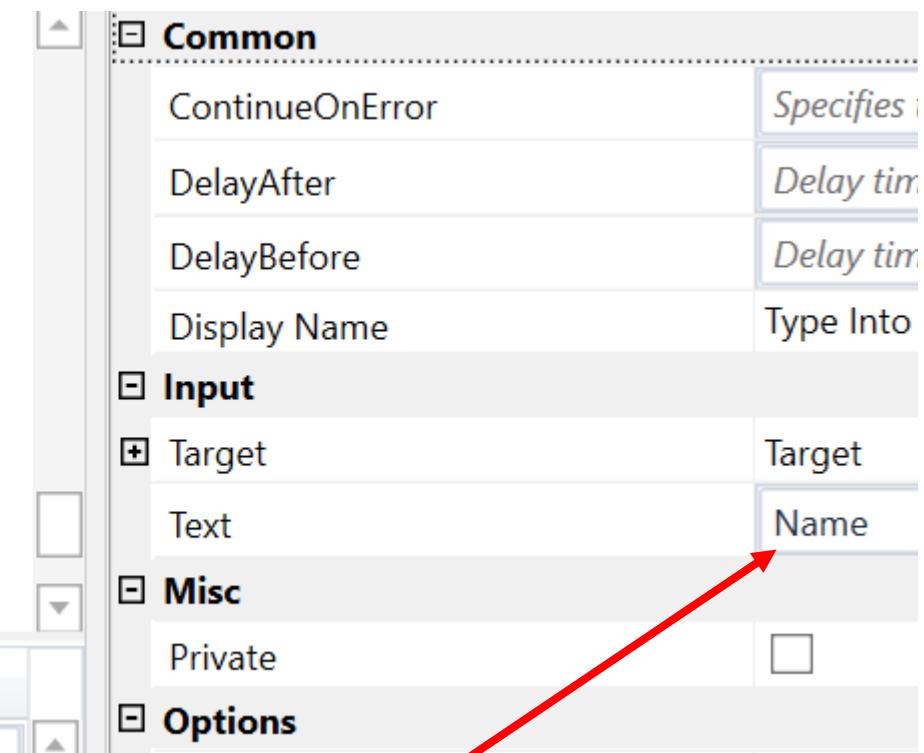
1

Insert a “Type Into”



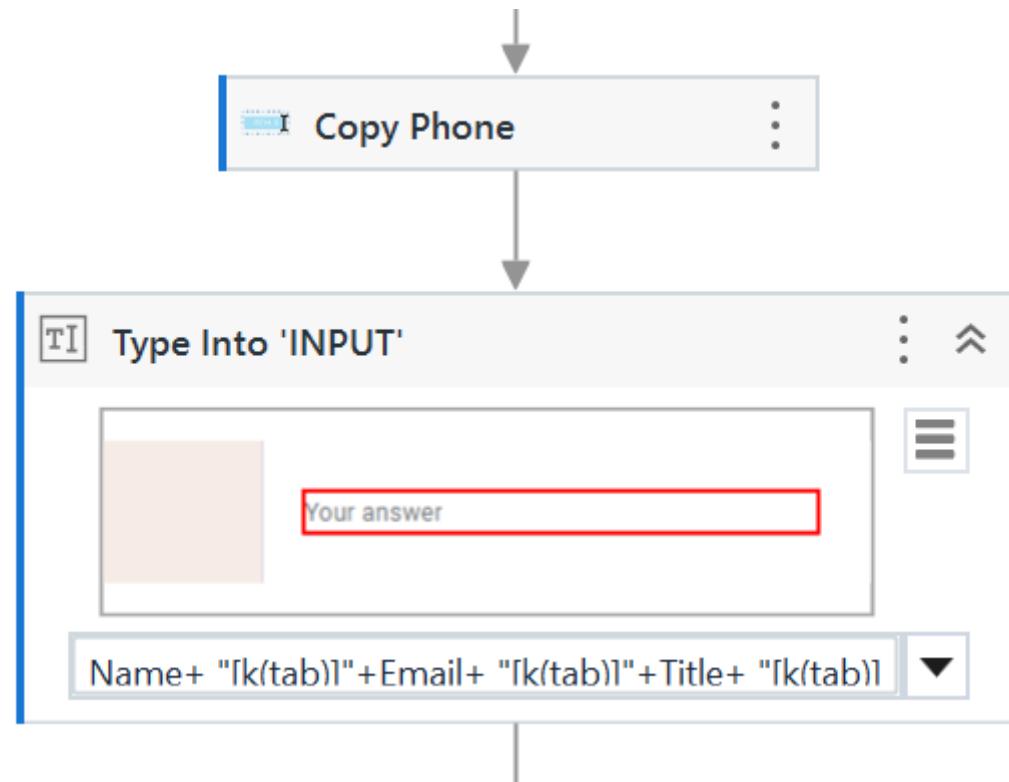
2

Type “N” or “n” and select the variable “Name” (without the “ ”)



The variable “Name” will also appear here.

Exercise 5.2 (Step 9)

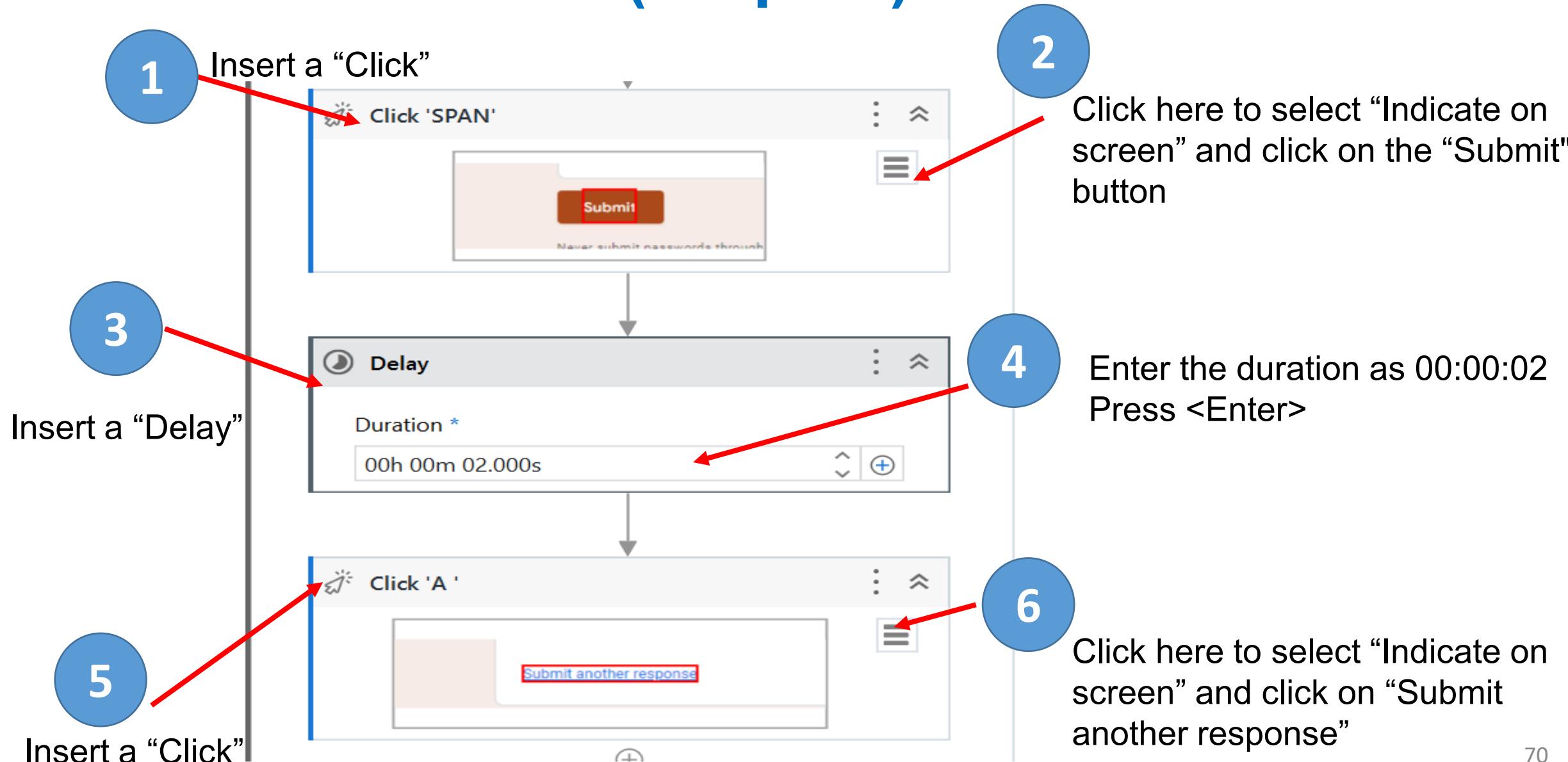


- Click on + and select tab
- Type + Email
- Click on + and select tab
- Type + Title
- Click on + and select tab
- Type + Phone

You should see:

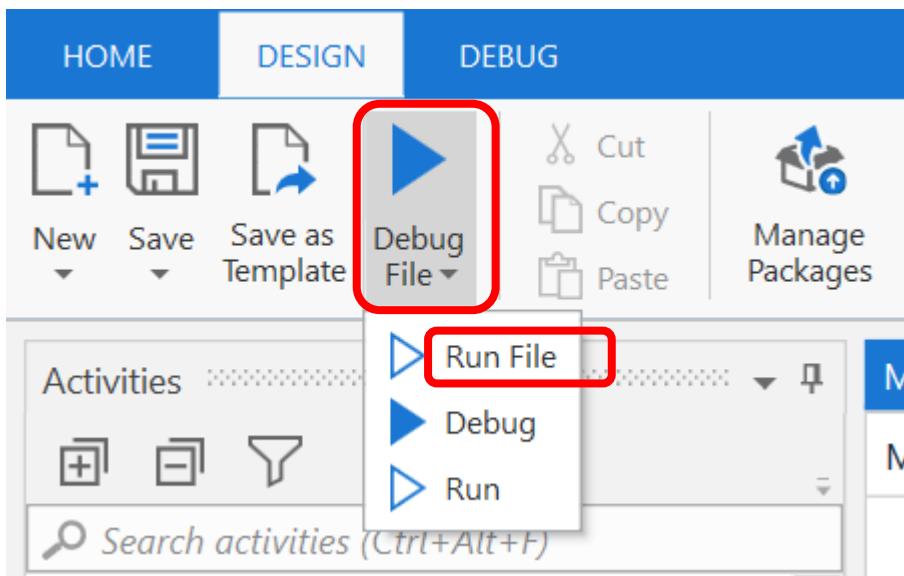
Name + "[k(tab)]" + Email +
"[k(tab)]" + Title + "[k(tab)]" + Phone

Exercise 5.2 (Step 10)



Exercise 5.2

Click the **Debug File** button and select “Run File”



Exercise 5.2 – Review

How would you repeat the data entry for the other records?



Ex5.3-5.5 Google Form Fill

1. Continue to follow Google Form Fill from Ex5.3 – 5.5
Creating a new Workflow for each exercises and name each accordingly

2. Use **Variables, Do-While, If-Else, While Control Flows**

3. Refer to Ex5.2-5.5 in “RPA UiPath Hands-On Guide”
for step-by-step instruction



RPA Form 1
RPA Training Form 1

Name
Your answer

Email
Your answer

Job Title
Your answer

Phone Number
Your answer

Submit

Google Form Fill

Exercise 5.3

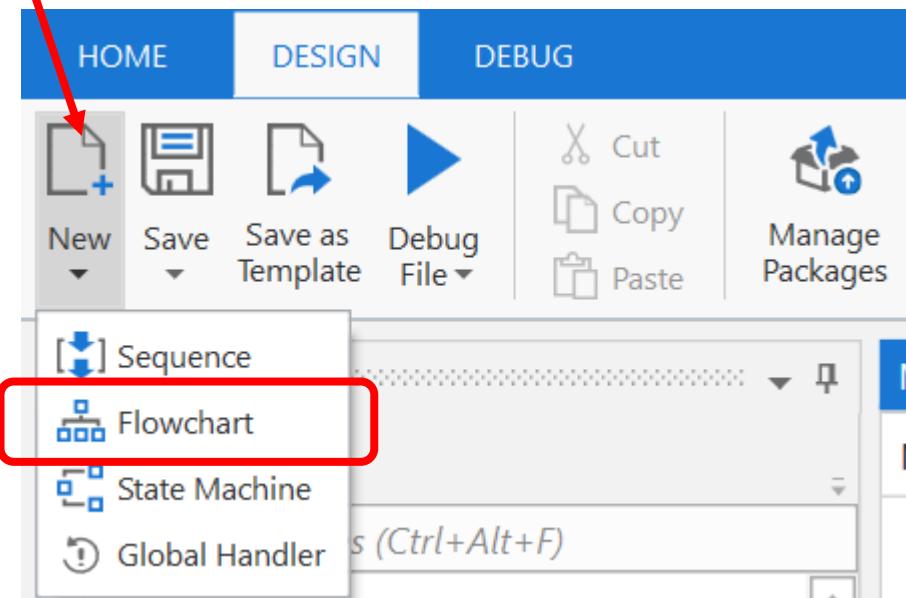
Exercise 5.3 – Looping using the Do While function

- **Do While** allows execution of contained activities first and then loops if a certain condition is true.
- In this exercise we will make use of the **Variable** and **Do While** function to execute Exercise 5.2 and complete data entry for the 5 records.

Exercise 5.3 (Step 1)

1

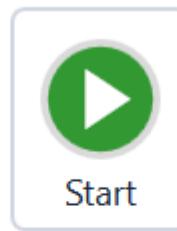
Click on New and select “Flowchart” and name it “Exercise 5.3”



Exercise 5.3 (Step 2)

1

Copy the “Formfill” sequence and paste it in the flowchart of Exercise 5.3.
But **Do Not** connect it to the start button.



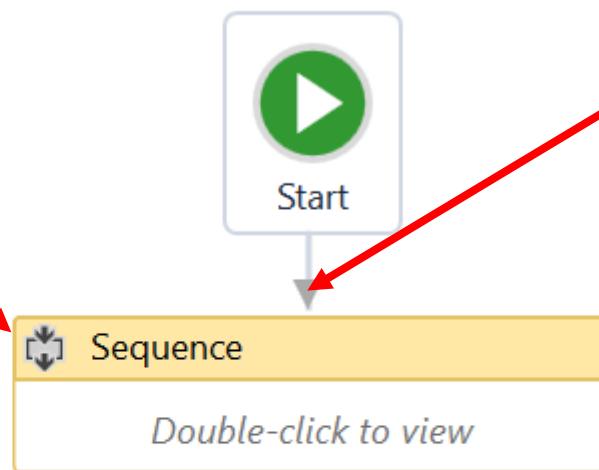
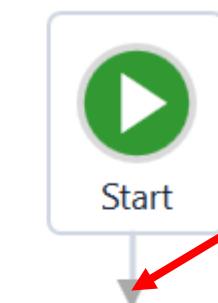
Formfill

Double-click to view

Exercise 5.3 (Step 3)

1

Insert a new “Sequence”

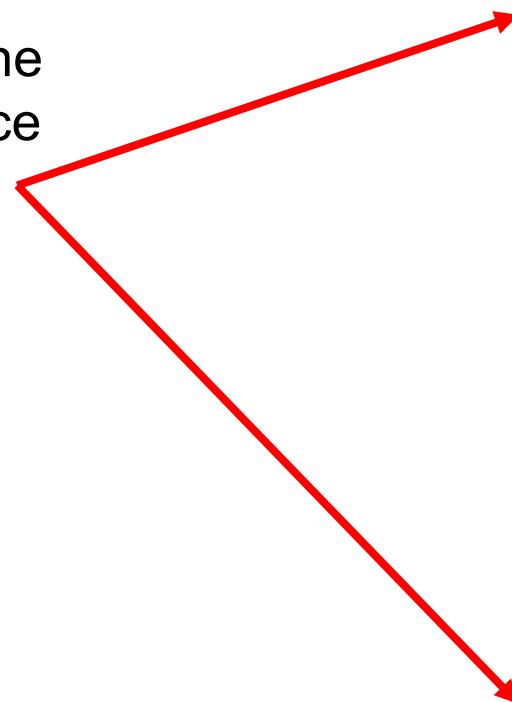


- Right click on the sequence and select “Set as start node” to join to the start button
- You can rename the Sequence to “Go to Cell A2”



Exercise 5.3 (Step 4)

Copy these two steps from the “Formfill” to the new sequence (“Go to Cell A2”)



The screenshot shows a software interface with two main panels:

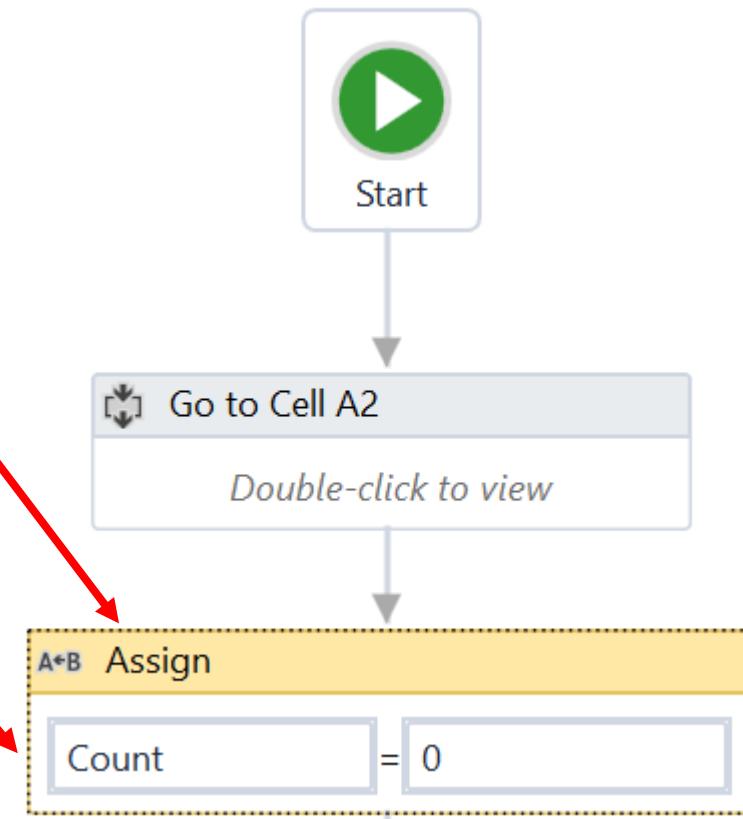
- Send Hotkey 'EXCEL7'**: This panel displays a table of data with columns: Name, Email, Job Title, Phone number, and Feedback. The data includes entries for Avin Up, Bob Lang, David, Kevin Drost, and Sue Lang. Below the table are checkboxes for Alt, Ctrl, Shift, and Win, and a dropdown menu for the Key, currently set to 'g'.
- Type Into 'EDTBX'**: This panel has a "Reference:" section containing a red rectangular box and a text input field containing the value "NList1!A2[enter]".

Exercise 5.3 (Step 5)

1 Insert “Assign” after the Sequence

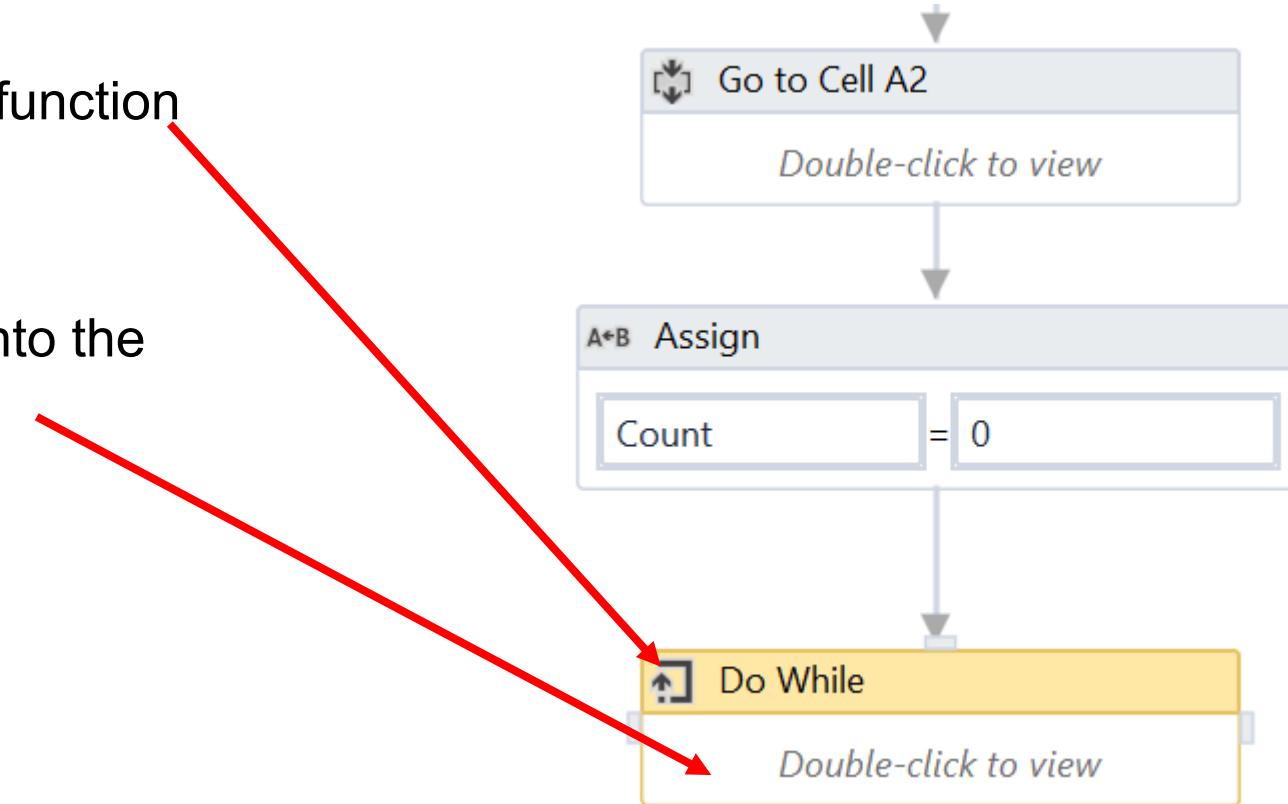
2 Create a new variable “Count”
of type Int32 and Assign
Count = 0

NB: We are using the variable “Count” as a counter and hence we are setting it to 0 at the beginning.



Exercise 5.3 (Step 6)

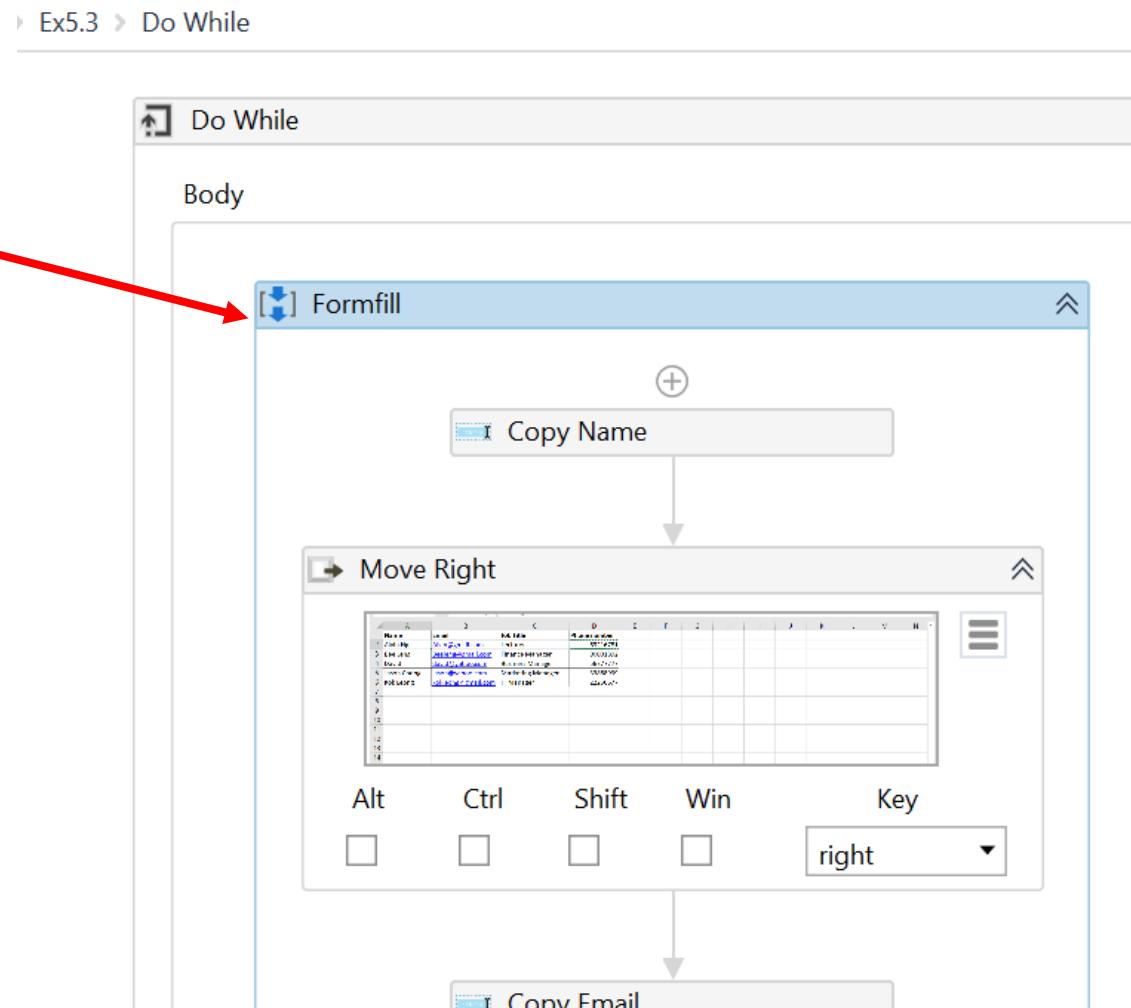
- 1 Insert a “Do While” function
- 2 Double click to go into the “Do While” function.



Exercise 5.3 (Step 7)

Copy and paste the “Formfill” into the Body of the **DO WHILE**.

NB: We do not need the original first 2 steps in the formfill. Hence you can delete the steps after it's copied over to the “Go to Cell A2” sequence.

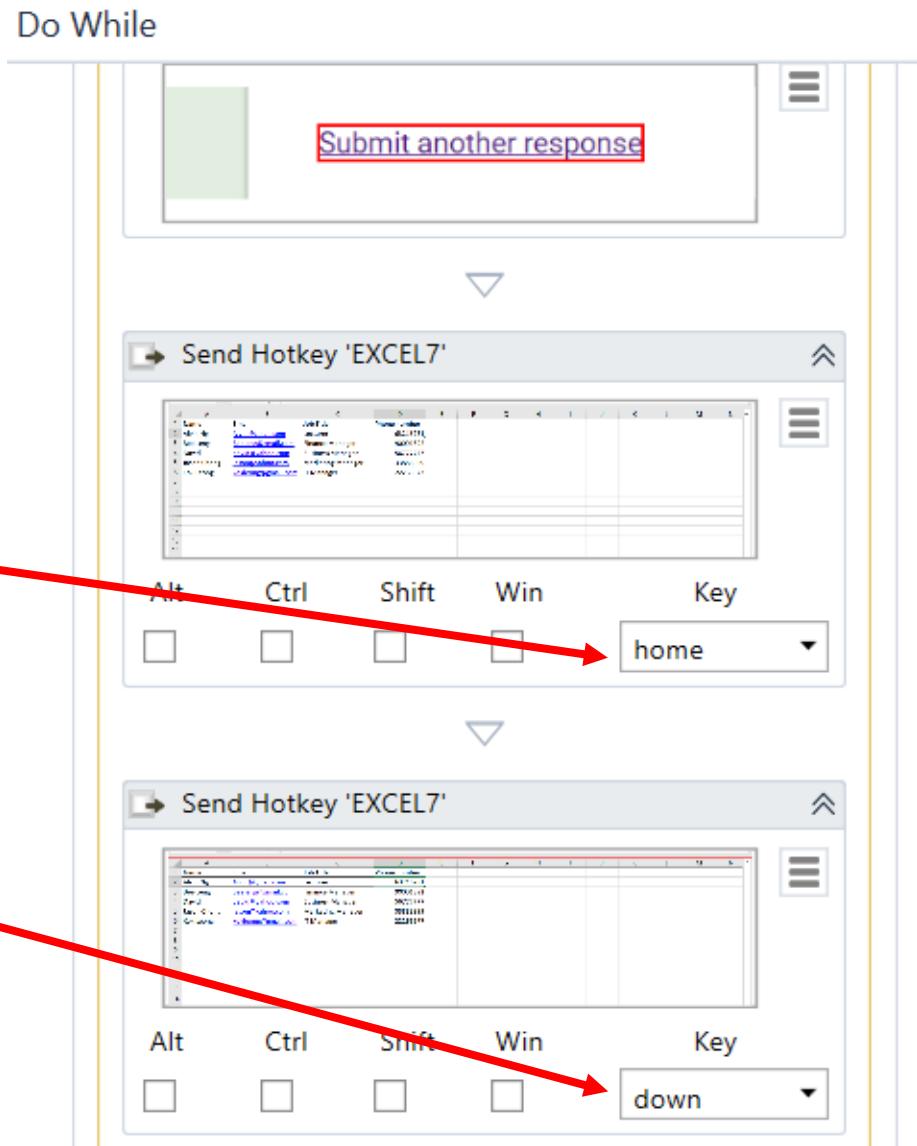


Exercise 5.3 (Step 8)

Insert these 2 steps at the end of
“Formfill”

- 1 Insert a “Send hotkey”
and register the “home” key

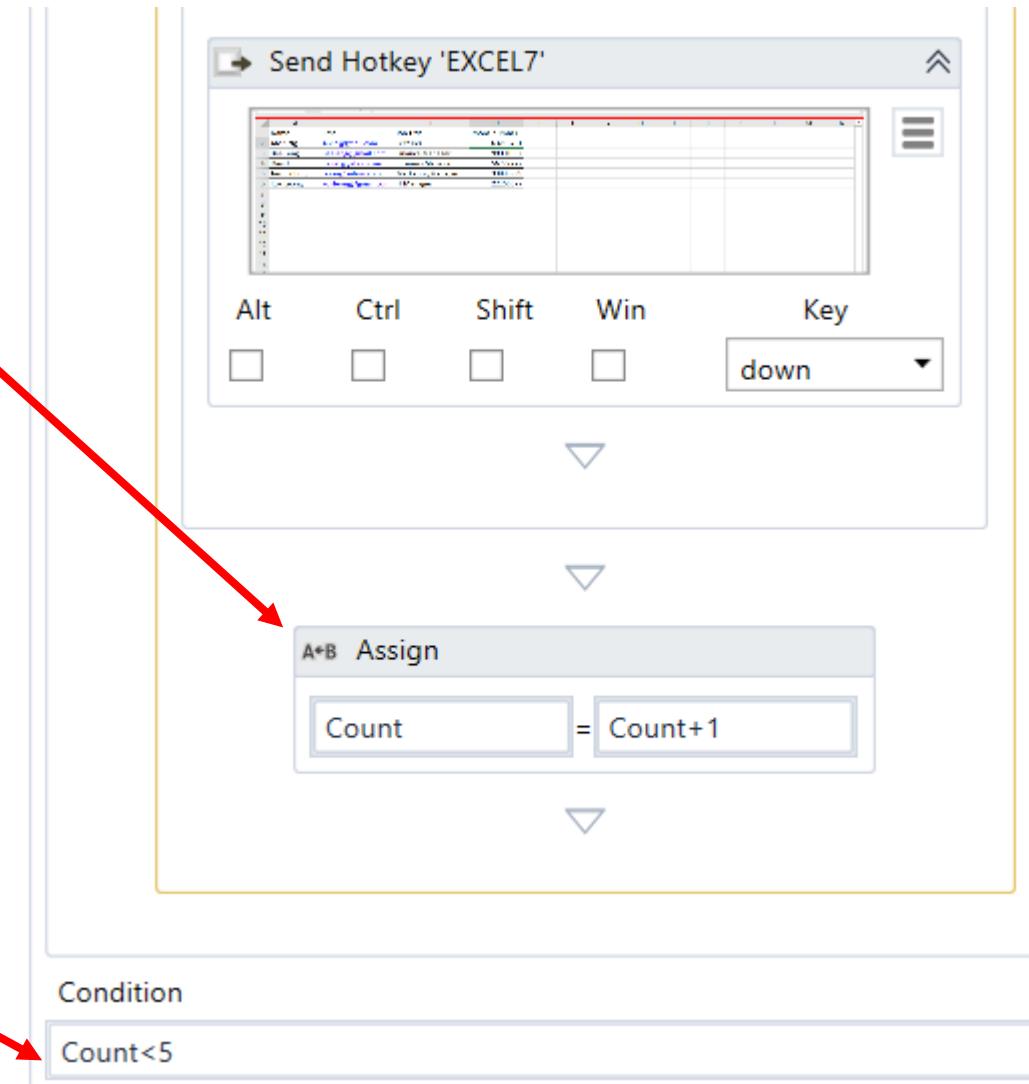
- 2 Insert a “Send hotkey”
and register the “down” key



Exercise 5.3 (Step 9)

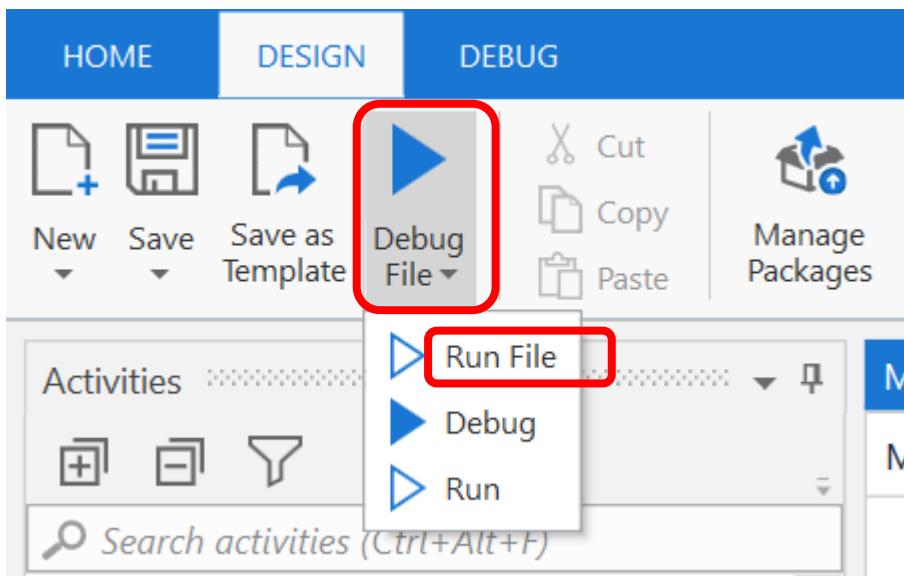
- 1 Insert an “Assign” after the hotkey “down” and assign Count = Count+1

- 2 To repeat 5 times, set the condition to Count<5.



Exercise 5.3

Click the **Debug File** button and select “Run File”



Google Form Fill

Exercise 5.4

Exercise 5.4 – Looping using the Do While and If-Then-Else function

- In this exercise we will make use of the **Do While** and **If-Then-Else** function to execute Exercise 5.3 and complete data entry for the 5 records with additional drop-down list.
- You need to use the excel file tab “NList 2” and
- Google form 2

Phone number *

Your answer

How do you find RPA? *

Choose

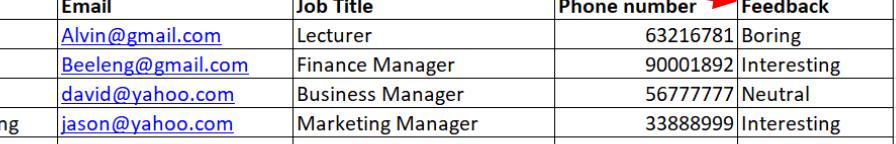
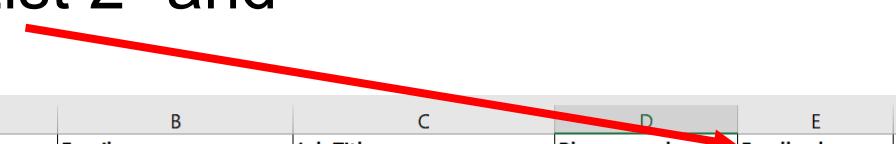
Boring

Neutral

Interesting

Submit

This form was created with Google Forms.

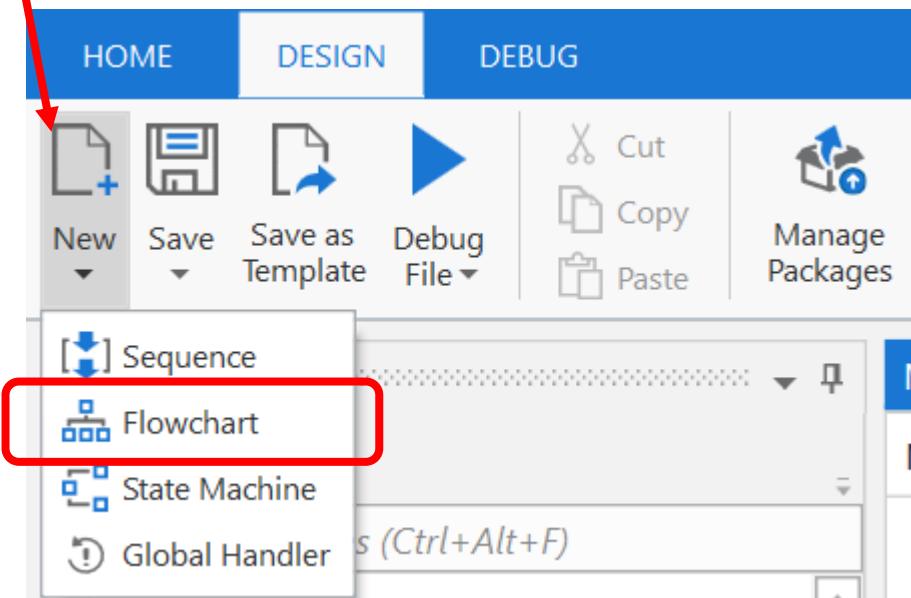


	A	B	C	D	E
1	Name	Email	Job Title	Phone number	Feedback
2	Alvin Ng	Alvin@gmail.com	Lecturer	63216781	Boring
3	Bee Leng	Beeleng@gmail.com	Finance Manager	90001892	Interesting
4	David	david@yahoo.com	Business Manager	56777777	Neutral
5	Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
6	Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring

Exercise 5.4 (Step 1)

1

Click on New and select “Flowchart” and name it “Exercise 5.4”



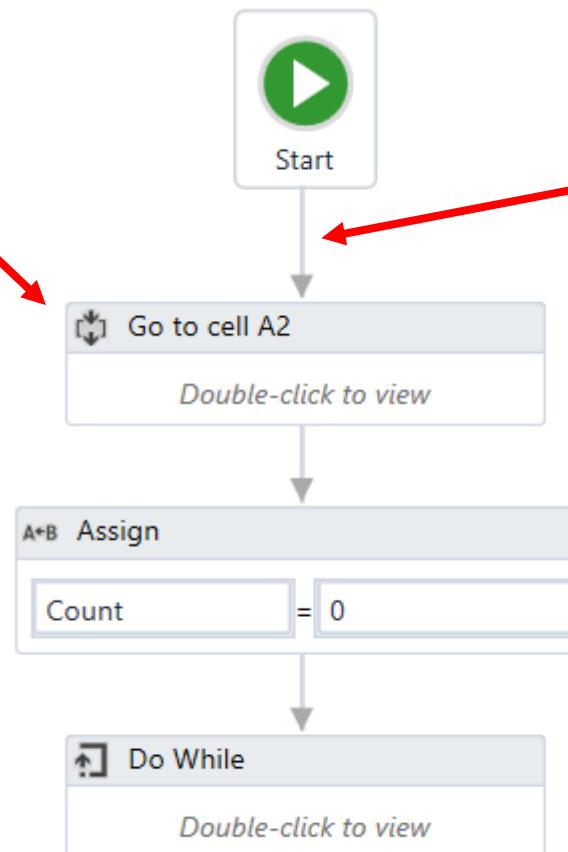
Exercise 5.4 (Step 2)

1

Copy the Exercise 5.3 sequences and paste it in the flowchart of Exercise 5.4

2

Right click on the sequence and select “Set as start node” to join to the start button.



Exercise 5.4 (Step 3)

- In the **Do While** loop, create a new variable, “Feedback” and add the following 2 steps after the “Copy Phone No”

1

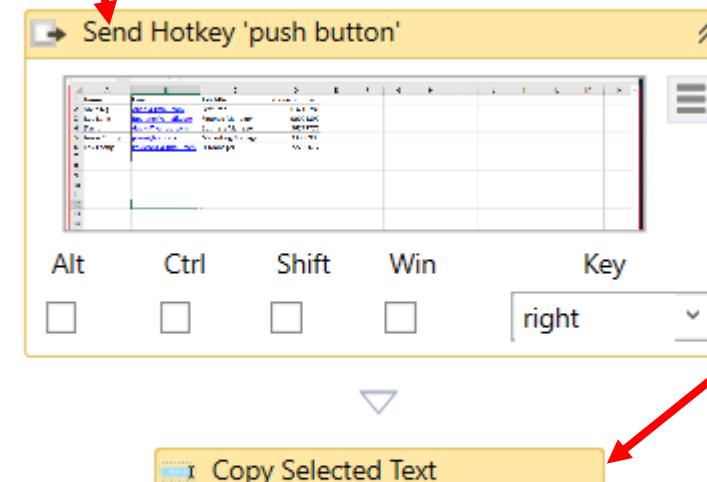
Create a new variable “Feedback”

2

Insert a hotkey to move 1 step to the right in excel (NList2)

3

Insert a “Copy Selected Text” and send the result to the variable ‘Feedback’



Name

Feedback

Variable type

String

Scope

Formfill_Va

Exercise 5.4 (Step 4)

1

Insert a “Click” to click on the “Choose” portion of the drop down list

NB: You may need to also re-image the form for filling in the fields as the form has now change from form 1 to form 2.

The screenshot shows a form builder interface with two forms and a variable table below them.

Top Form: A text input field labeled "Type Phone No" with the placeholder "Your answer". Below it is a dropdown menu labeled "Phone".

Bottom Form: A text input field labeled "Click 'DIV'" with the placeholder "How do you find RPA?". To its right is a dropdown menu labeled "Choose".

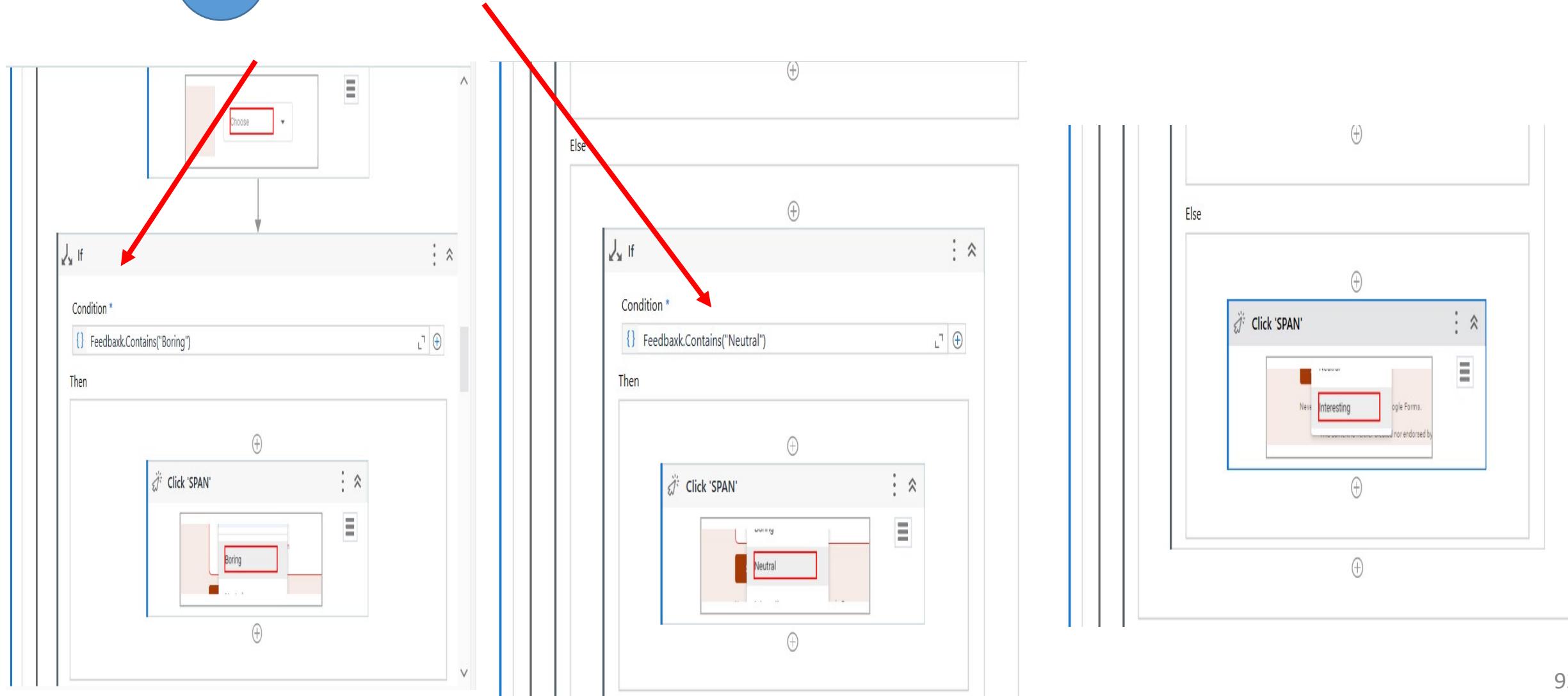
Variable Table:

Name	Variable type	Scope
Phone	String	Formfill
Feedback	String	Formfill

Exercise 5.4 (Step 5)

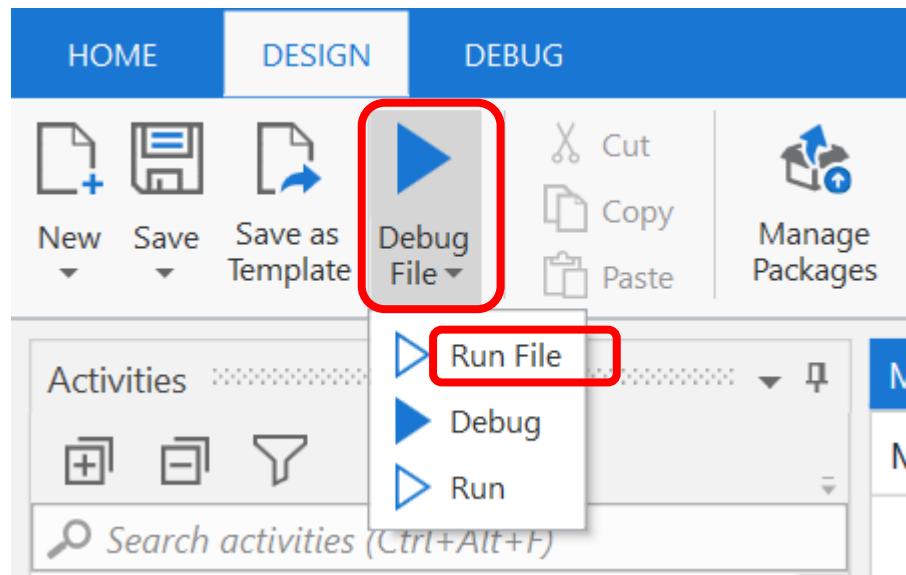
1

Insert the following “If-Then-Else” and enter the conditions accordingly



Exercise 5.4

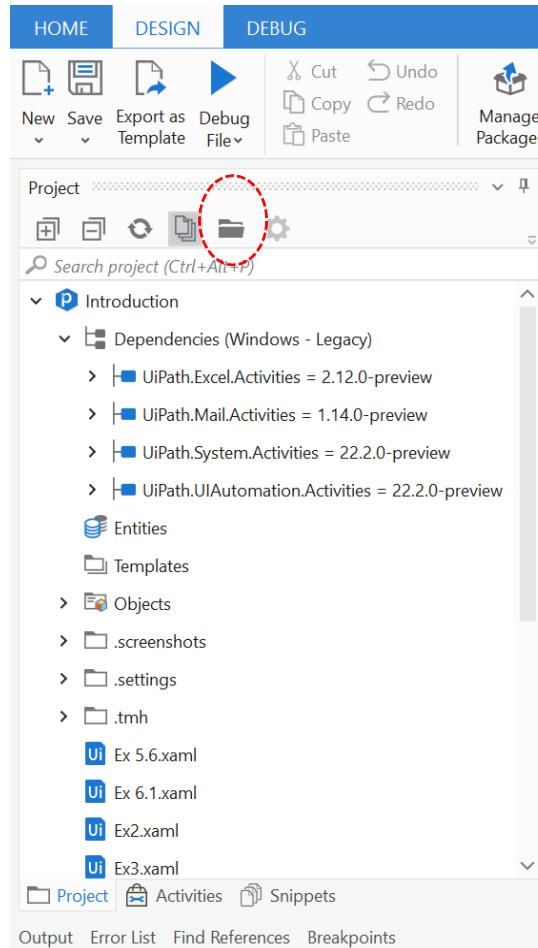
Click the **Debug File** button and select “Run File”

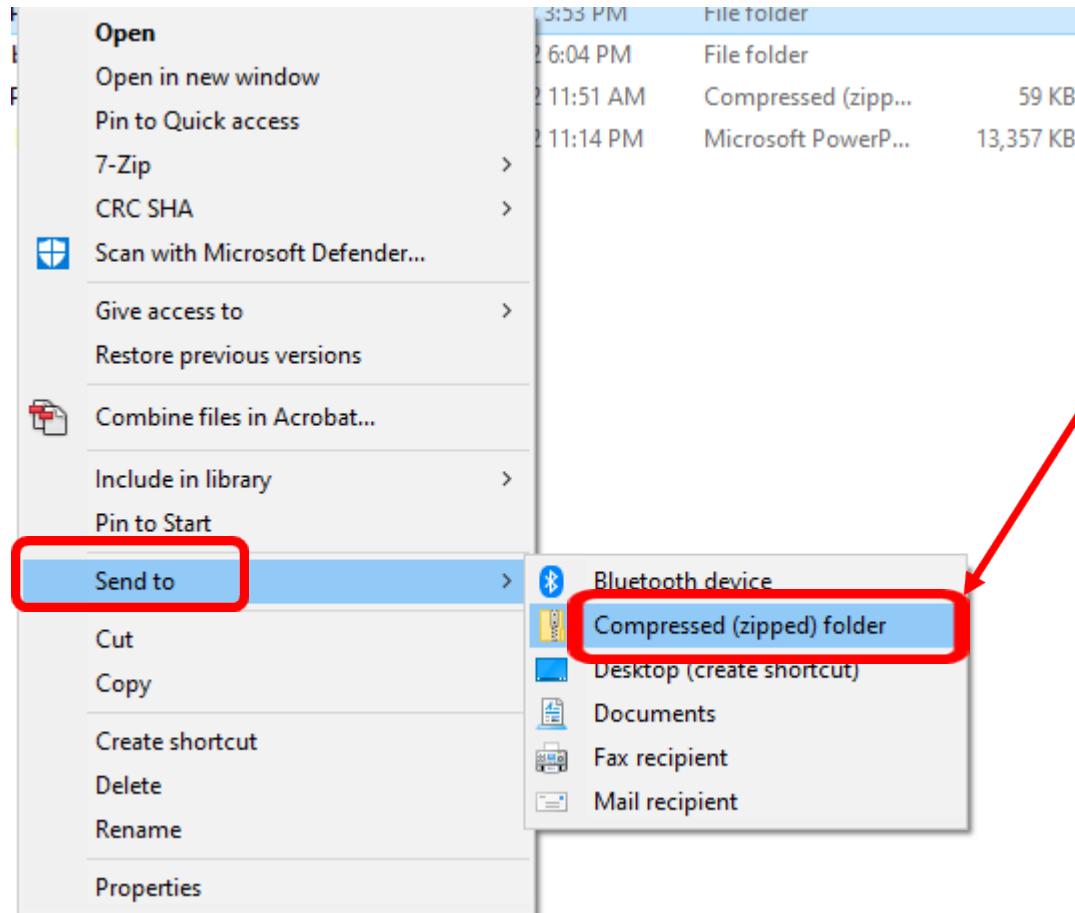


How to share the scripts

Official (Closed) Non-sensitive

To go to the project folder in file explorer, click on the folder button at the project panel here.





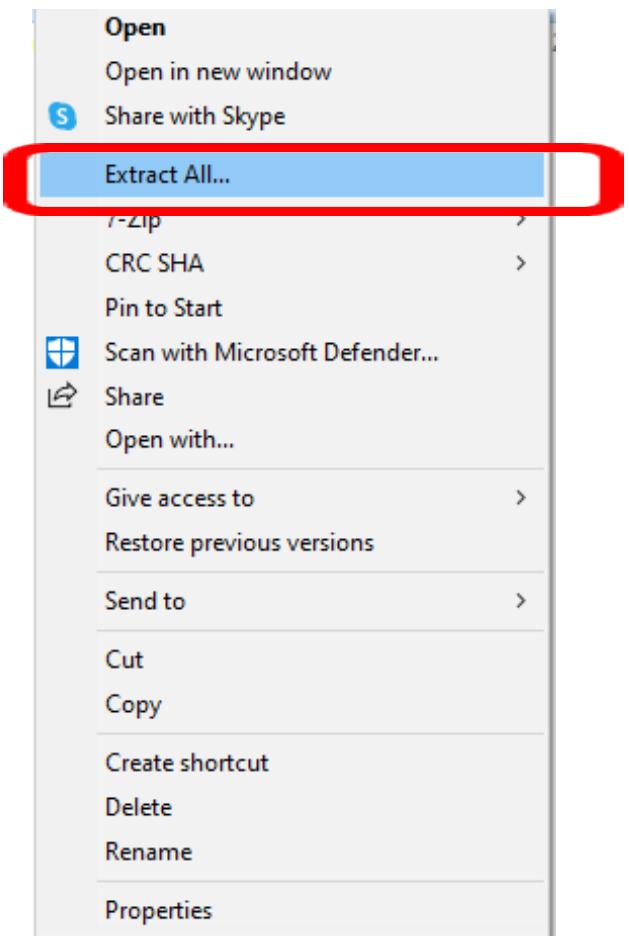
1

Right click on the project folder followed by clicking Send to > Compressed (zipped) folder to zip the project folder.

2

Remember to zip the .screenshots file.

Name	Date modified	Type	Size
.entities	23 Mar 2022 1:41 AM	File folder	
.objects	23 Mar 2022 1:41 AM	File folder	
.screenshots	1 Apr 2022 4:23 PM	File folder	
.settings	23 Mar 2022 1:41 AM	File folder	
.templates	23 Mar 2022 1:41 AM	File folder	
.tmh	23 Mar 2022 1:41 AM	File folder	
Ex2	23 Mar 2022 5:50 PM	Windows.XamlDo...	15 KB

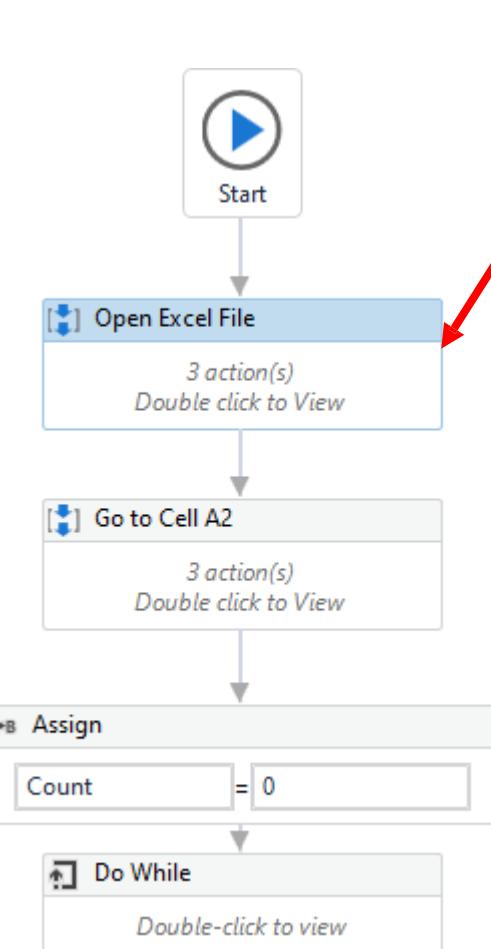


Upon receiving the zipped project folder and zipped .screenshot folder, extract the both of them with a right click then clicking Extract All...

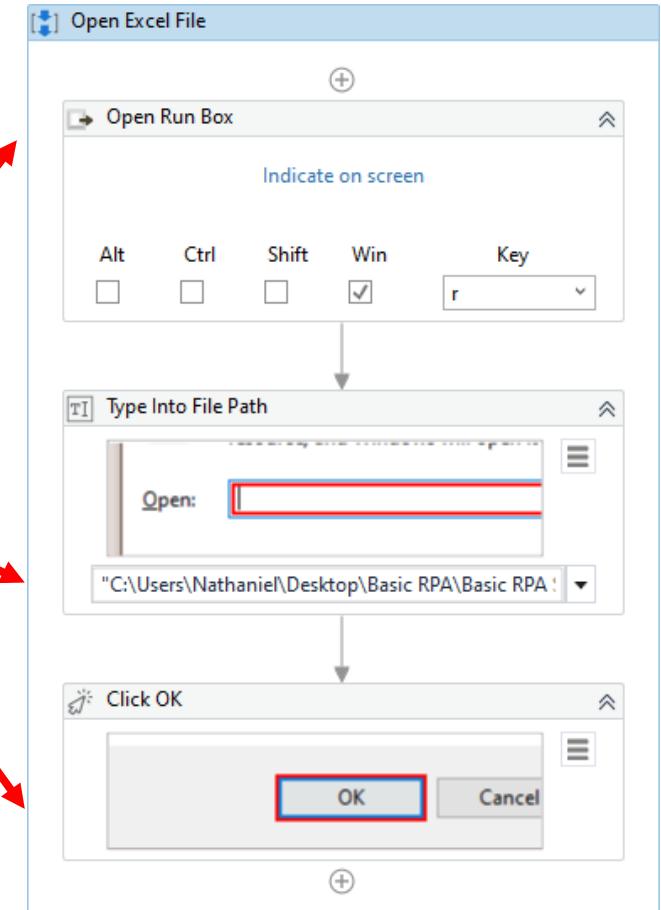
NB: The .screenshot folder needs to be within the project folder after extraction.

Open Files and Webpages from scratch

Open File from Scratch

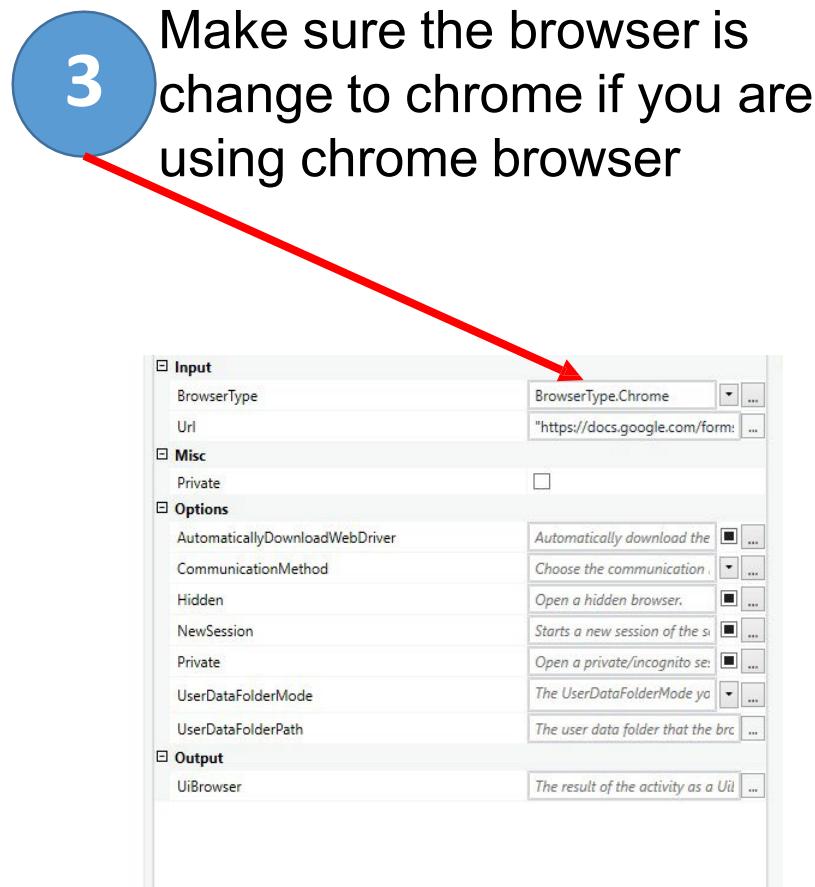
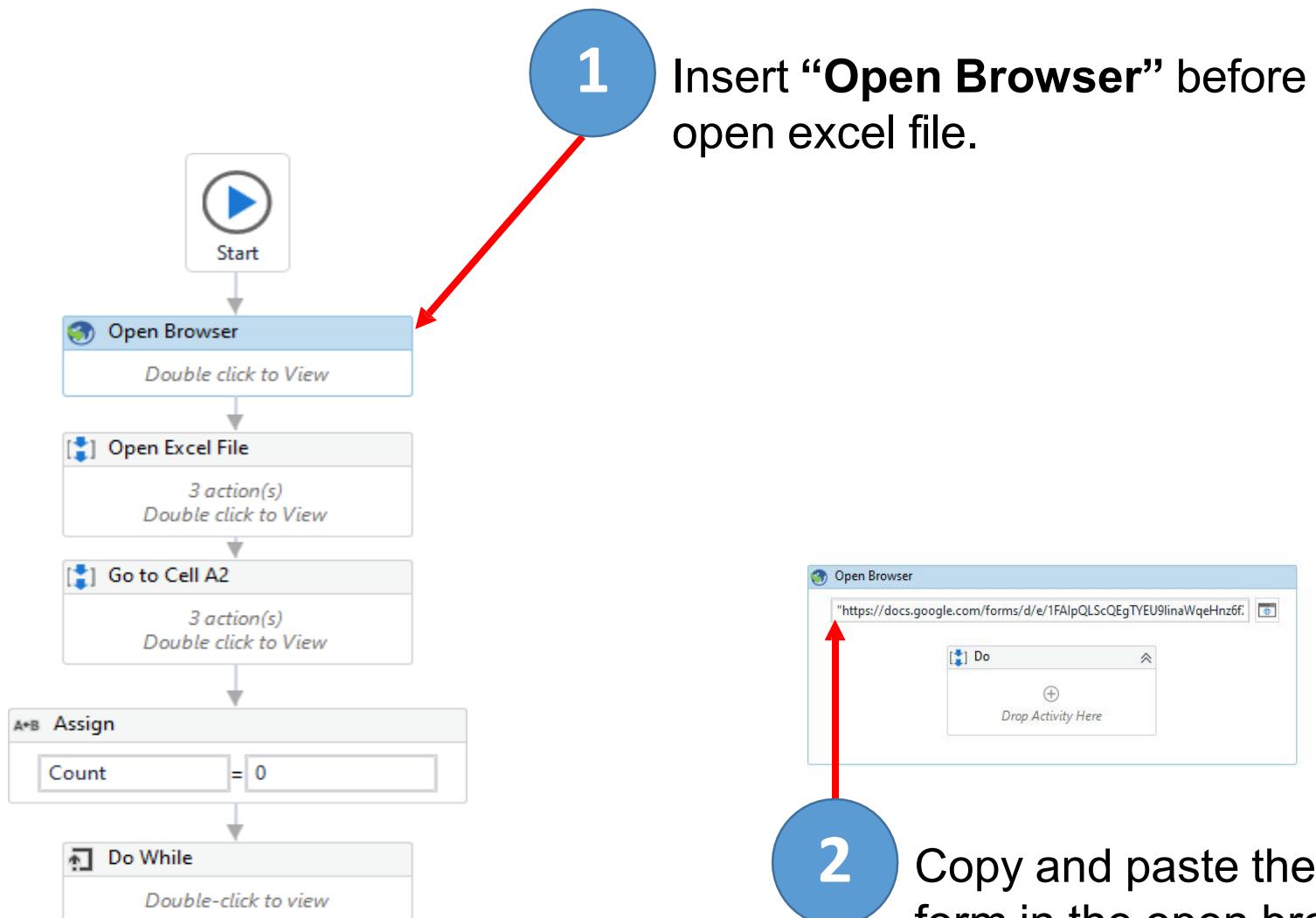


1 Insert a “Sequence” at the beginning of Exercise and rename it “Open Excel File”



2 Add these 3 steps to the sequence.

Open Webpage from Scratch



Google Form Fill

Exercise 5.5

Exercise 5.5 – Looping using the While and If-Then-Else function

- The **While** loop allows execution of contained activities while a condition is true.
- In this exercise we will make use of the **While** and **If-Then-Else** function to execute Exercise 5.4 and complete data entry for the 5 records with additional drop-down list.

Phone number *

Your answer

How do you find RPA? *

Choose

Boring

Neutral

Interesting

Google Forms.

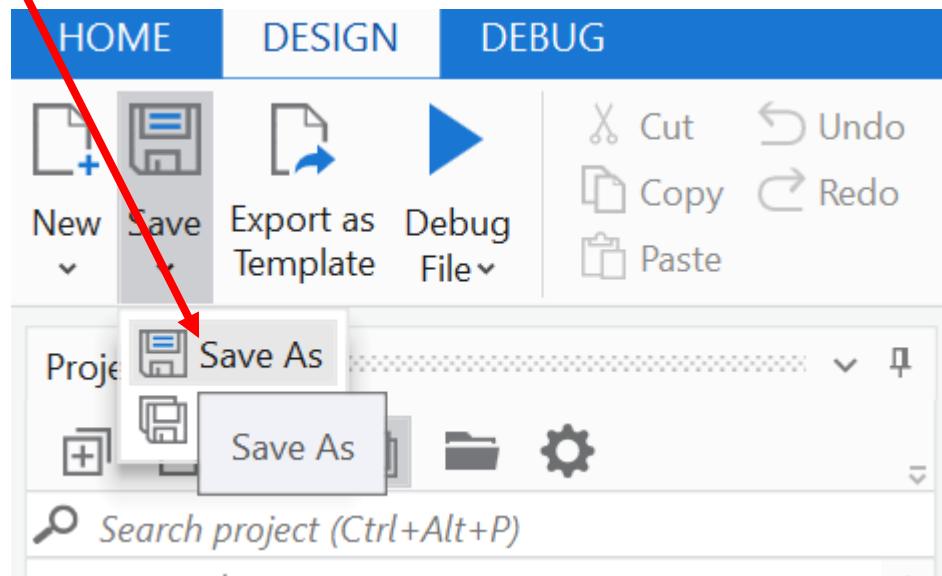
- You need to use the excel file tab “NList 2” and Google form 2

	A	B	C	D	E
1	Name	Email	Job Title	Phone number	Feedback
2	Alvin Ng	Alvin@gmail.com	Lecturer	63216781	Boring
3	Bee Leng	Beeleng@gmail.com	Finance Manager	90001892	Interesting
4	David	david@yahoo.com	Business Manager	56777777	Neutral
5	Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
6	Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring

Exercise 5.5 (Step 1)

1

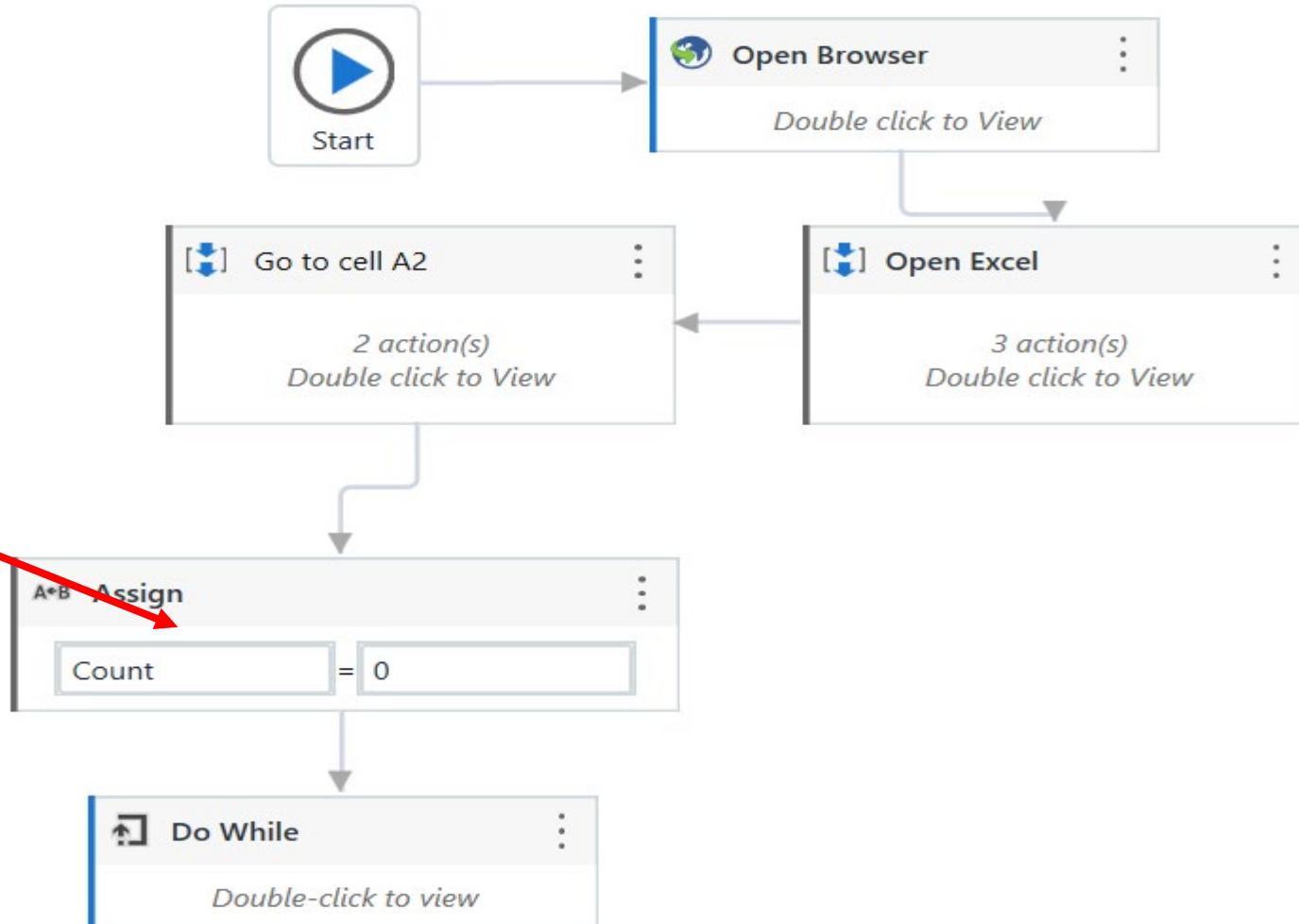
Click on Save\Save As. Save Ex 5.4 as Ex 5.5



Exercise 5.5 (Step 2)

1

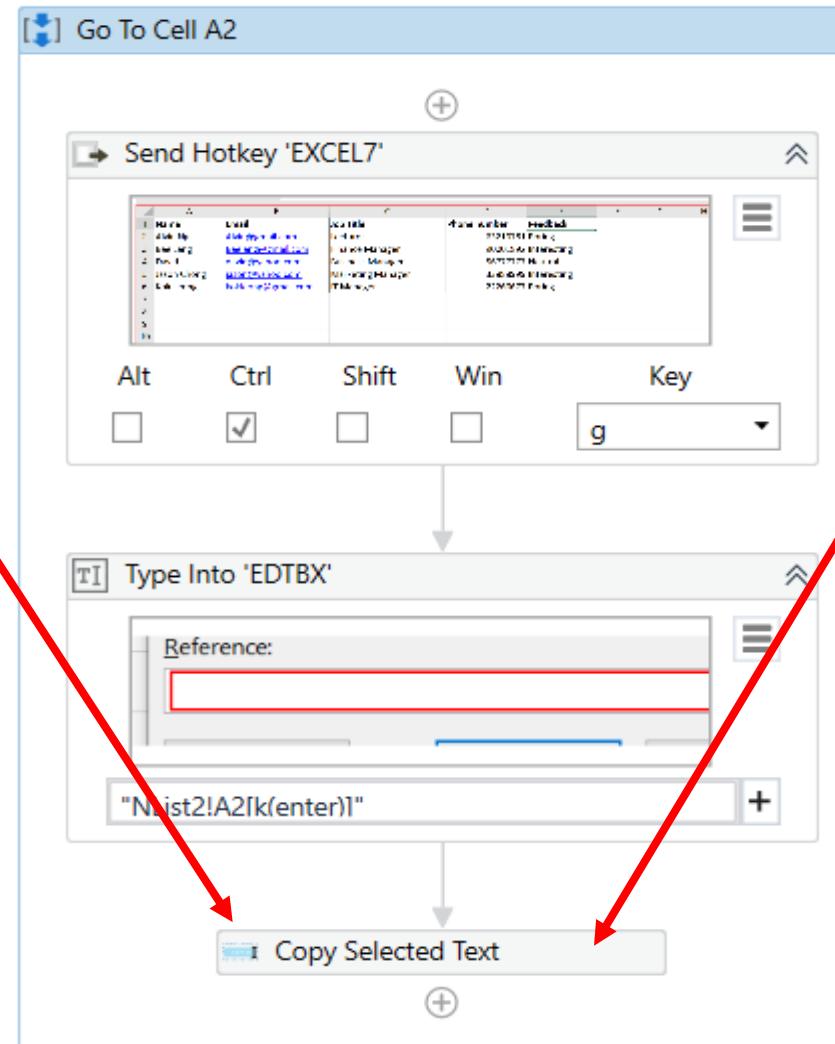
- Delete “Assign”



Exercise 5.5 (Step 3a)

1

Insert “Copy Selected Text” in the “Go to cell A2” sequence



2

The selected text should Output result to the variable “Name”

This is to allow the **While** function to later check if the variable Name is empty

Exercise 5.5 (Step 3b)

- 1 In FormFill sequence, click Variable tab

The screenshot shows a software interface for a FormFill sequence named 'Ex5_4'. The sequence consists of two steps: 'Copy Name' and 'Send Hotkey 'item "D" 9''. Below the steps is a table of variables:

Name	Variable type	Scope	Default
Name	String	FormFill	Enter a VB expression
Email	String	FormFill	Enter a VB expression
Title	String	Body	Enter a VB expression
Phone	String	Ex5_4	Enter a VB expression
Feedback	String	FormFill	Enter a VB expression
Count	Int32	For Ex5_4	Enter a VB expression

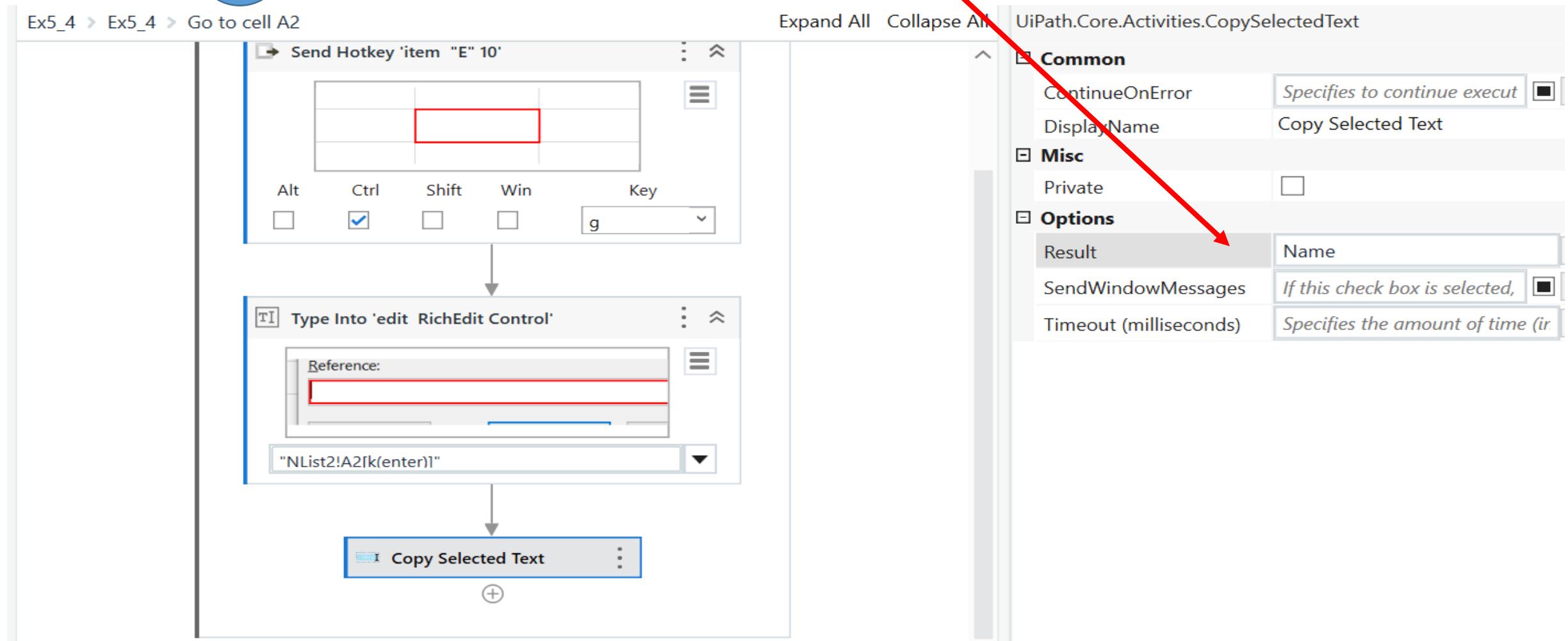
The 'Variables' tab is selected at the bottom left. A red arrow points from the 'Variables' tab to the 'Scope' dropdown for the 'Name' row. Another red arrow points from the 'Ex5_4' option in the dropdown to the 'Ex5_4' entry in the 'Default' column for the 'Count' variable.

- 2 Change the scope of Name variable from "FormFill" to "Ex 5_4"

Exercise 5.5 (Step 3c)

1

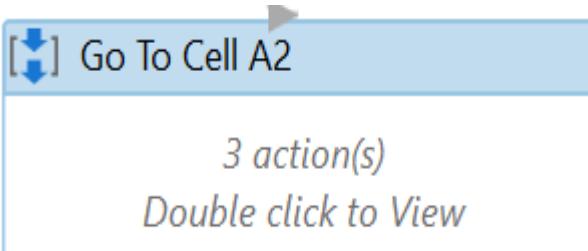
In the “Go to cell A2” sequence, Assign “Copy Selected Text” to Name variable



Exercise 5.5 (Step 4)

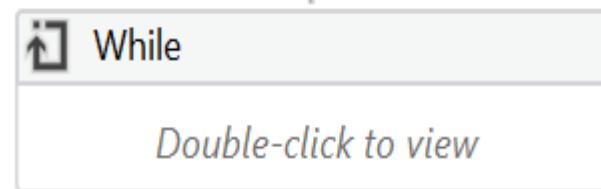
1

Replace the “Do While” with the “**While**” loop



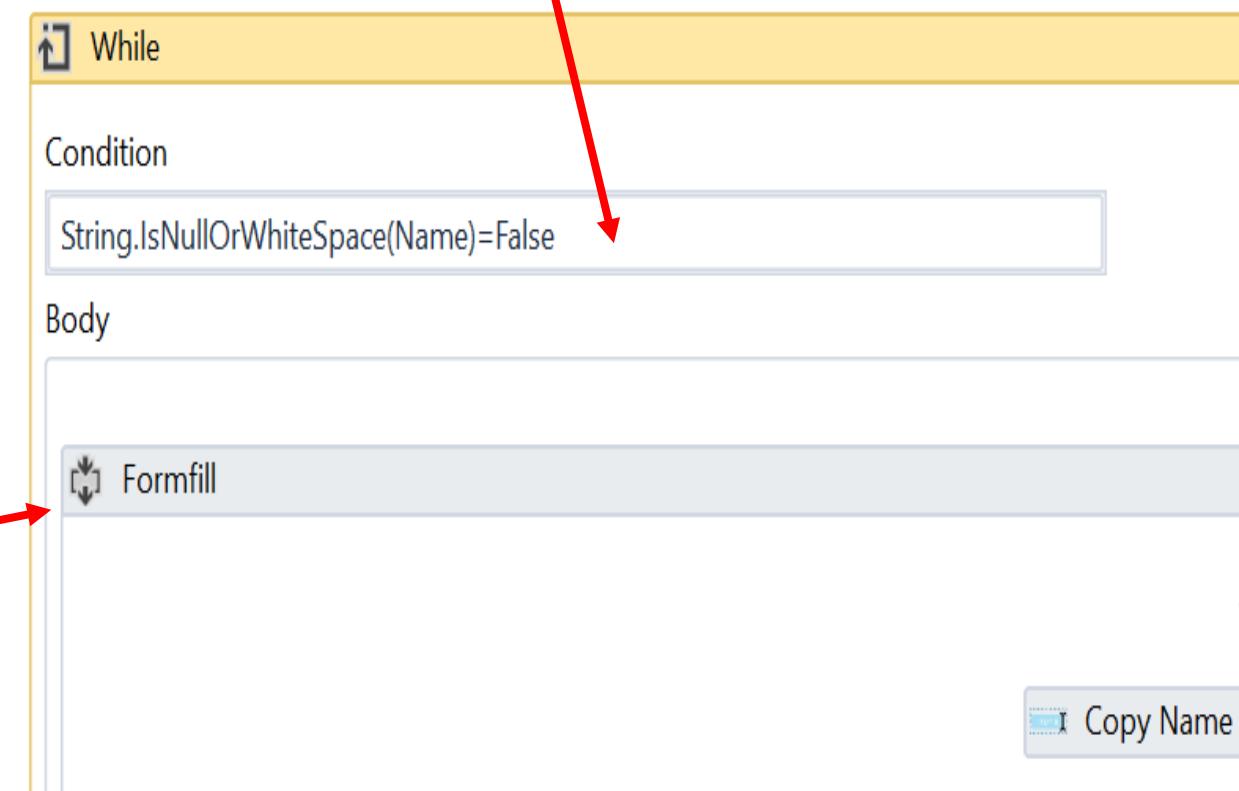
2

Copy the “Formfill” from the **Do While** to the Body of the **While** Loop



3

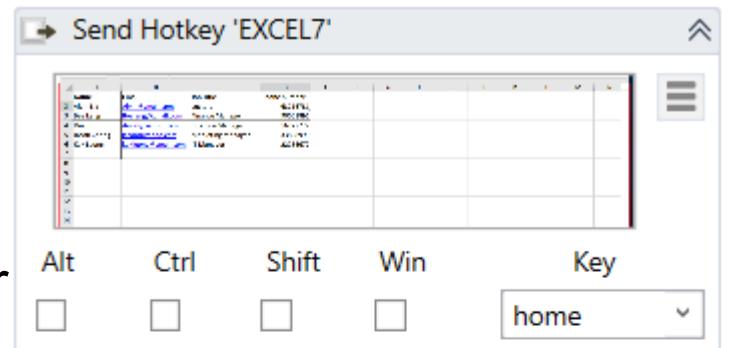
Enter the condition as follows



Exercise 5.5 (Step 5)

1

In the “FormFill”, insert “Copy Selected Text” after the “down” key



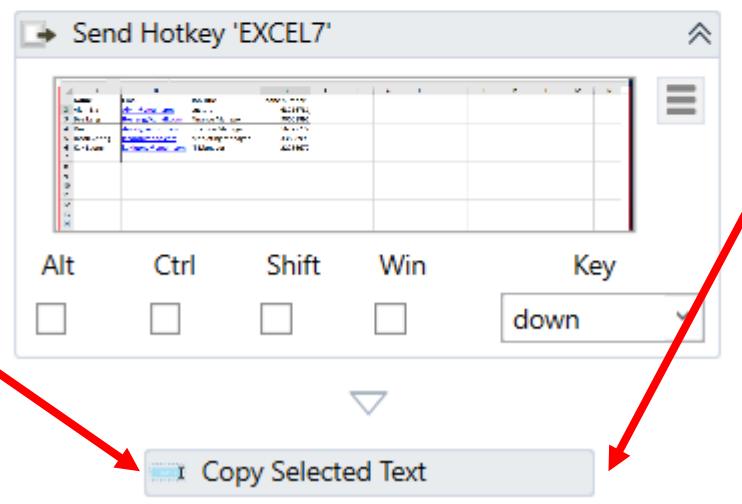
2

The selected text should output result to the variable “Name”

This step copy and store the name of the next record to the variable “Name”

3

In the “FormFill”, delete the Assign activity



4

In the “FormFill”, delete the first “Copy Selected Text”

Exercise 5.5 (Step 6)

1

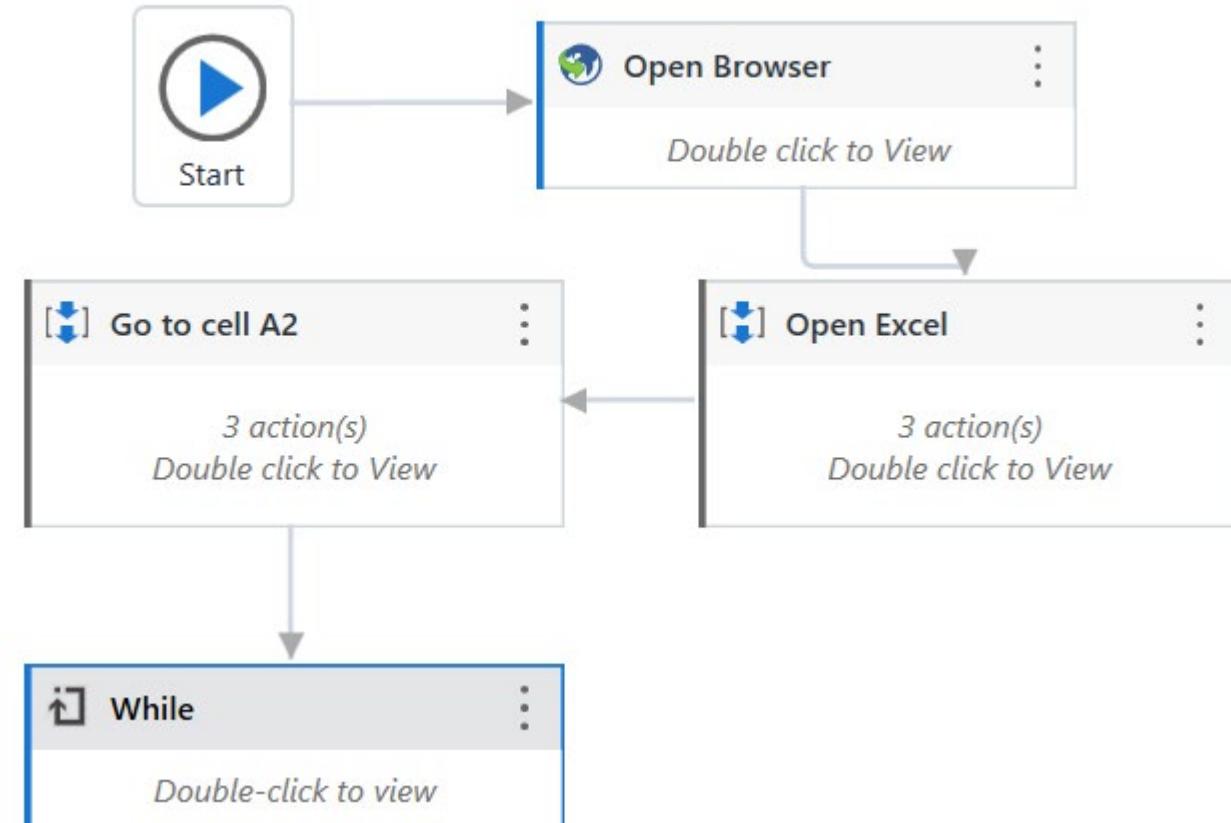
At the highest level, delete
Assign activity

2

Del DO WHILE activity

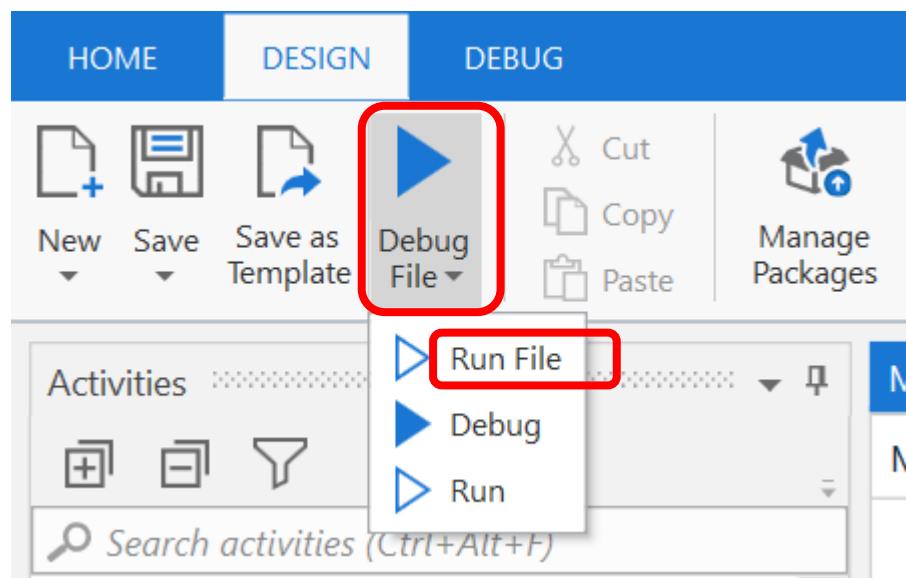
3

Link WHILE activity to “Go
to cell A2”



Exercise 5.5

Click the **Debug File** button and select “Run File”



Working with Excel in the Background and Table variable



Ex5.6 Google Form Fill

1. Open the created “*Robot2_XXXXXX*” process created in the previous lesson.
Create a new Workflow and name it Ex5.6
2. Script Robot to fill in and submit a Google Form from
an Excel table using [For Each Row and Data Table](#),
3. Refer to Ex5.6 in “RPA UiPath Hands-On Guide” for
step-by-step instruction

RPA Form 1

RPA Training Form 1

Name

Your answer

Email

Your answer

Job Title

Your answer

Phone Number

Your answer

Submit



Google Form Fill

Exercise 5.6

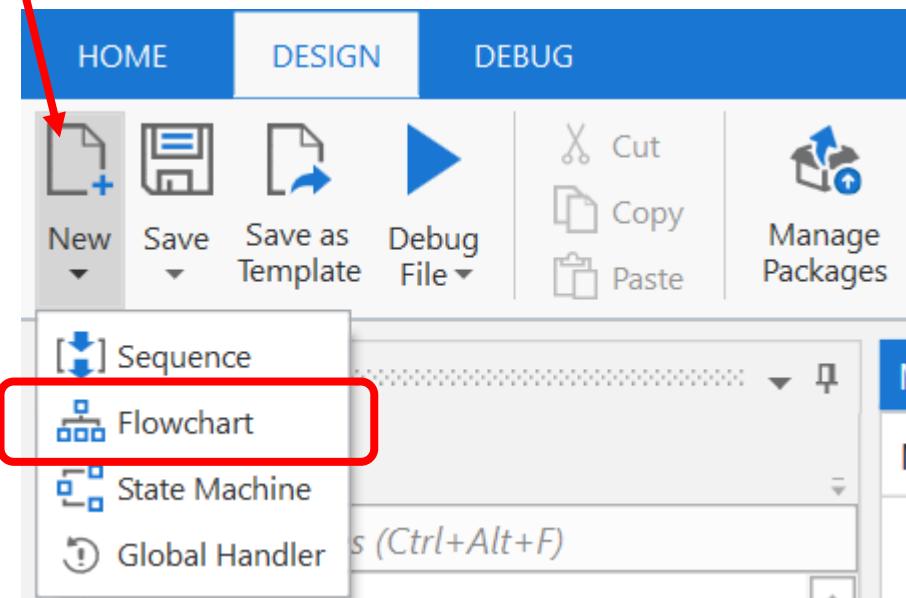
Exercise 5.6 – Excel and Data Entry

- In this exercise information from a closed excel file will be extracted into a **table** variable and update to a google contact form.
- We will be using the “**Read Range**” function
- Along with “**Read Range**” function, we will be using new functions such as “**For each Row**” and “**Get row item**”

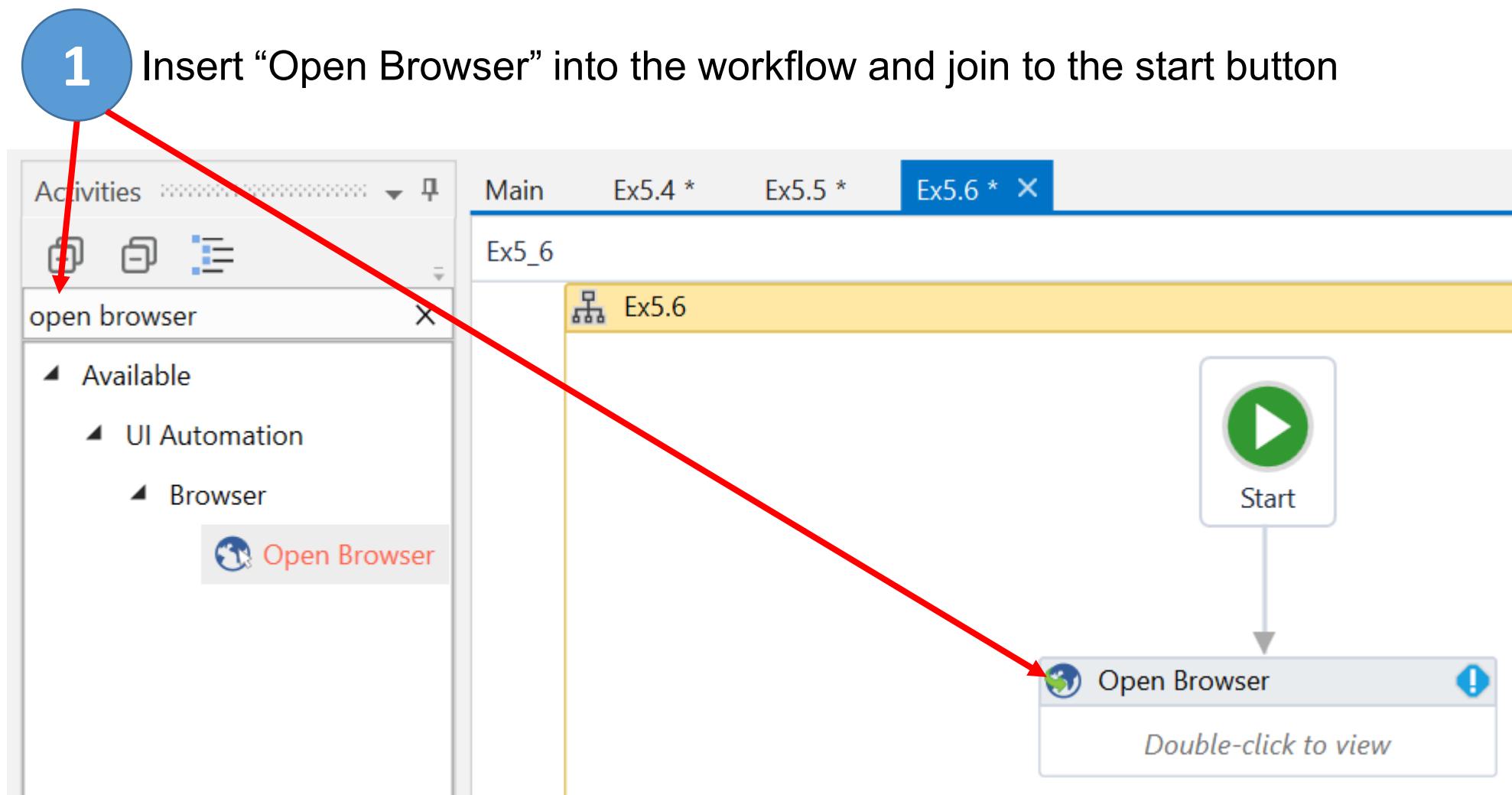
Exercise 5.6 (Step 1)

1

Click on New and select “Flowchart” and name it “Ex 5.6”



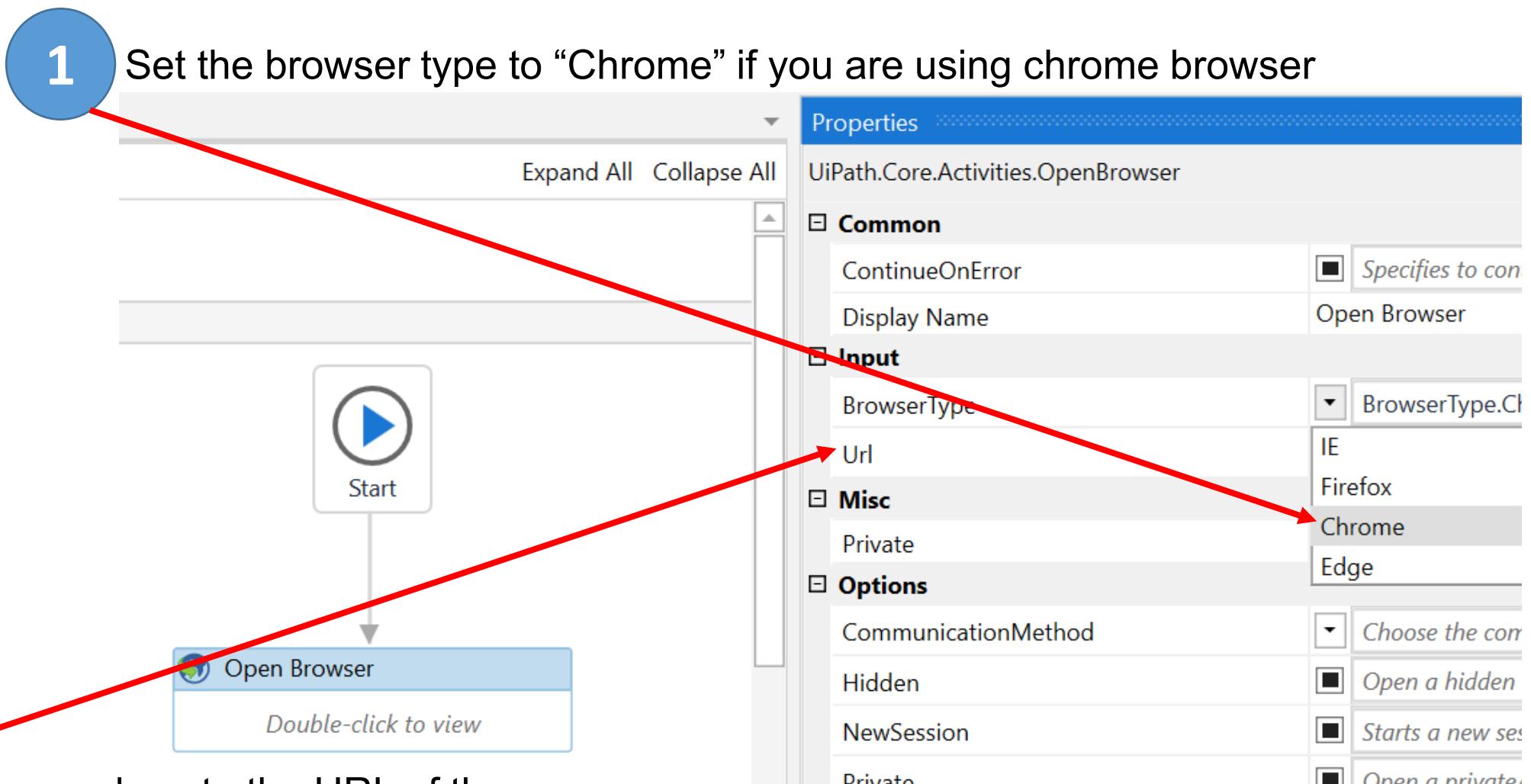
Exercise 5.6 (Step 2)



Exercise 5.6 (Step 3)

1

Set the browser type to “Chrome” if you are using chrome browser



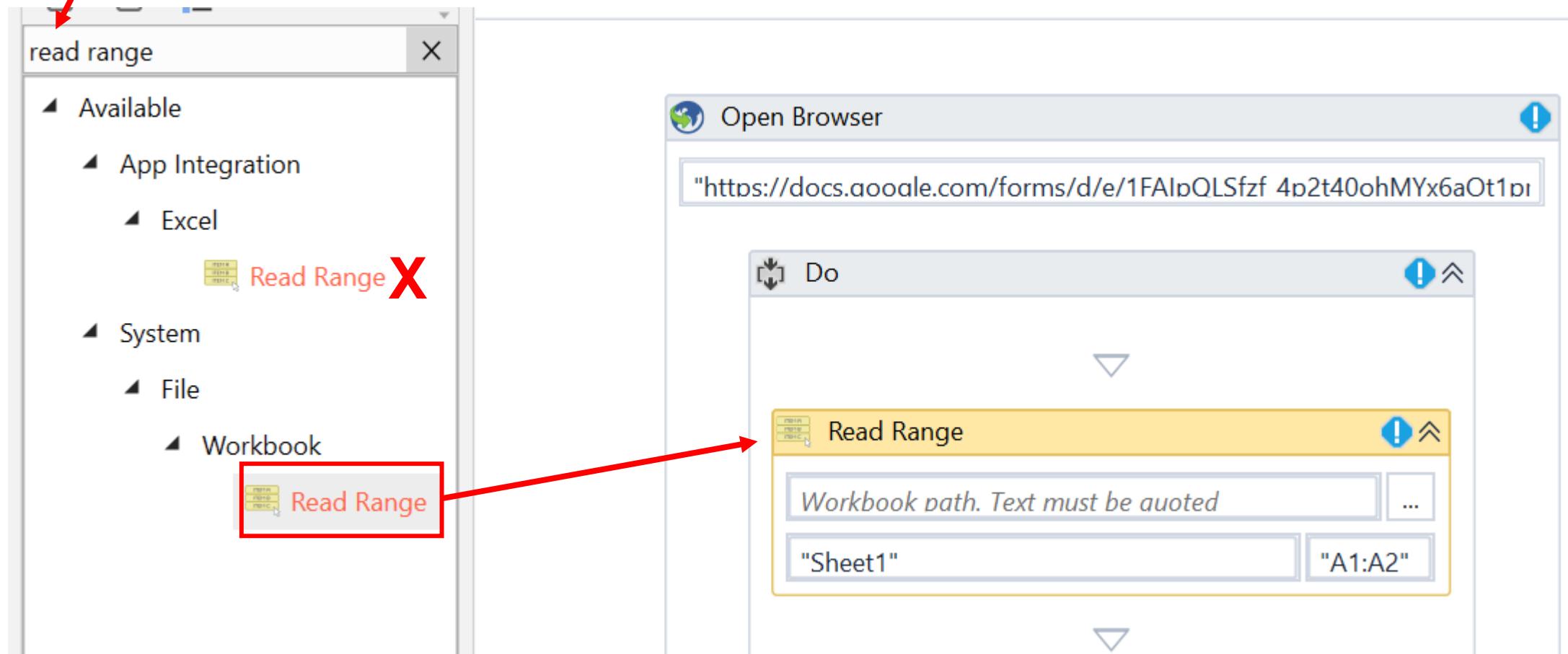
2

Copy and paste the URL of the Google form here. The URL must be quoted, ie “ ”

Exercise 5.6 (Step 4)

1

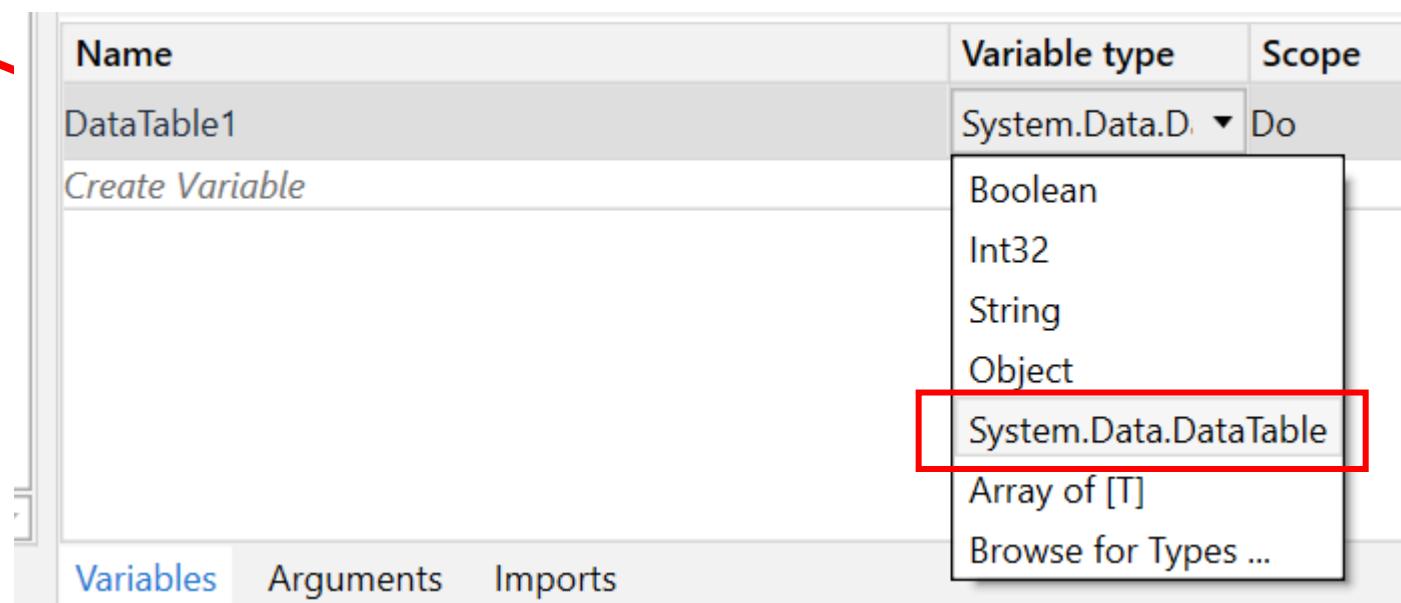
Search for “Read Range” and insert into the Do container



Exercise 5.6 (Step 5)

1

Create a variable “**DataTable1**” of type DataTable by selecting “System.Data.DataTable”



Exercise 5.6 (Step 6)

1 Specify the path of the excel file, the worksheet name and the range

2 Make sure the “AddHeaders” is checked

Common	DisplayName	Read Range
Input	Range	""
	SheetName	"NList2"
	Workbook path	"D:\DATA\EP0
Misc	Private	<input type="checkbox"/>
Options	AddHeaders	<input checked="" type="checkbox"/>
	Password	The password
	PreserveFormat	<input type="checkbox"/>
Output	DataTable	DataTable1

Open Browser

"https://docs.google.com/forms/d/e/1FAIpQLScQEaTYEU9linaWqeHnz6f"

Do

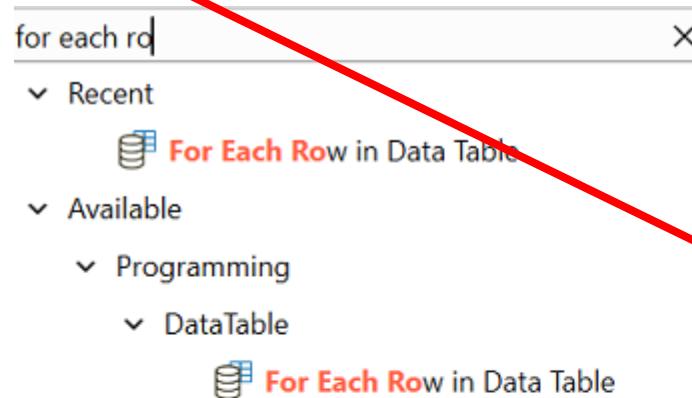
Read Range

"D:\DATA\EP0705 RPA\AY2021-2022 Sem 2\Hands" "NList2"

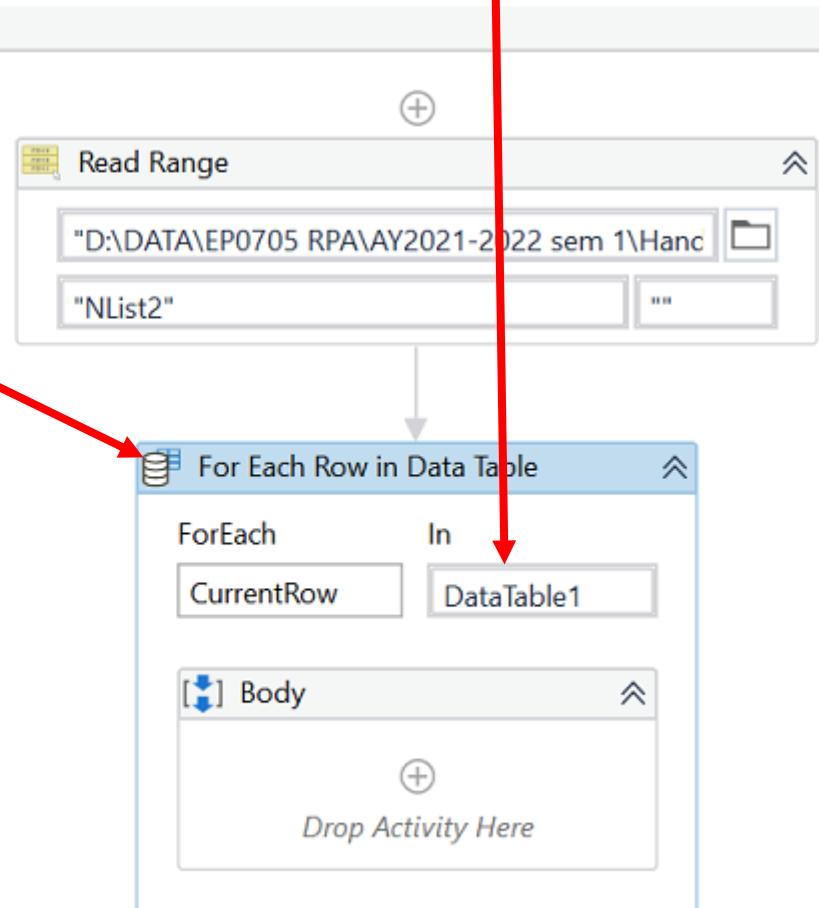
Assign “DataTable1” to the output.

Exercise 5.6 (Step 7)

1 Insert “For Each Row”



2 Type “D” and select “DataTable1”



Exercise 5.6 (Step 8)

1 Insert “Get Row Item (you can rename it to “Get Name”)

2 Click radio button Name

3 Create all the variables required

4 Type “Name” and select Name respectively

The screenshot shows the UiPath Studio interface with a workflow titled "Ex5_6 > Ex5_6 > Open Browser". The workflow starts with a "ForEach" loop over "CurrentRow". Inside the loop, there is a "Body" section containing a "Get Row Item" activity. The "Get Row Item" activity has its "Row" set to ".CurrentRow" and its "Column" set to "Name", with the value "Name" specified. A red arrow from step 1 points to the "Get Row Item" activity. A red arrow from step 2 points to the "Name" radio button in the "Column" section of the activity's configuration. A red arrow from step 3 points to a variable table at the bottom of the screen. A red arrow from step 4 points to the "Name" input field in the properties pane of the "Get Row Item" activity.

Name	Variable type	Scope	Default
Name	String	Body	Enter a VB expression
Email	String	Body	Enter a VB expression

Expand All Collapse All

UiPath.Core.Activities.GetRowItem

- Common
- Input
- Misc
- Output

DisplayName

Column

RowIndex

ColumnName

Row

Private

Value

The DataCol

The index of

"Name"

.CurrentRow

Name

In

DataTable1

124

Exercise 5.6 (Step 9)

1

Repeat “Get Row Item” for the other variables, ie Email/Job/Phone/Feedback

2

Repeat accordingly

The screenshot shows the UiPath Studio interface with a workflow and a variable table.

Workflow:

- A **Get Email** activity is selected. Its configuration shows:
 - Row: CurrentRow
 - Column: Name (radio button selected)
 - Value: Email
- An arrow points from this activity to a **Get Job** activity below it.
- On the right, the **UiPath.Core.Activities.GetRowItem** activity is expanded, showing its properties:
 - Common**: DisplayName = Get Email
 - Input**: Column = The DataCol, ColumnIndex = The index of, ColumnName = "Email", Row = CurrentRow
 - Misc**: Private = False
 - Output**: Value = Email

Variable Table:

Name	Variable type	Scope	Default
Name	String	Body	Enter a VB expression
Email	String	Body	Enter a VB expression
Job	String	Body	Enter a VB expression
Phone	GenericValue	Body	Enter a VB expression
Feedback	String	Body	Enter a VB expression
DataTable1	DataTable	Do	Enter a VB expression

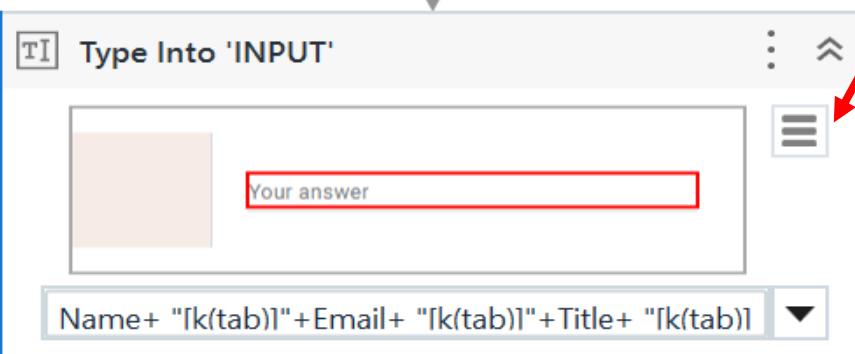
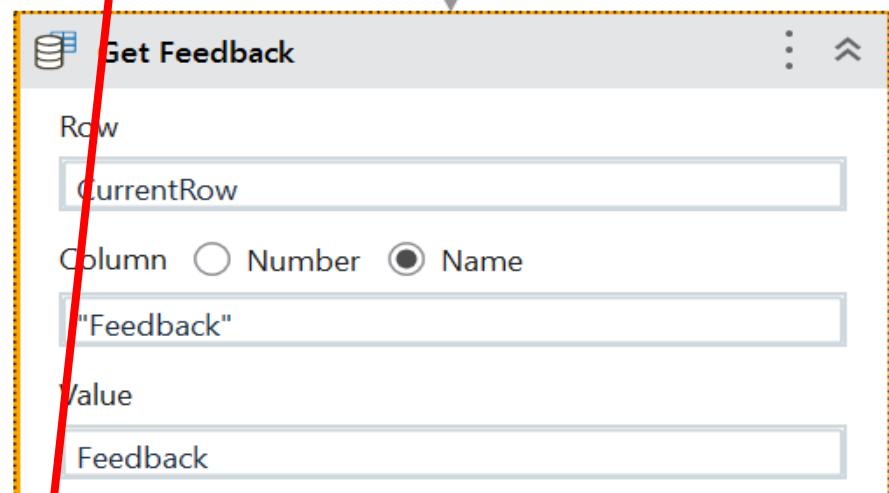
Note: A red box highlights the "Phone" row in the variable table, and a red arrow points from the "GenericValue" entry in the "Variable type" column to a callout box containing the note.

NB: The variable “Phone” must be of the variable type “GenericValue”

Exercise 5.6 (Step 10)

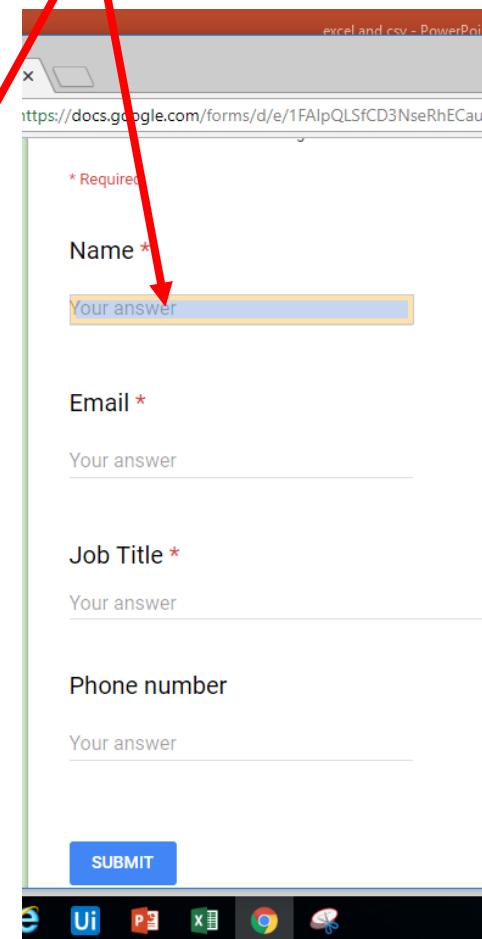
1

Insert “Type Into” (Alternatively you can copy the scripts from Exercise5.5)



2

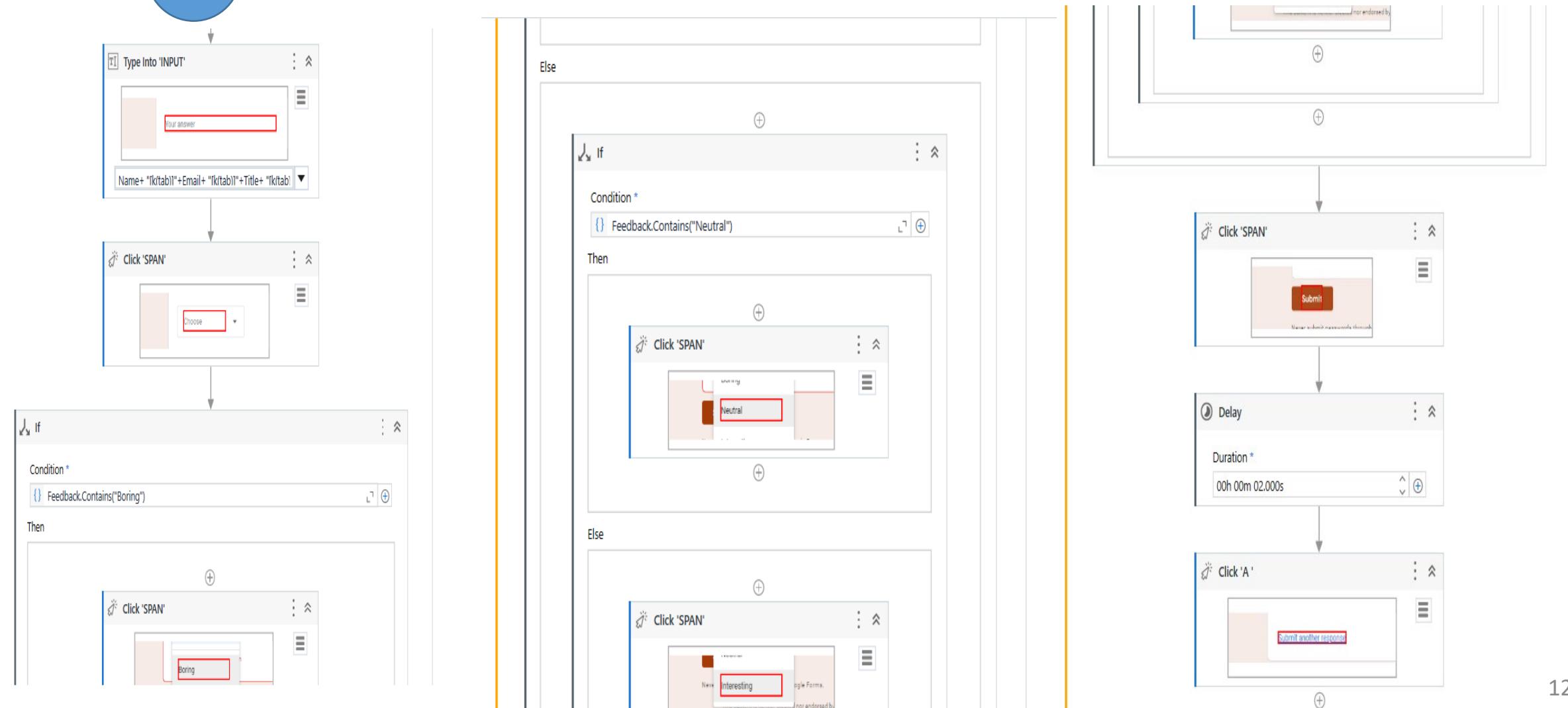
Click on “Indicate element” and select accordingly



Exercise 5.6 (Step 11)

1

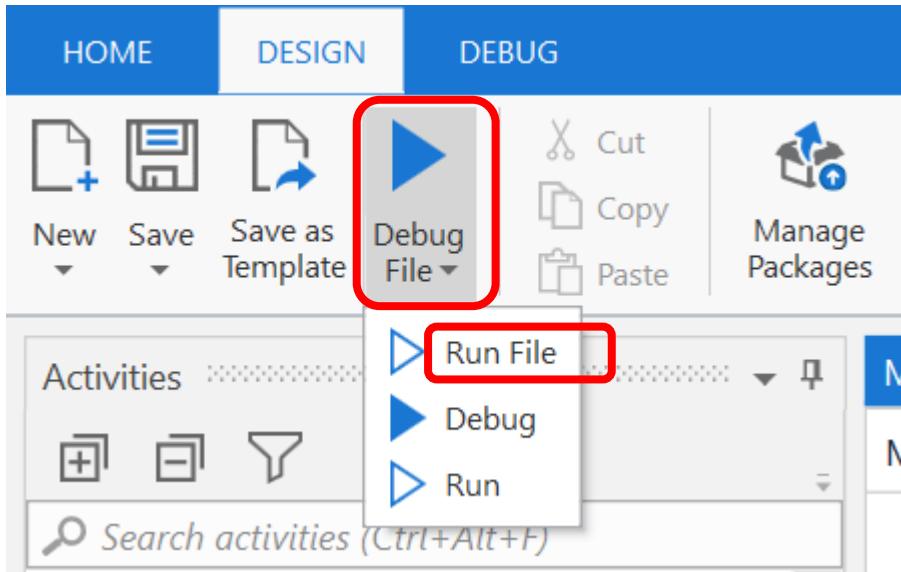
You can manually create these steps or copy them from Exercise 5.5



Exercise 5.6

Ensure that Exercise 5.xlsx is closed.

Click the **Debug File** button and select “Run File”



Exercise 5.6 – Review

How is looping with Table (Read Range), For Each Row and Get Row Item?



Ex6.1 to 6.2 Fake Name Generator

1. In UiPath Studio, create a new Process and name it “Robot3_XXXXXX” with description as “RPA Lesson5 Ex6.1 to Ex6.2”.
XXXXXXX is your first name, e.g. Jun_Hao or Jeremy etc
2. Script Robot to harvest fake name information and build an Excel database using [Write Range](#), [Get Text](#)
3. Refer to Ex6.1 in “RPA UiPath Hands-On Guide” for step-by-step instruction



FAKE ADDRESS GENERATOR

Home	US Real Address	UK Real Address
US Fake Address	CA Fake Address	UK Fake Address
Home / US Fake Address Generator		

Fake Name Generator

Exercise 6.1

Scraping Data from a Website/System

Click on “Random Address” to generate new name and particulars

Work virtually — anywhere.

Create, edit, sign, and share PDFs in Google Drive and Microsoft apps, and on any device.

Try free

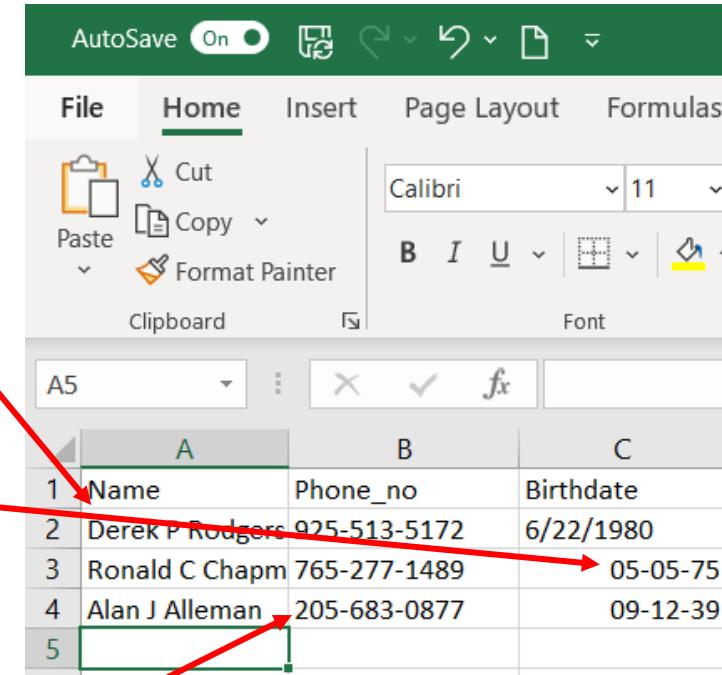
BASIC INFORMATION	
Full Name	Judy N Gonzalez
Gender	female
Title	Ms.
Race	White
Birthday	1/18/1959
Social Security Number	265-05-7827

SPONSORED SEARCHES

- get usa phone number free
- random address generator
- free credit card that works
- uk postal address

ADDRESS

Street	3590 Grand Avenue
City	Orlando
State	FL
State Full	Florida
Zip Code	32805
Phone Number	407-748-4388



The screenshot shows a Microsoft Word document with a table containing five rows of data. The columns are labeled A, B, and C. Row 1 contains headers: Name, Phone_no, and Birthdate. Rows 2 through 4 contain data: Derek P Rodgers, 925-513-5172, 6/22/1980; Ronald C Chapm, 765-277-1489, 05-05-75; and Alan J Alleman, 205-683-0877, 09-12-39. Row 5 is empty. Red arrows from the previous image point to the "Random Address" button, the birthday field, and the phone number field in the table.

A	B	C
1	Name	Phone_no
2	Derek P Rodgers	925-513-5172
3	Ronald C Chapm	765-277-1489
4	Alan J Alleman	205-683-0877
5		

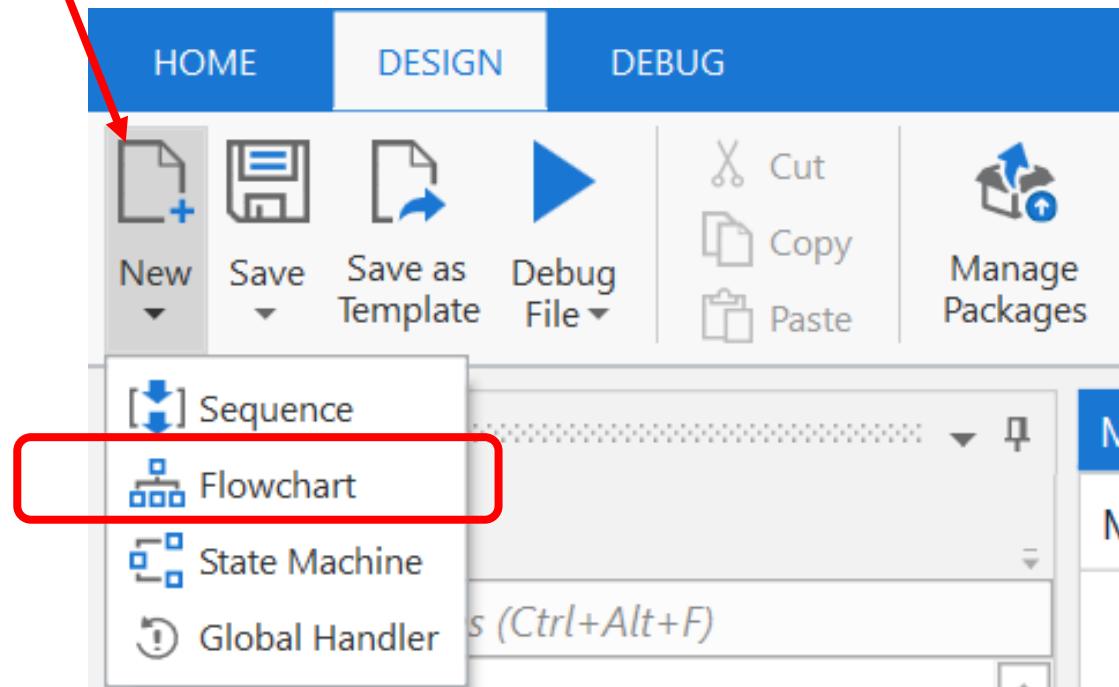
Set Up For The Exercise

- What you need:
 - Excel template, “Exercise 6”
 - Website: <https://fakenamegenerator.com/>
 - (alternate website: <https://www.fakepersongenerator.com/>)
- Others:
 - make sure the website is opened (preferably google chrome)
 - Open the excel template, “Exercise 6” and set the cursor at cell “A2”

Exercise 6.1 (Step 1)

1

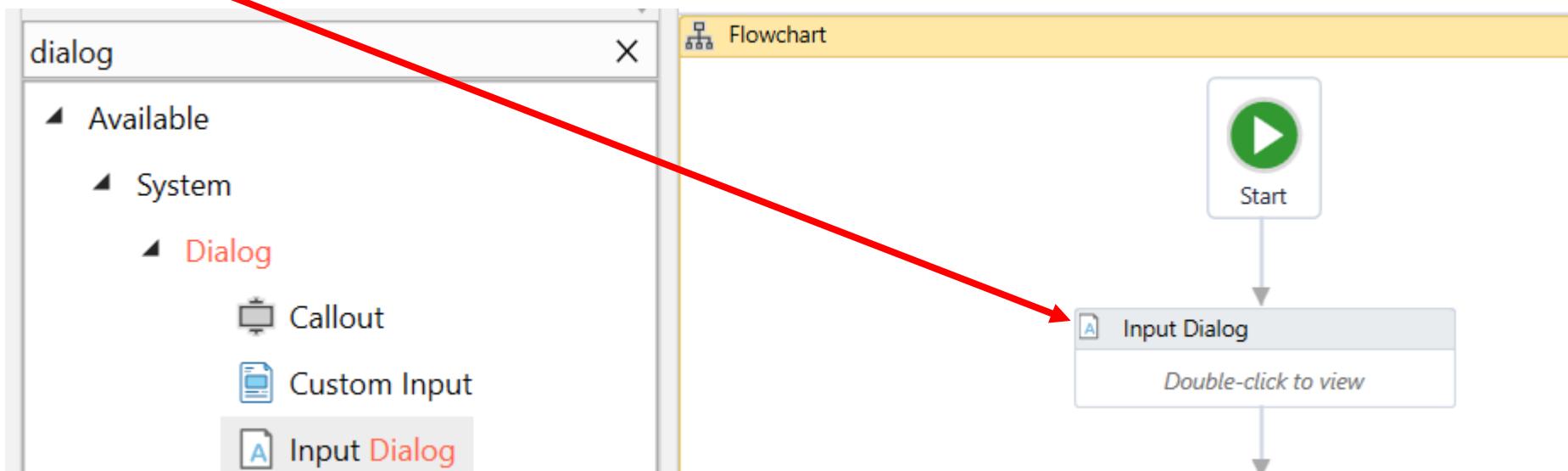
Click on New and select “Flowchart” and name it “Exercise 6.1”



Exercise 6.1 (Step 2)

1

Insert “Input Dialog”



NB: Input Dialog displays a dialog box that prompts the user with a label message and an input field. The input entered can be saved to a variable for further processing.

Exercise 6.1 (Step 3)

1

Enter “Title” and
“Label” accordingly



3

Save result to variable
“Record”

2

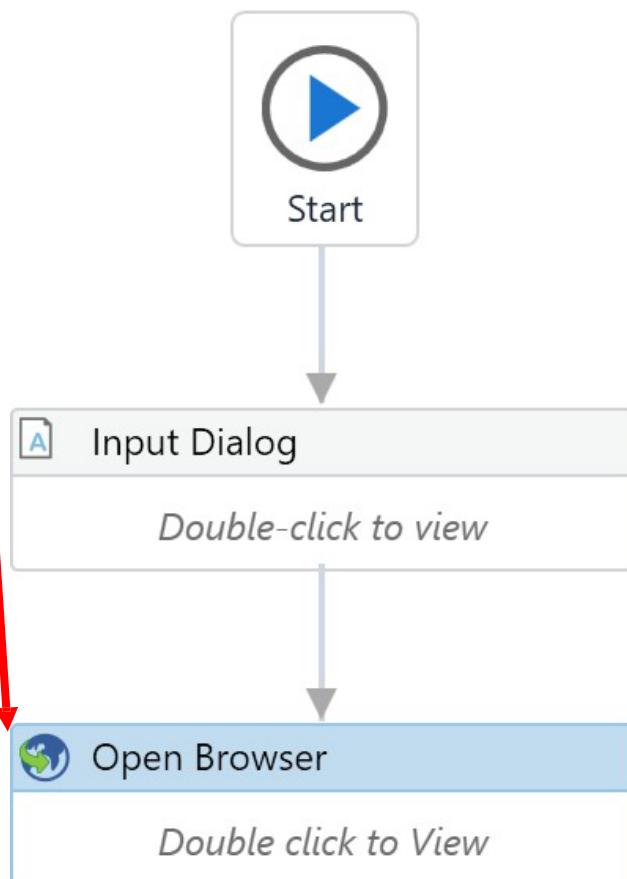
Create a variable “Record” of type Int32

Name	Variable type	Scope
Record	Int32	Ex6_1

Exercise 6.1 (Step 4)

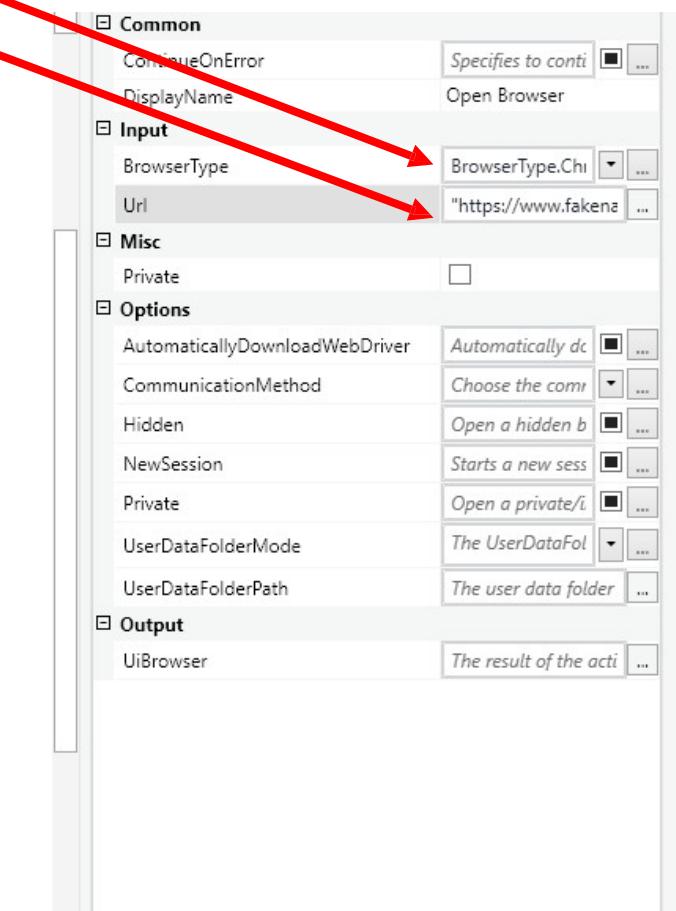
1

Insert “Open Browser”

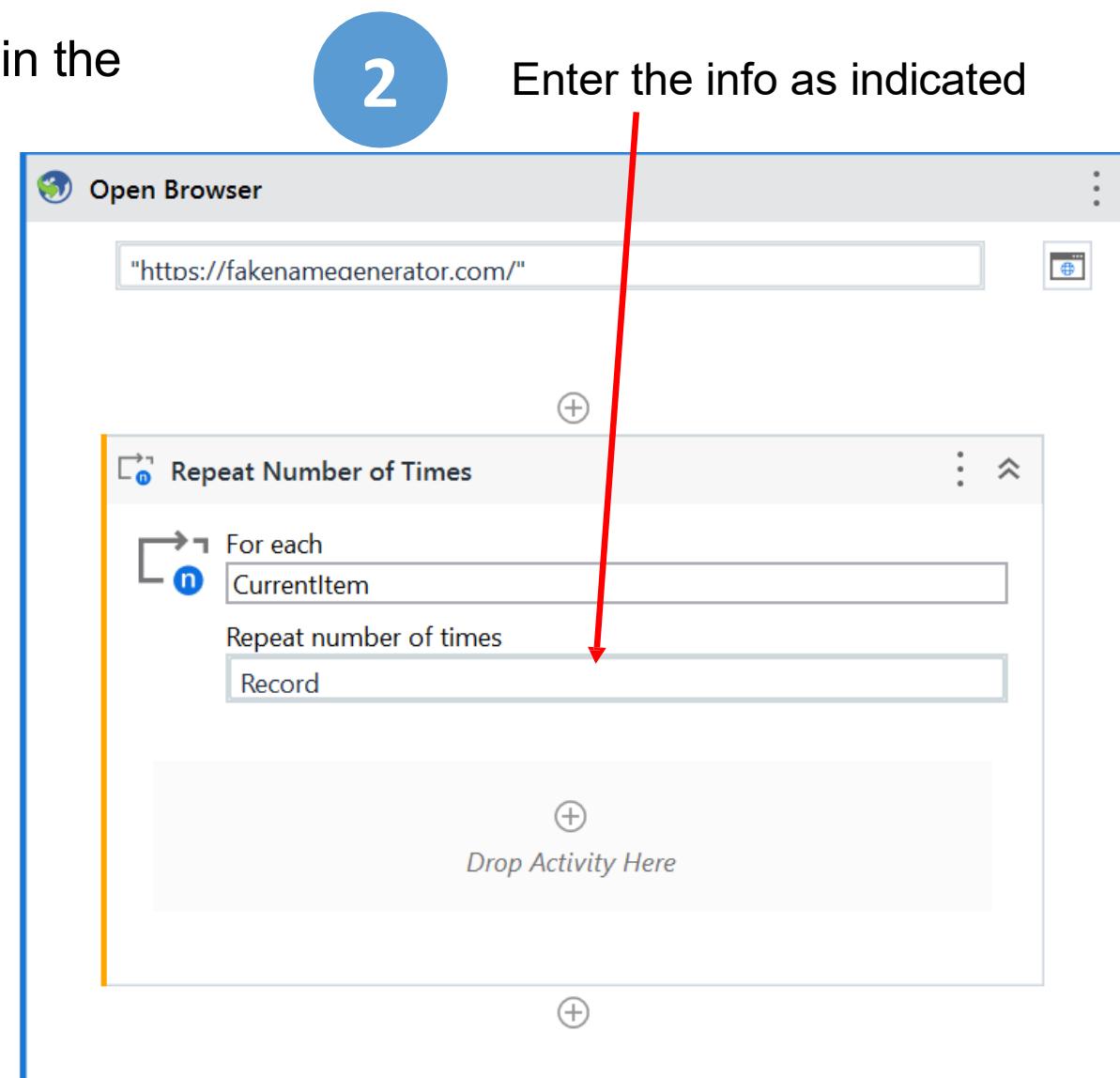
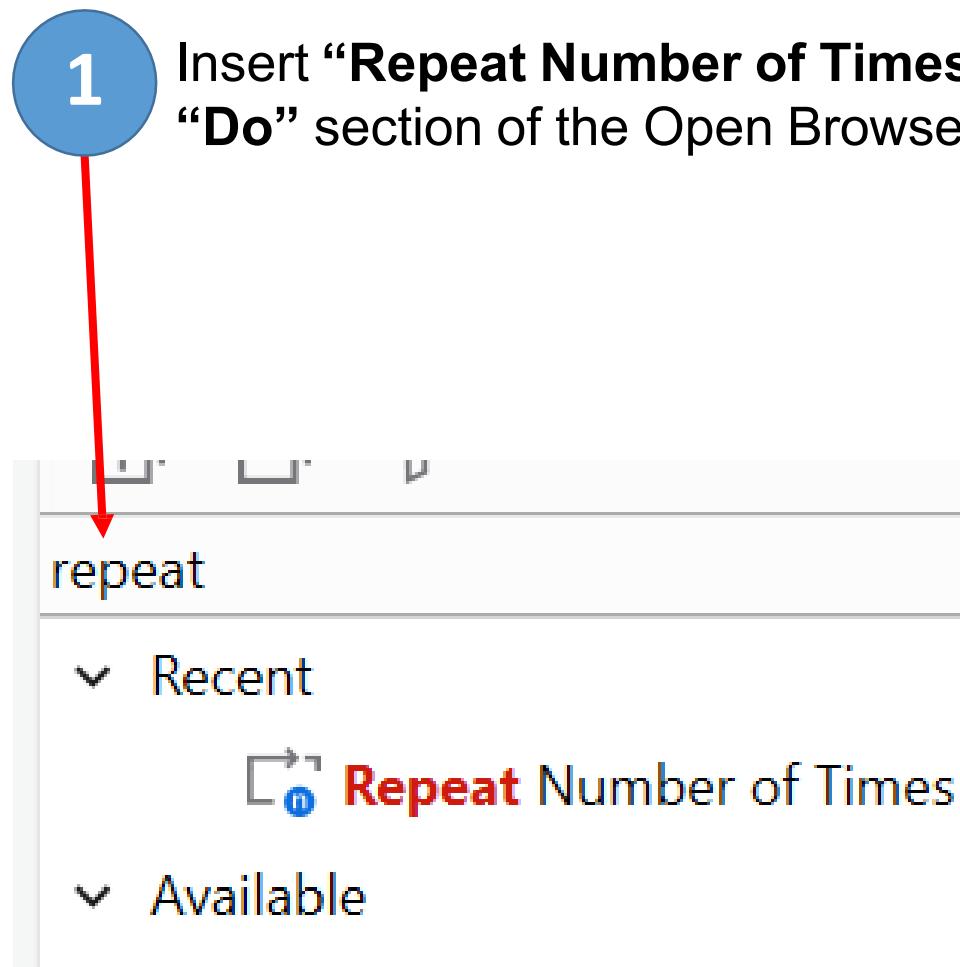


2

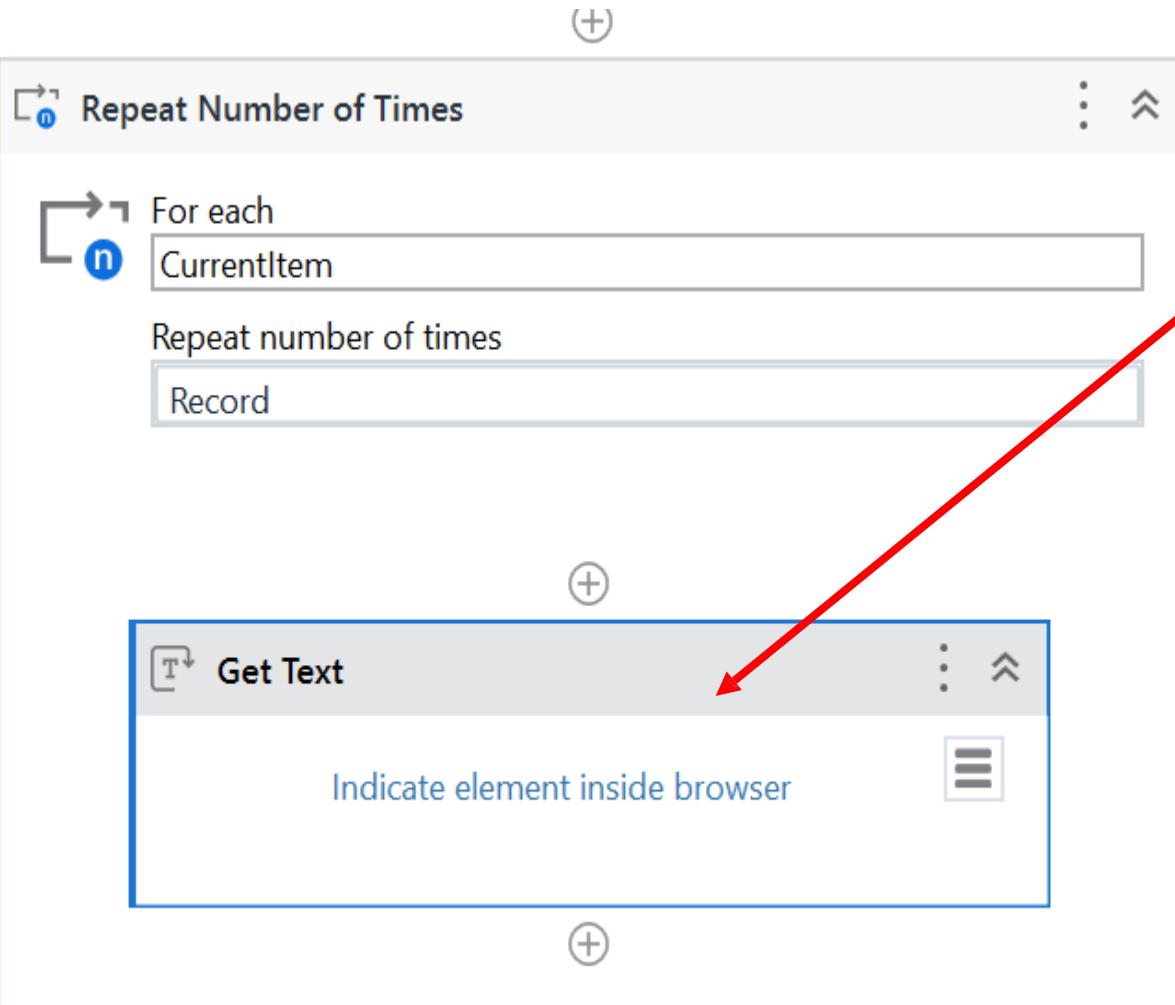
Change browser type to “Chrome” (if you are using chrome browser and copy and paste Url accordingly.)



Exercise 6.1 (Step 5)



Exercise 6.1 (Step 6)



Insert a “**Get Text**” and follow the next slide

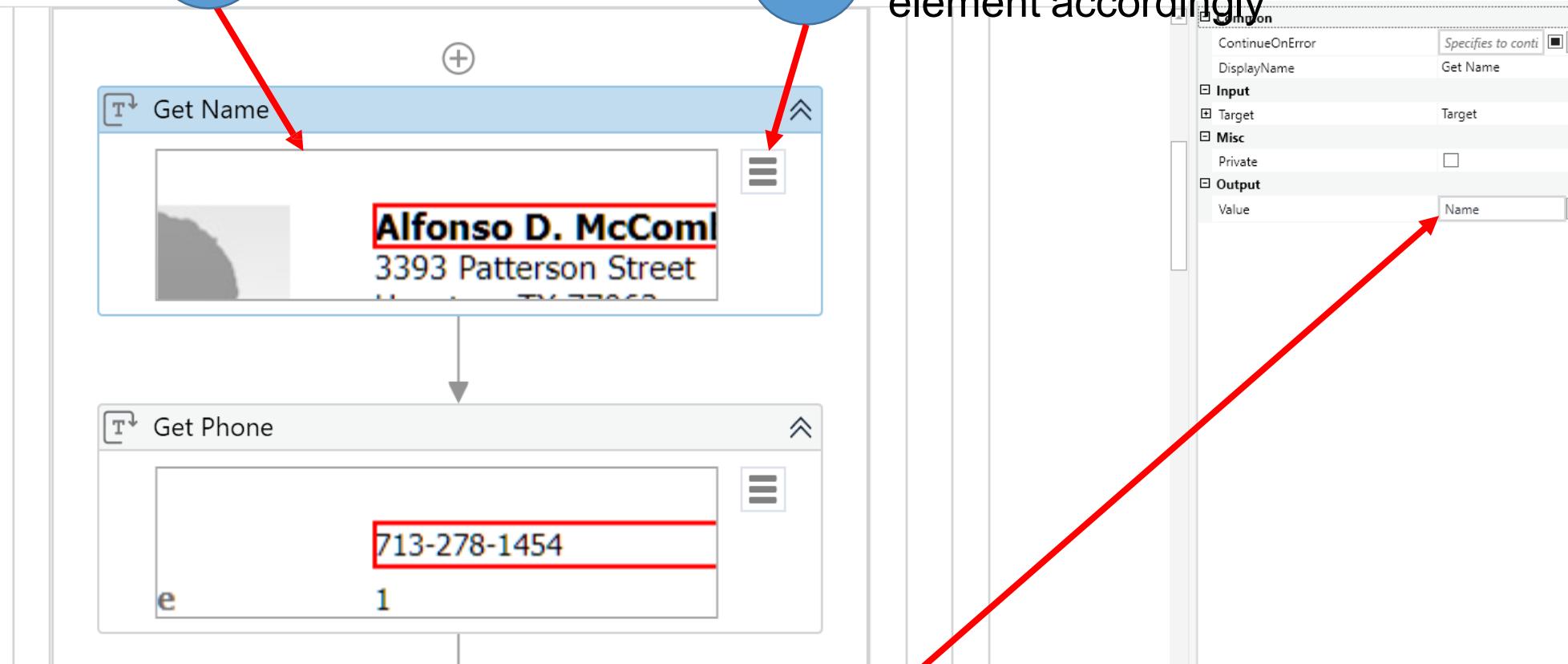
Exercise 6.1 (Step 7)

1

rename to “Get Name”

2

Select “indicate on screen” and click on element accordingly



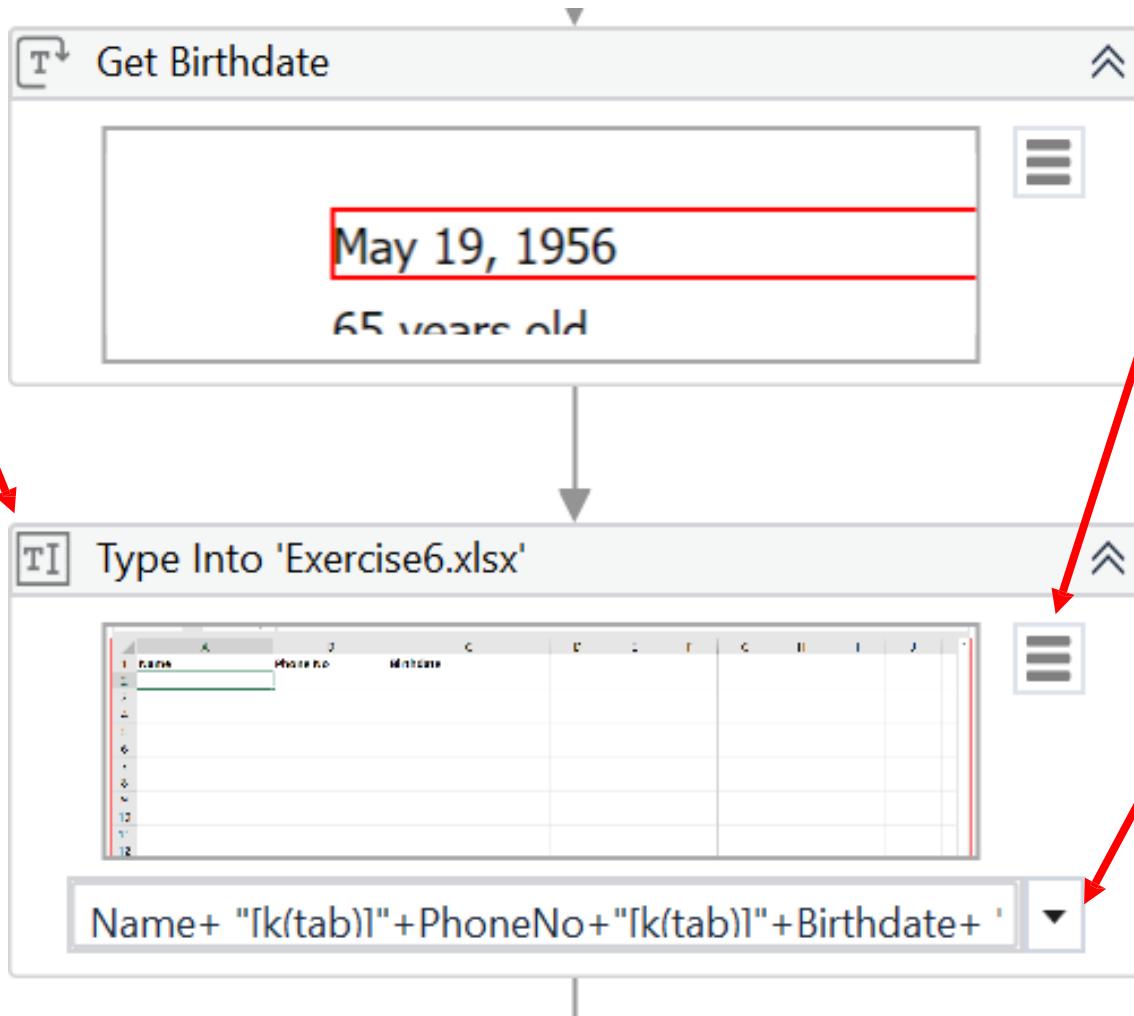
3

Create a variable “Name” and Output value to variable “Name”. Repeat for Phone no and Birthdate.

Exercise 6.1 (Step 8)

Insert
“Type into”

1



2

Select “indicate on screen” and capture the excel screen. Input the variable “Name” into the excel file at cell A2.

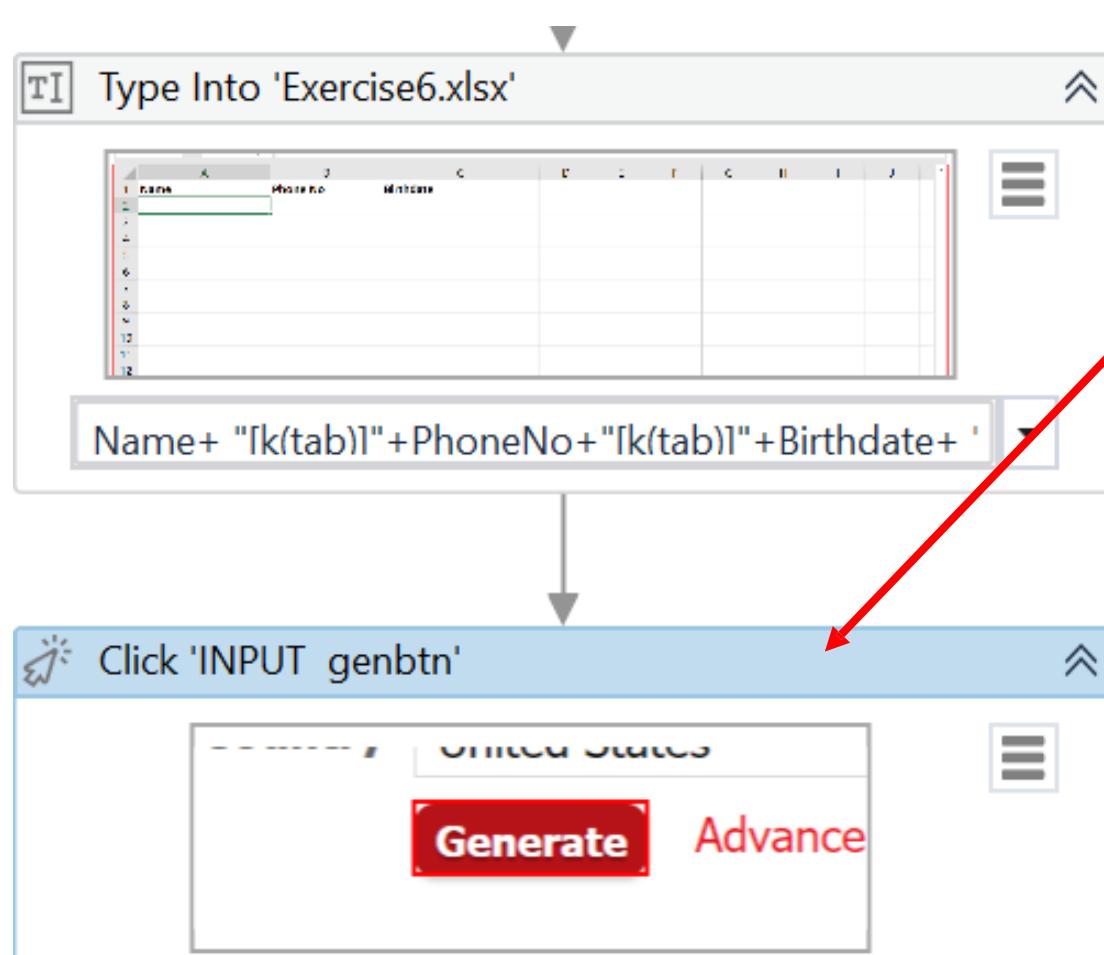
3

Indicate Name and add the “[k(tab)]” using the special key list, ie click on the drop down arrow.

Repeat until you have the below:

Name+ "[k(tab)]"+PhoneNo+[k(tab)]+
Birthdate+ "[k(home)][k(down)]"

Exercise 6.1 (Step 9)

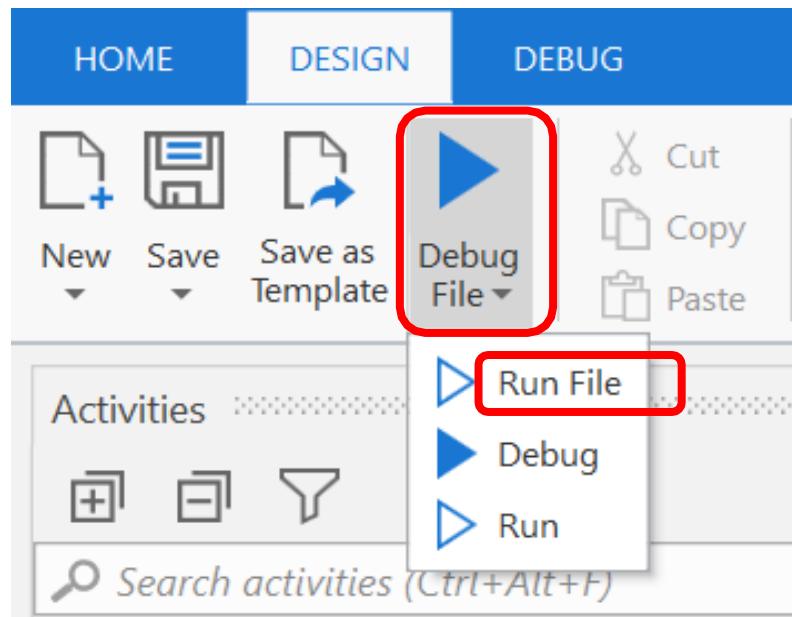


1

Insert “click” and select “indicate on screen” to click on the “Generate” button, so that the next set of info can be generated.

Exercise 6.1

Click the **Debug File** button and select “Run File”



Exercise 6.1 – Review

**Is there an alternative/better way
to complete the exercise?**

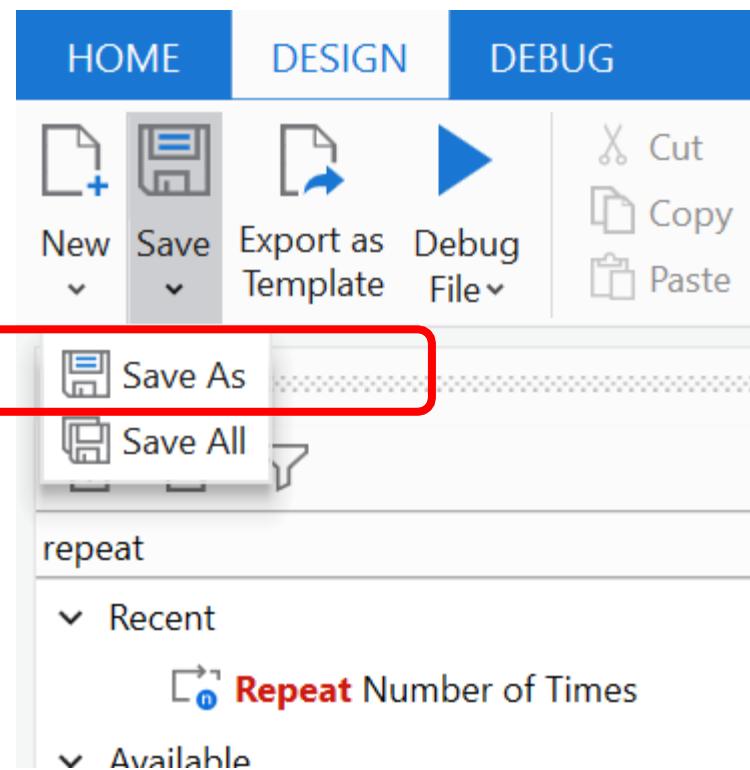
Fake Name Generator

Exercise 6.2

Exercise 6.2 (Step 1)

1

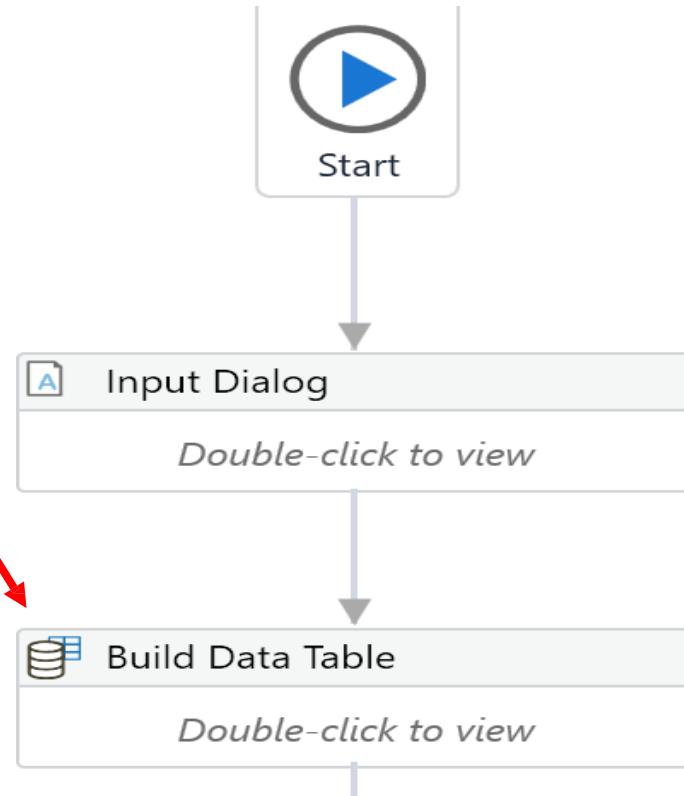
Save “Exercise 6.1” as “Exercise 6.2”



Exercise 6.2 (Step 2)

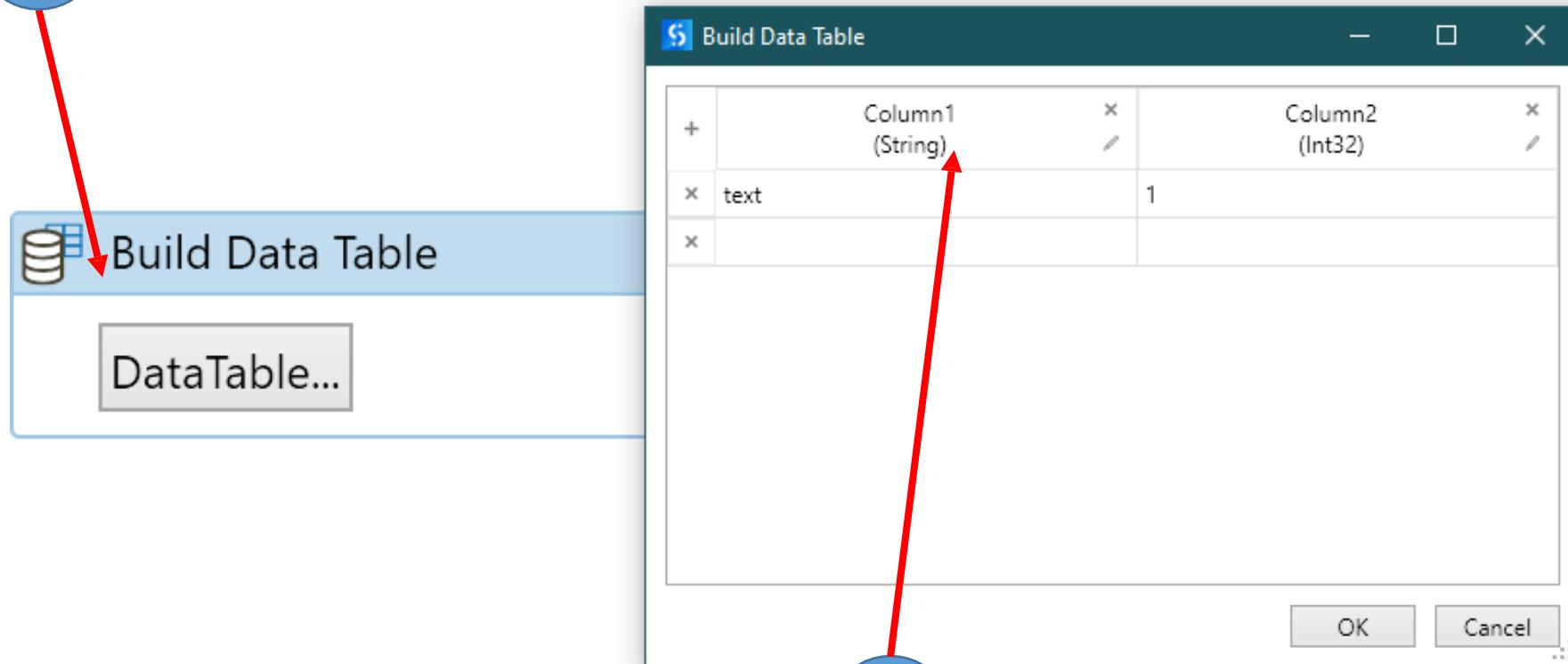
1.

Insert “Build Data Table” after Input Dialog



Exercise 6.2 (Step 3)

1. Inside the “Build Data Table”, click on the DataTable.

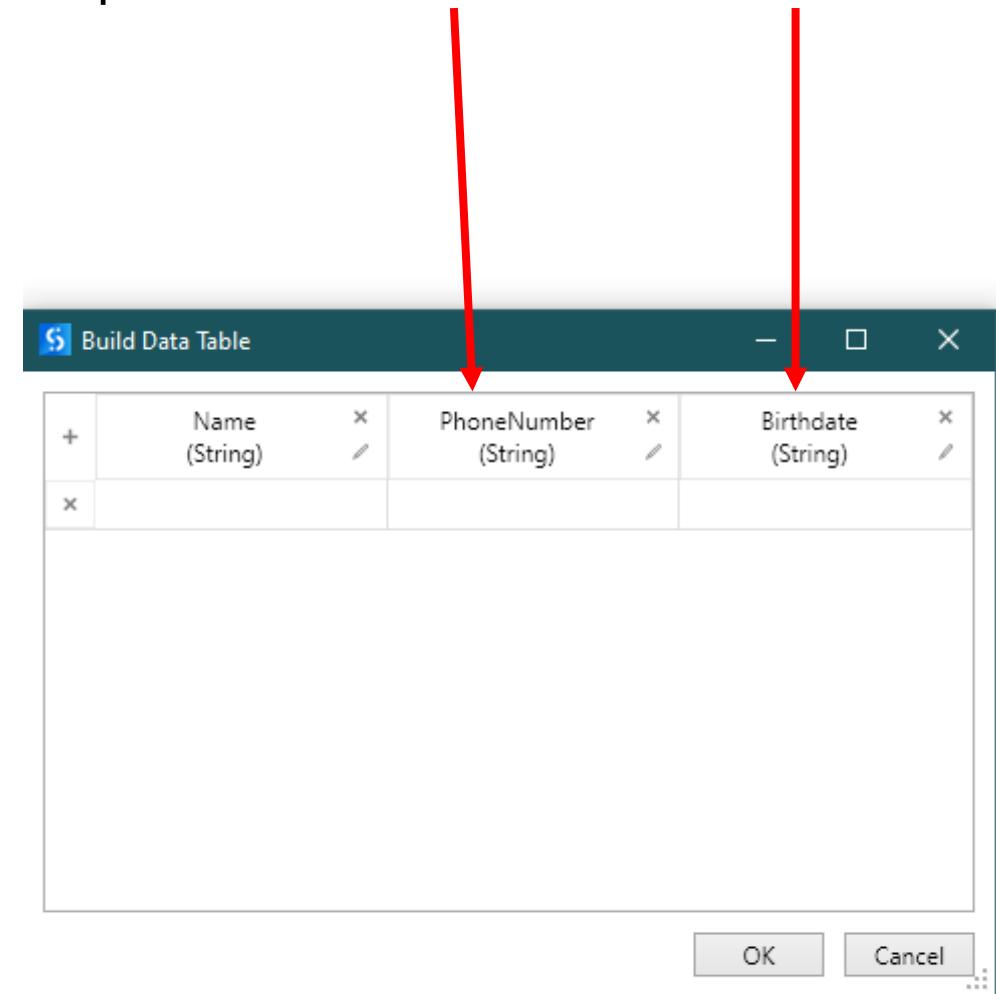
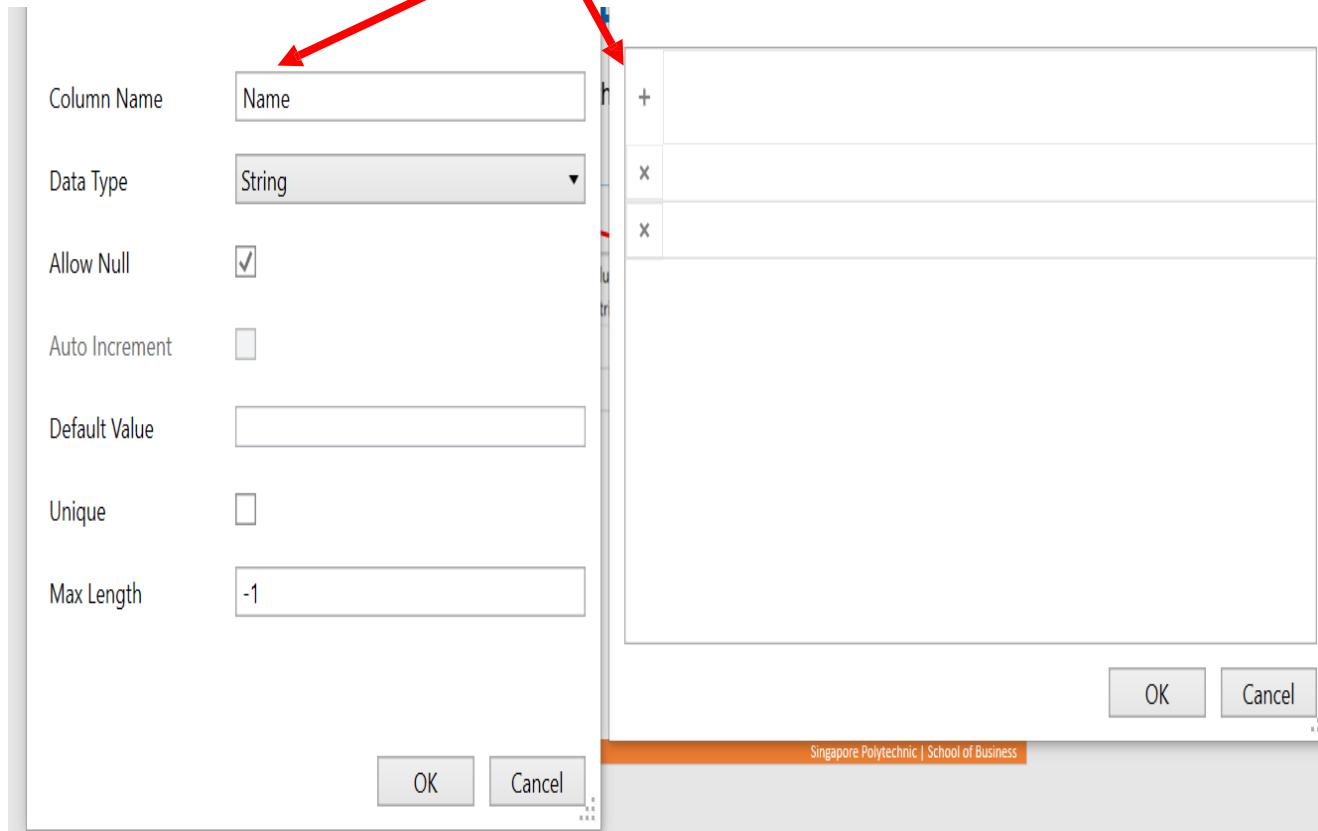


2. Remove all the existing columns and row.

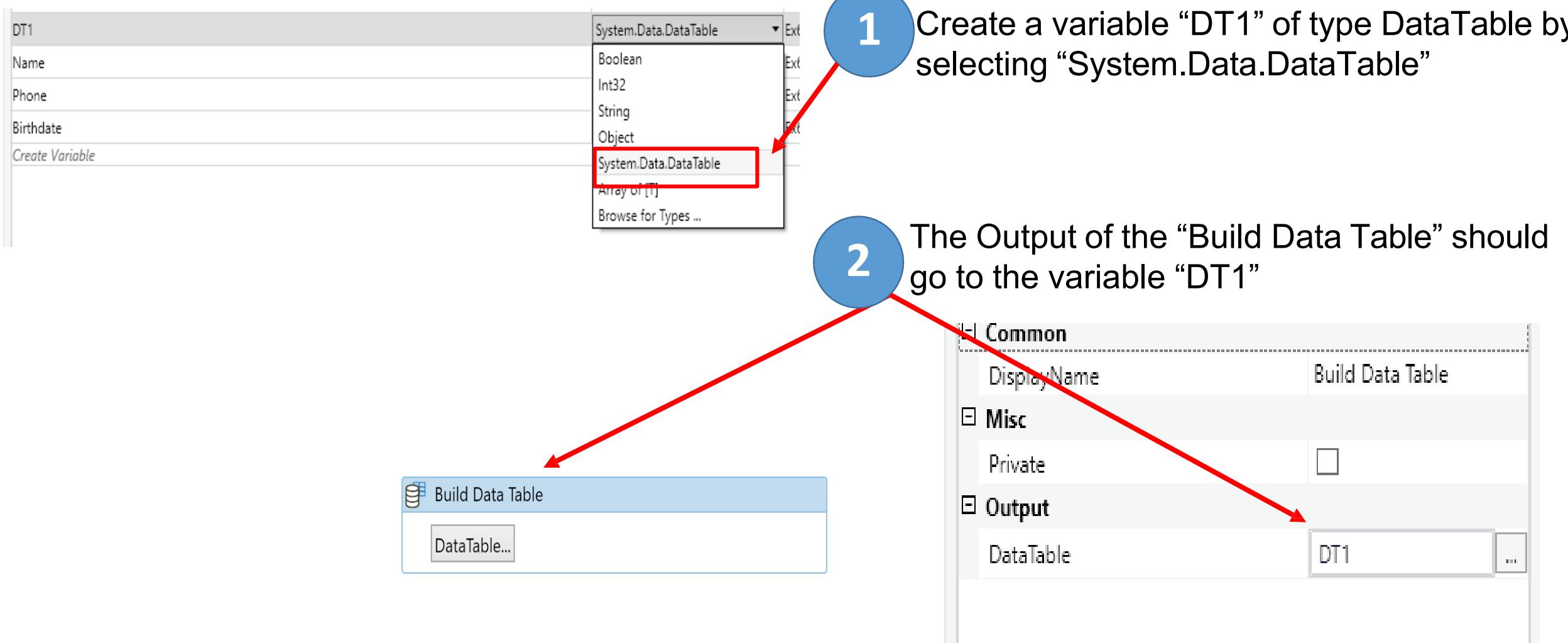
Exercise 6.2 (Step 4)

1

Click on “+” and type in “Name” for Column Name. Repeat for the Phone and Birthdate



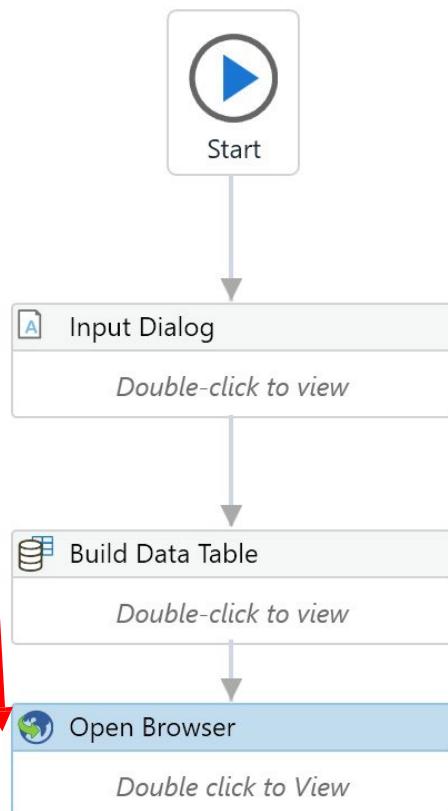
Exercise 6.2 (Step 5)



Exercise 6.2 (Step 6)

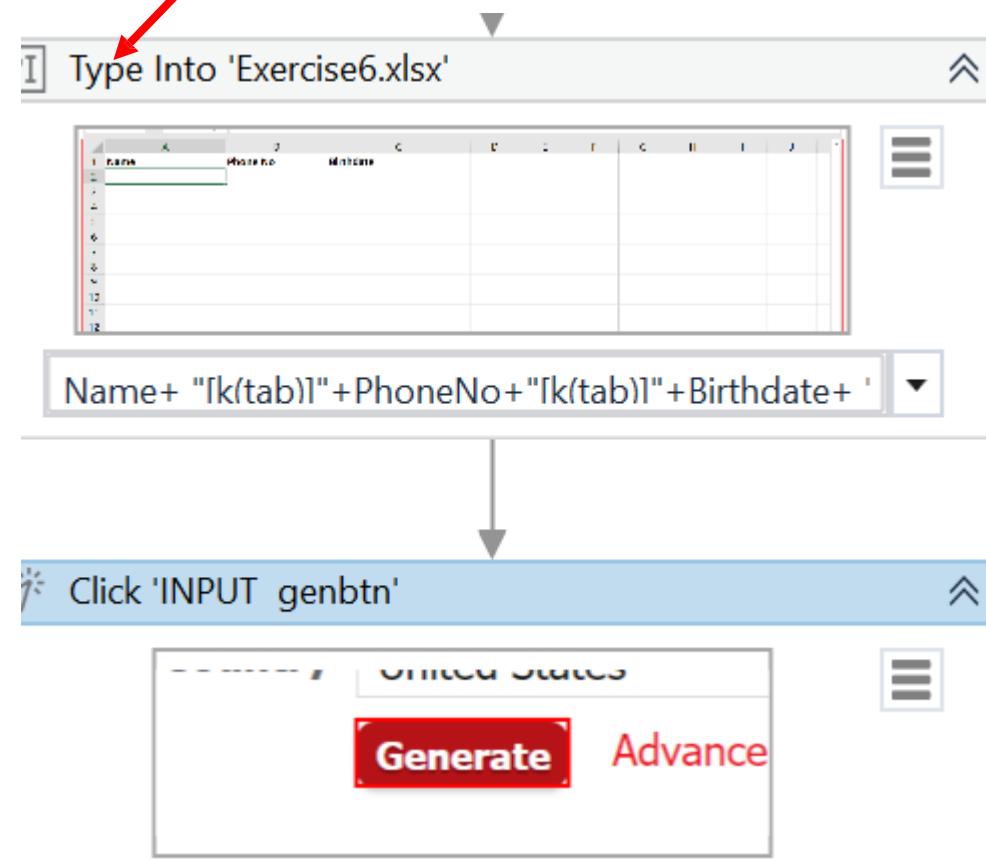
1

Ensure the Open Browser is connected.



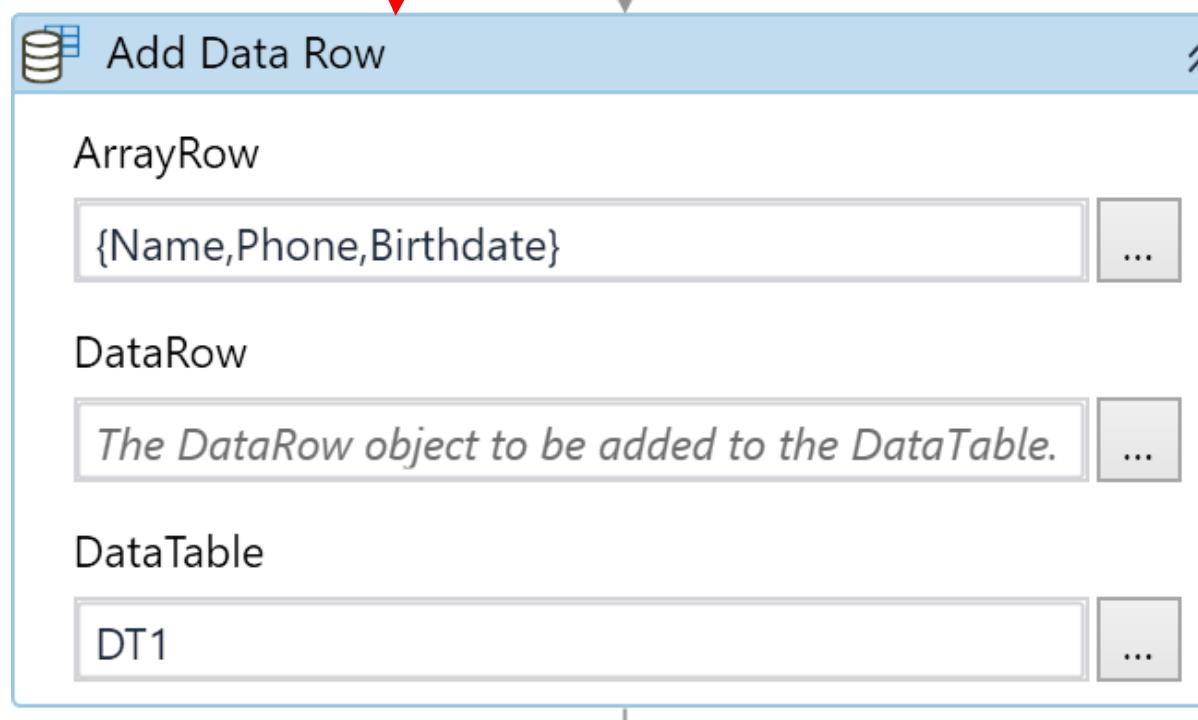
2

Remove this with excel screen. It's no longer required since we are working with the excel file closed.

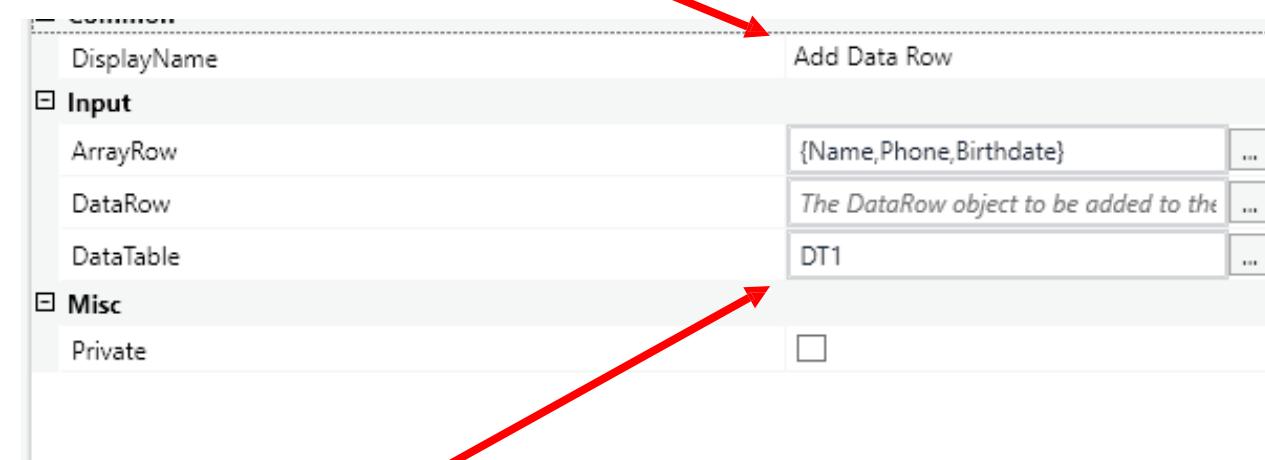


Exercise 6.2 (Step 7)

1. Insert “Add Data Row” after the click on “Generate”.



2. Add in the variable using Curly bracket: {Name,Phone,Birthdate} in the ArrayRow

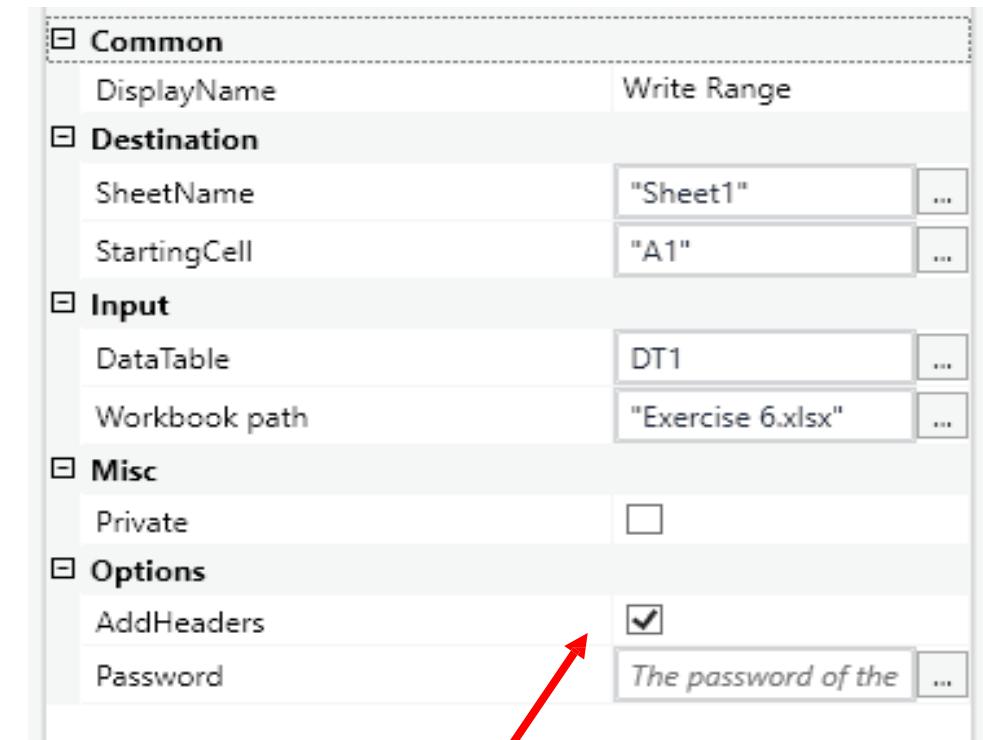
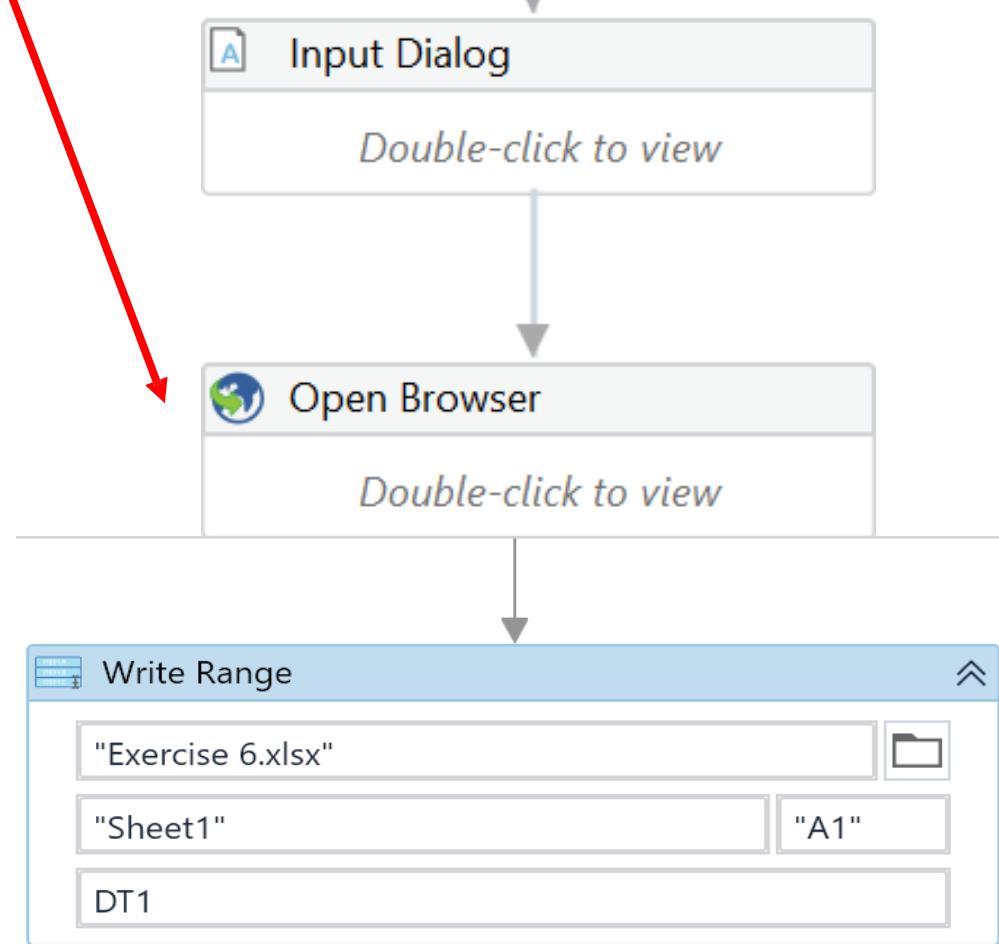


3. Enter “DT1” for DataTable. This will ensure that every row of data get saved into the DataTable “DT1”

Exercise 6.2 (Step 8)

1

Insert “Write Range” **Outside** the Open Browser and fill in the info accordingly



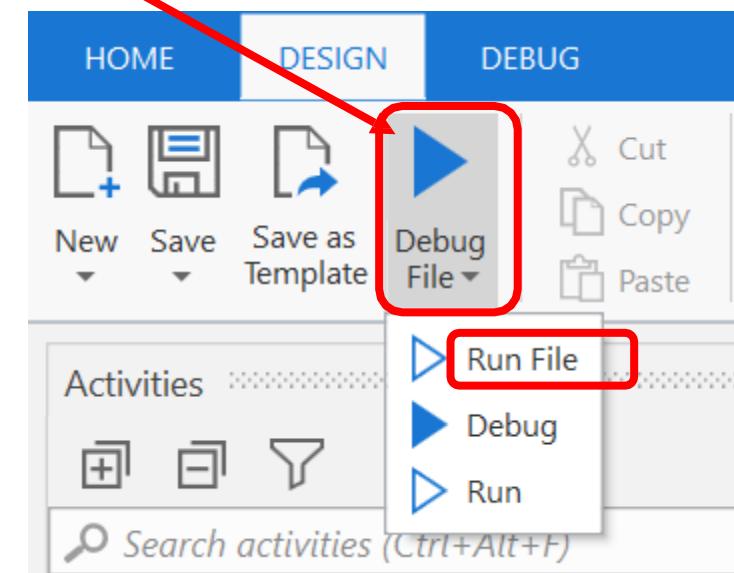
2

Check “AddHeaders”

Exercise 6.2

Click the **Debug File** button and select “Run File”

NB: Excel and google chrome webpage should be closed before you run the program.





Ex7.1 Multiple Excel File

1. Create a new Process and name it “*Robot4_XXXXXX*” with description as “*RPA Lesson6 Ex7.1 to Ex7.2*”.
XXXXXX is your first name, e.g. *Jun_Hao* or *Jeremy* etc
2. Script Robot to consolidate 5 Excel files to submit to Google Form using For Each File in a Folder (ex 7.1) and Using a Masterfile where the 5 different files are appended to (“Append Range” Ex 7.2)
3. Refer to Ex7.1 to 7 in “RPA UiPath Hands-On Guide” for step-by-step instruction



A screenshot of a Google Form titled "RPA Form 1". The form includes fields for Name (with placeholder "Your answer"), Email (with placeholder "Your answer"), Job Title (with placeholder "Your answer"), and Phone Number (with placeholder "Your answer"). A "Submit" button is at the bottom.

Exercise 7.1

Processing Multiple Excel Files

Exercise 7.1

This exercise shows how to process multiple excel files to extract data for input into a Google form.

Name	Email	Job Title	Phone number	Feedback	
Alvin Ng	Alvin@gmail.com	Lecturer	63216781	Boring	
Bee Leng	Beeleng@gmail.com	Finance Manager	90001892	Interesting	
Name David	david@yahoo.com	Business Manager	56777777	Neutral	
F2_Alv	Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
F2_Be	Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring
Name F2_David	david@yahoo.com	Business Manager	56777777	Neutral	
F3_Alv	F2_Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
F3_Be	F2_Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring
Name F3_David	david@yahoo.com	Business Manager	56777777	Neutral	
F4_Alv	F3_Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
F4_Be	F3_Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring
Name F4_David	david@yahoo.com	Business Manager	56777777	Neutral	
F5_Alv	F4_Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
F5_Be	F4_Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring
F5_David	david@yahoo.com	Business Manager	56777777	Neutral	
F5_Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting	
F5_Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring	



RPA Form 2
RPA Training Form 2
**Required*

Name * Your answer
Email * Your answer
Job Title * Your answer
Phone Number * Your answer
Feedback * Choose
Submit

Exercise 7.1

What you need:

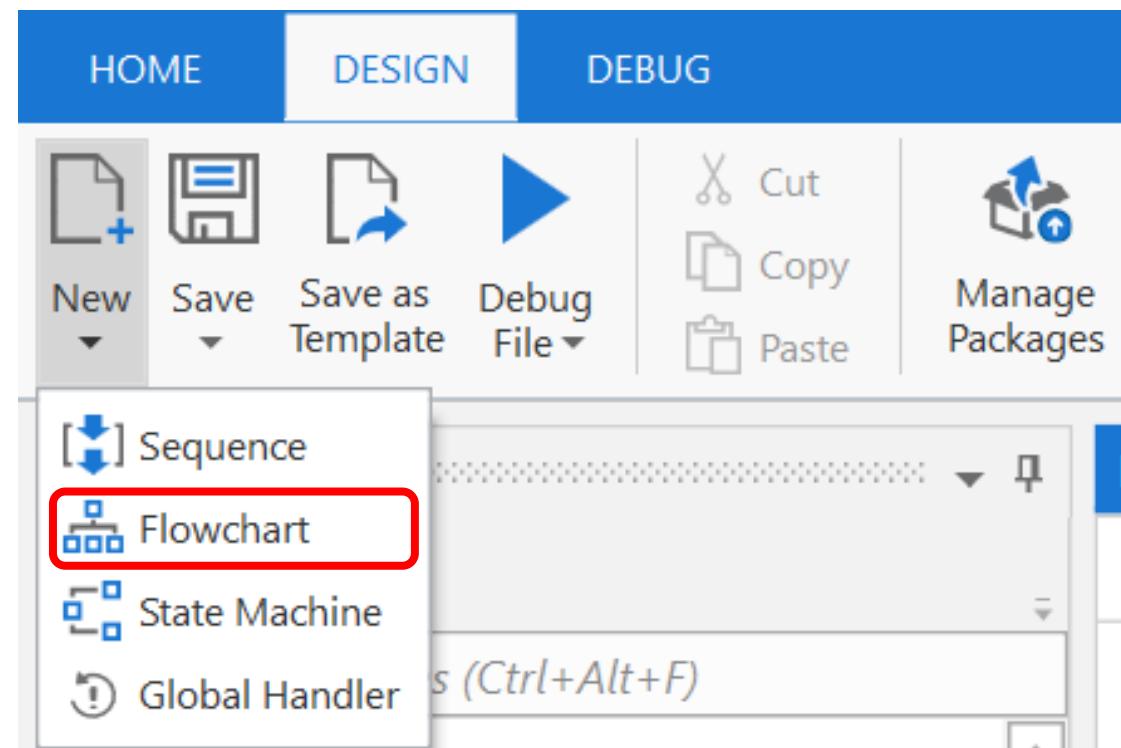
- Folder with 5 Excel files, File_01 to File_05
- Google form

https://docs.google.com/forms/d/e/1FAIpQL_ScQEgTYEU9linaWqeHnz6fXMFkzmF5KM_CQMSJFAOunx262hig/viewform

Recommended to use Google Chrome Browser!

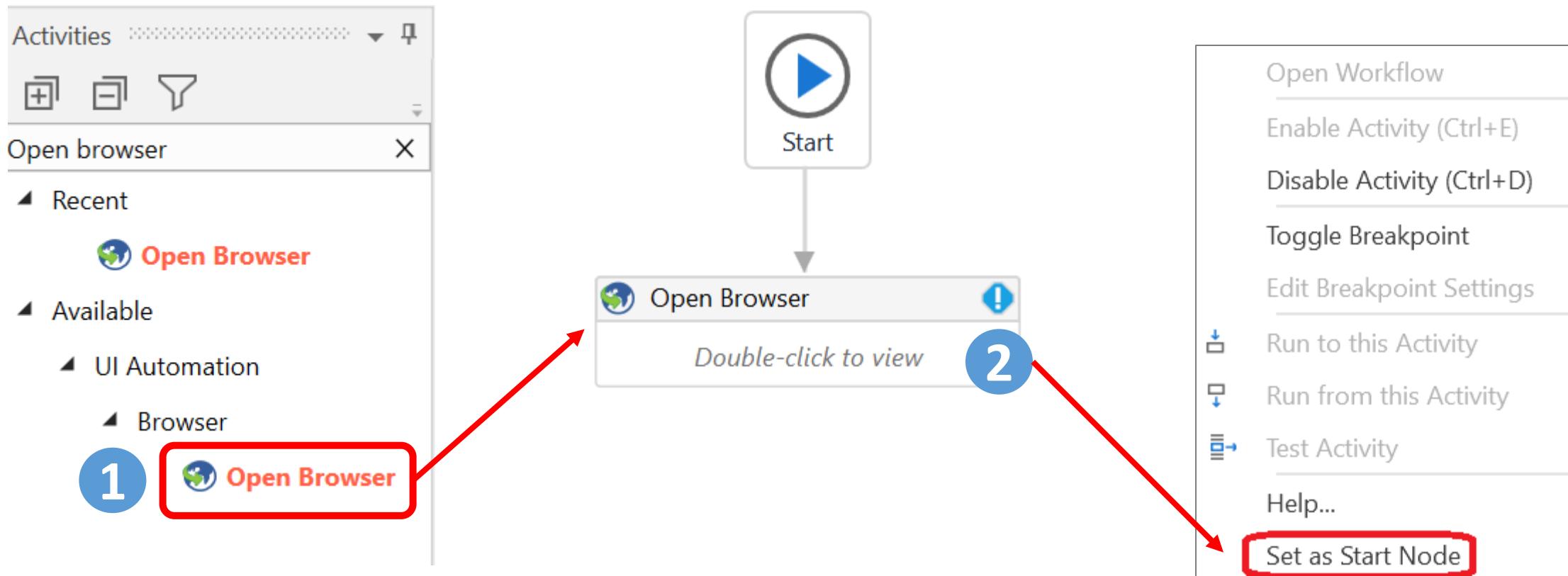
Exercise 7.1 (Step 1)

In UiPath Studio, create a new flowchart, and name it “Ex7.1”.



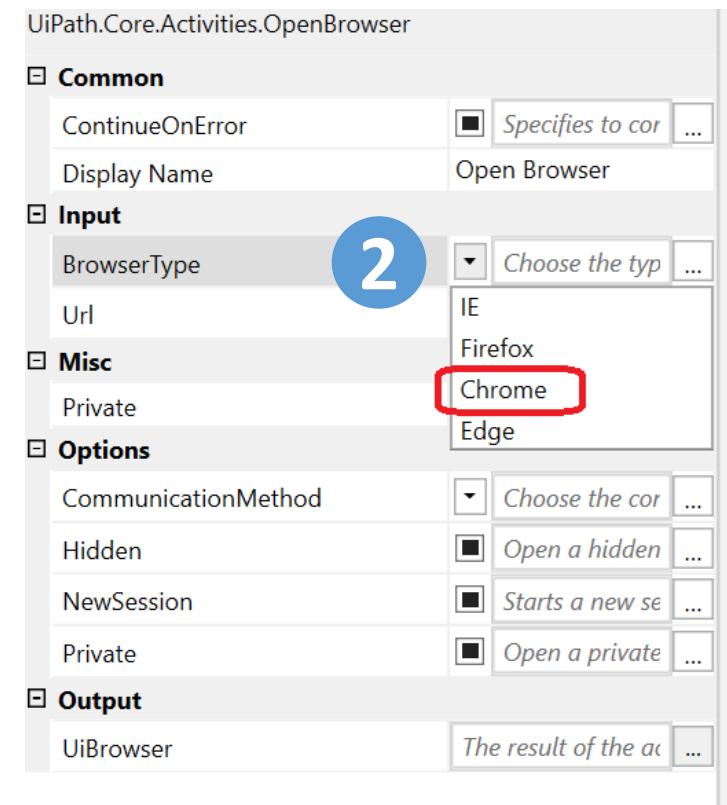
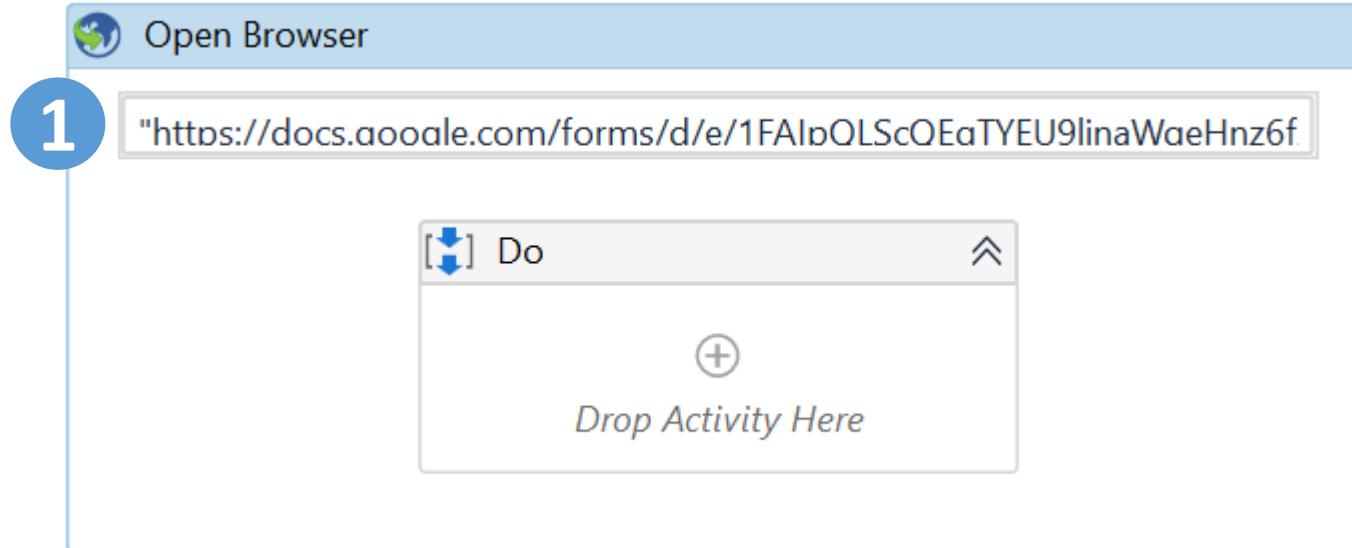
Exercise 7.1 (Step 2)

1. Search for “Open Browser” and drag into the Designer Panel.
2. Right-click on “Open Browser”, and select “Set as Start Node”.



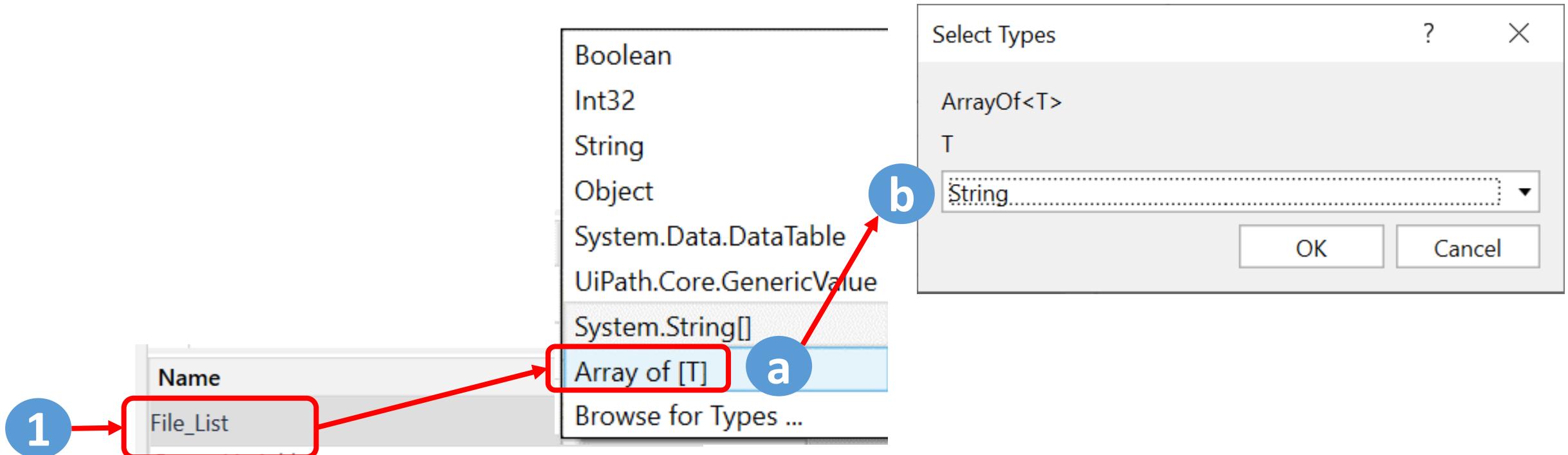
Exercise 7.1 (Step 3)

1. In the “Open Browser” window, input the URL of the Google Form.
Remember to enclose this string with double quotation marks.
2. In the Properties Panel, select “Chrome” as the browser type.



Exercise 7.1 (Step 4)

1. Create a new variable with name “**File_List**”
 - a) Click the drop-down and select Array of [T]
 - b) Choose “String” then click ok



Exercise 7.1 (Step 4)

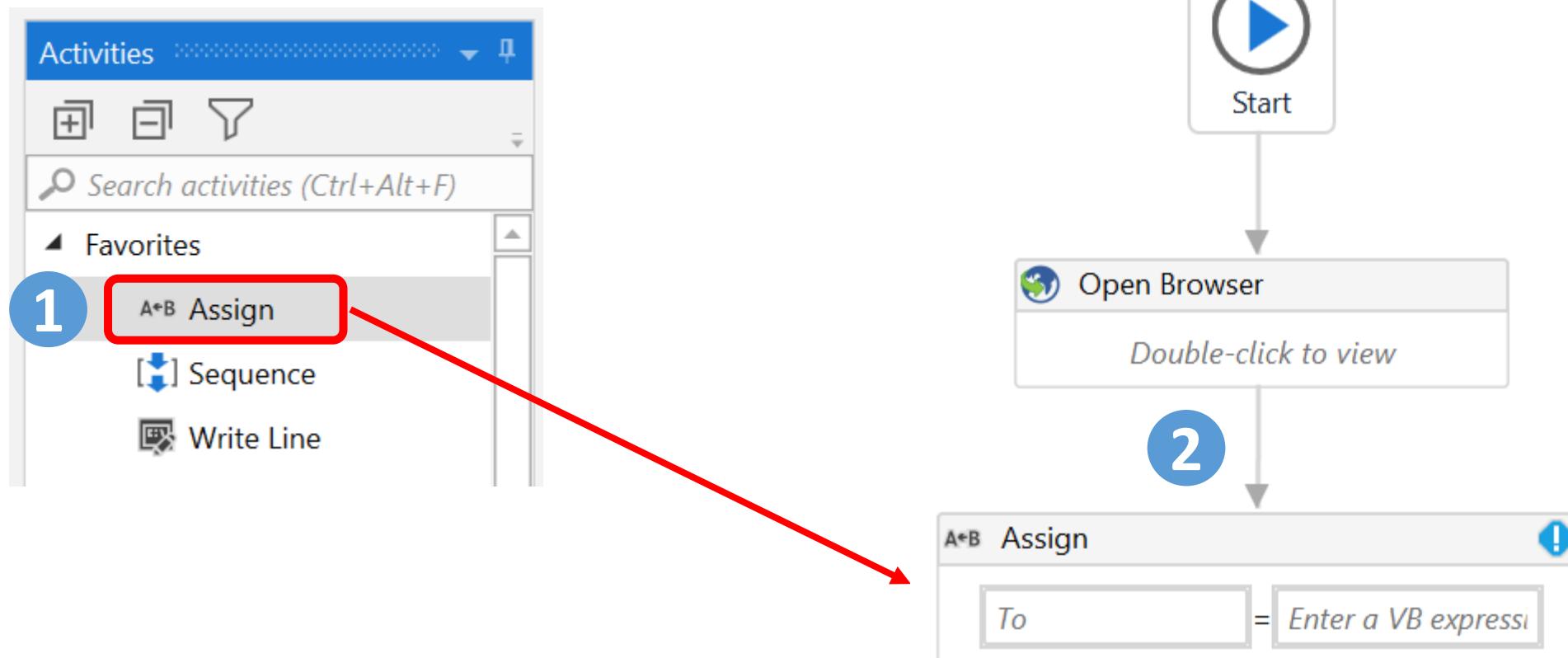
2. Create a new string variable with name “**File_Directory**”.
 - a) Input as the default value the path of the folder where the 5 Excel files, File_01 to File_05, are stored. Remember to enclose this string with double quotation marks.

2

Name	Variable type	Scope	Default
File_List	String[]	Ex1	<i>Enter a VB expression</i>
File_Directory	String	Ex1	a "D:\RPA\RPA_Intermediate\Training\Files_Multiple"
<i>Create Variable</i>			

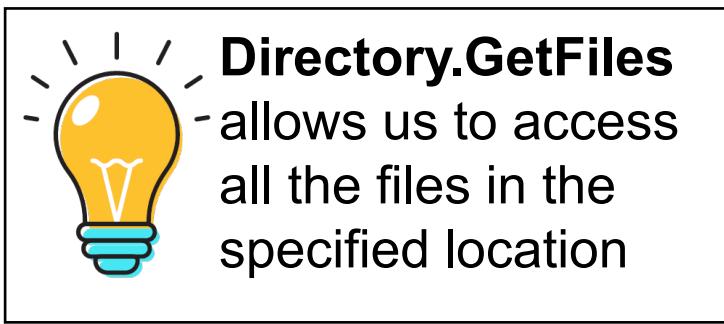
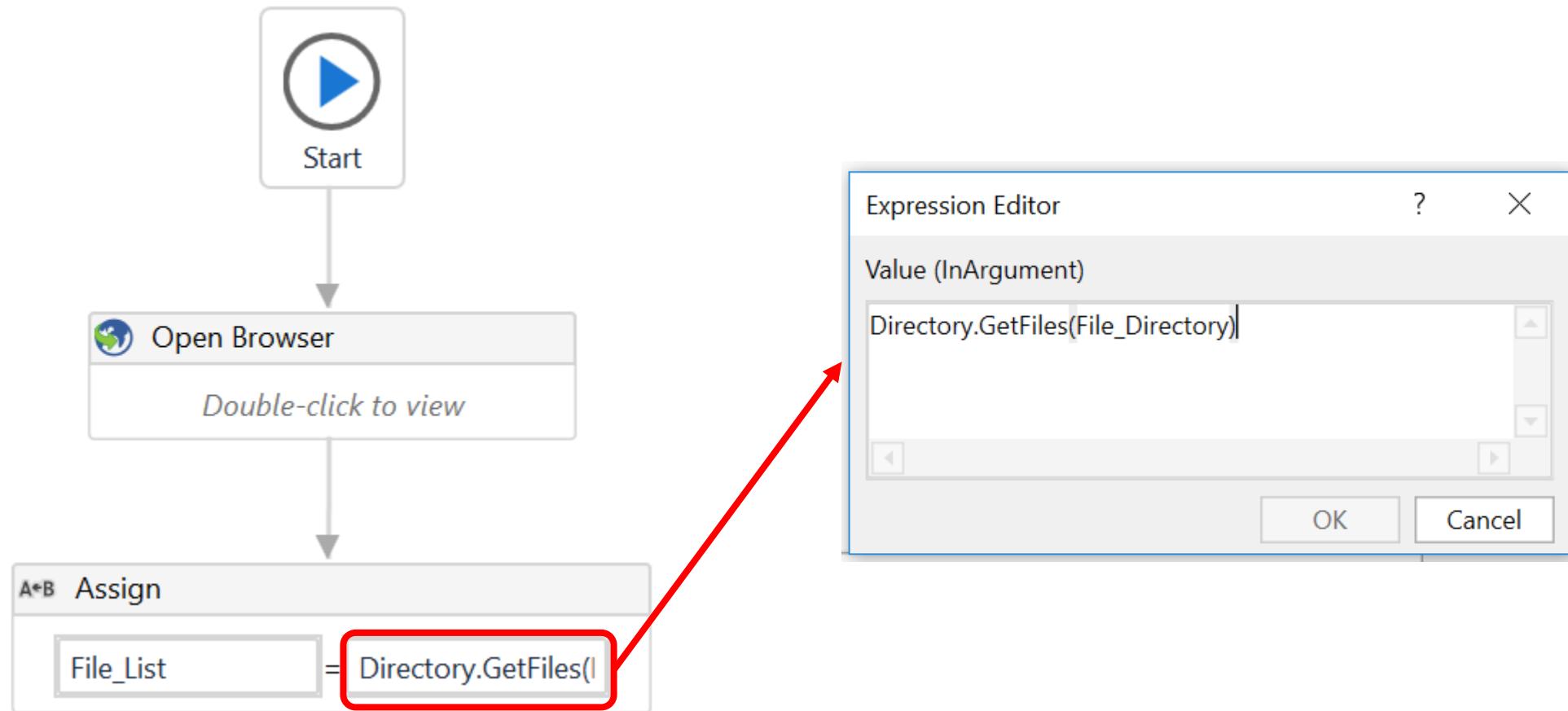
Exercise 7.1 (Step 5)

1. Search for “Assign” and drag into the Designer Panel.
2. Join “Assign” to “Open Browser”.



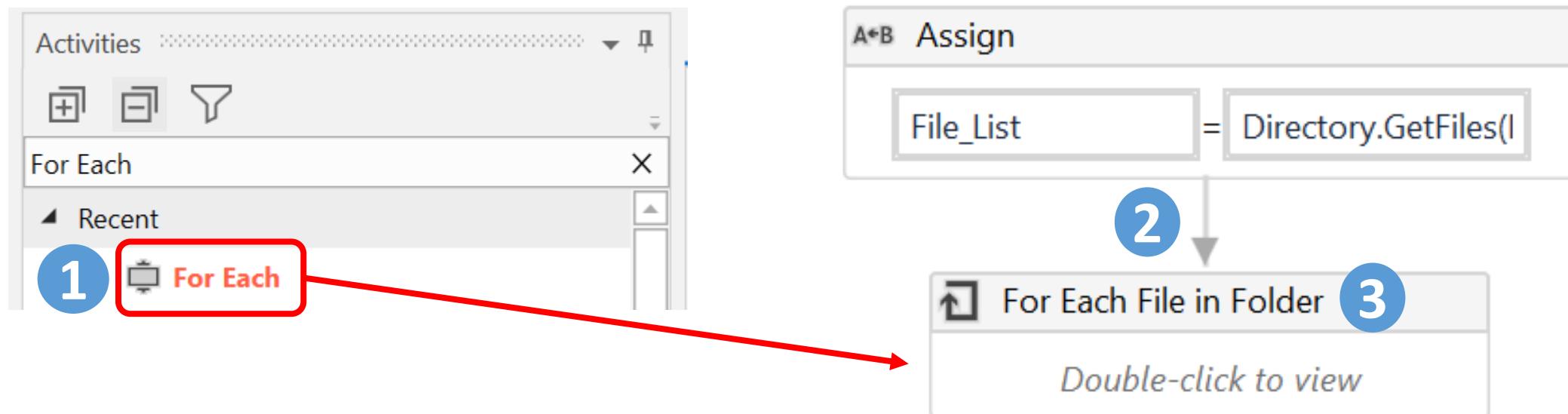
Exercise 7.1 (Step 6)

We will use “**Directory.GetFiles**” to access the folder where the 5 Excel files are stored.



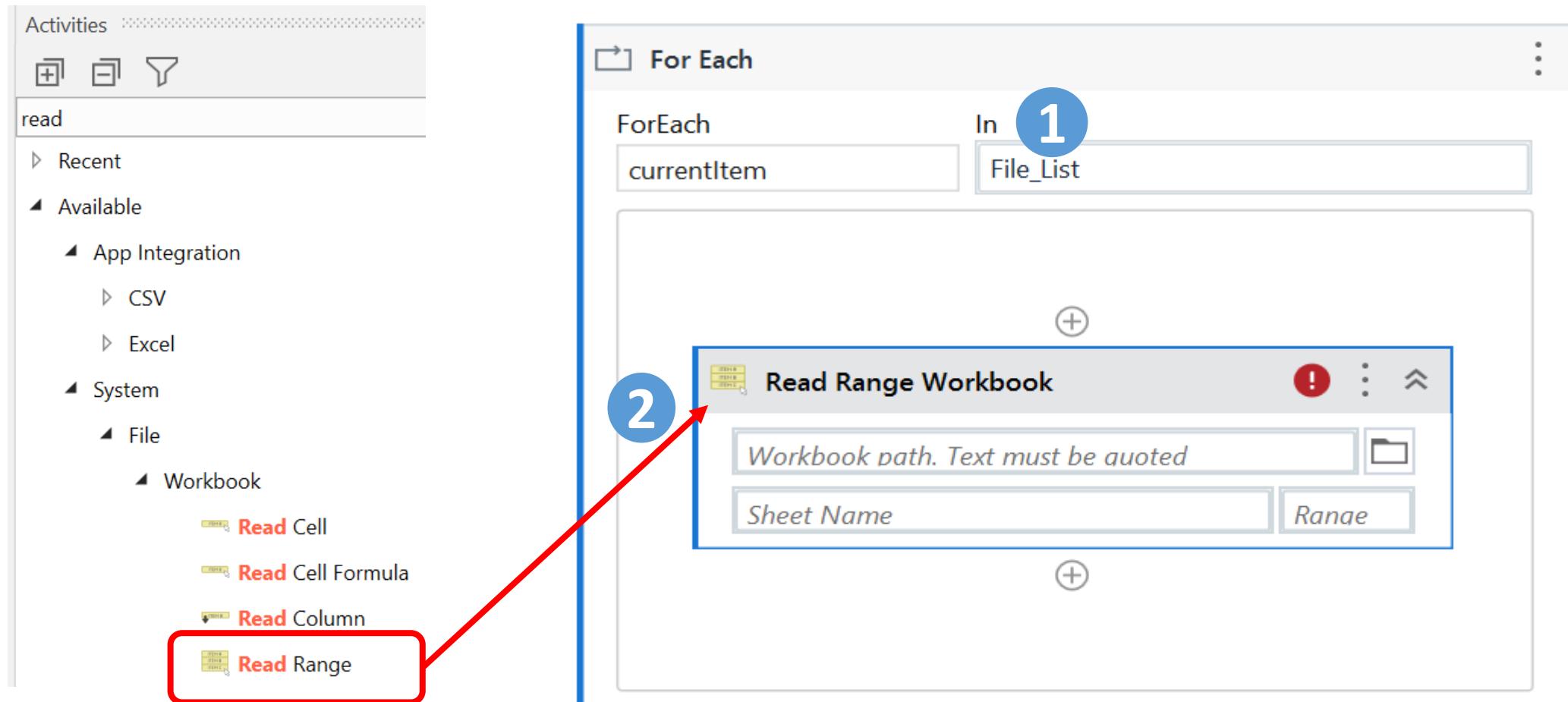
Exercise 7.1 (Step 7)

1. Search for “For Each” and drag into the Designer Panel.
2. Join “For Each” to “Assign”.
3. Change the description to “For Each File Folder”



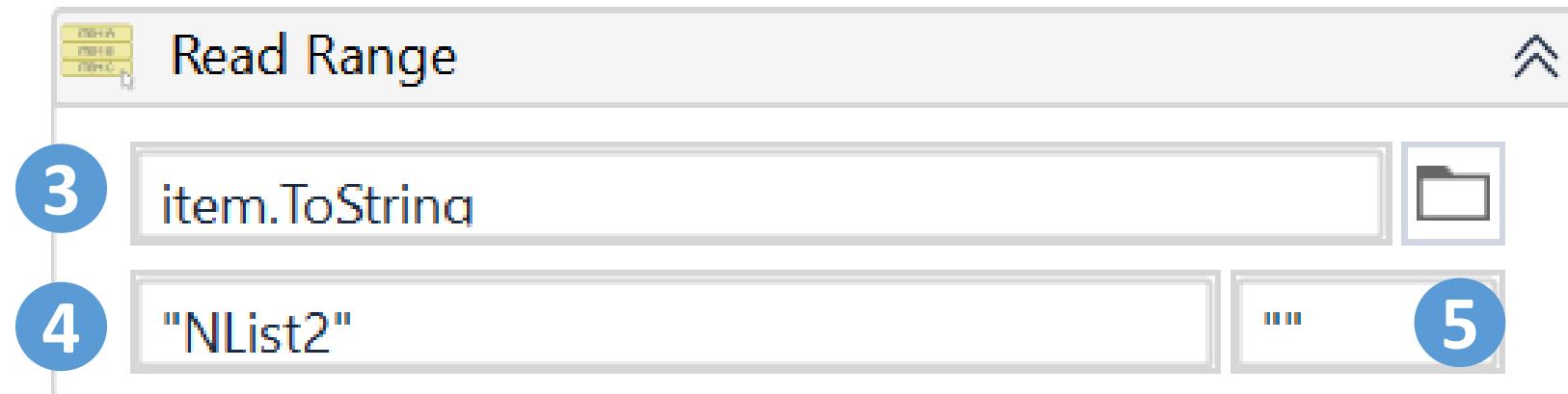
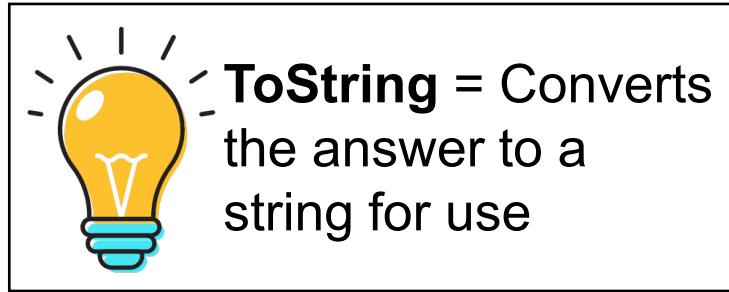
Exercise 7.1 (Step 8)

1. Select the variable “File_List” at the ForEach item field.
2. Insert the activity “Read Range” in the Body. Note that this is the “Read Range” under Workbook.



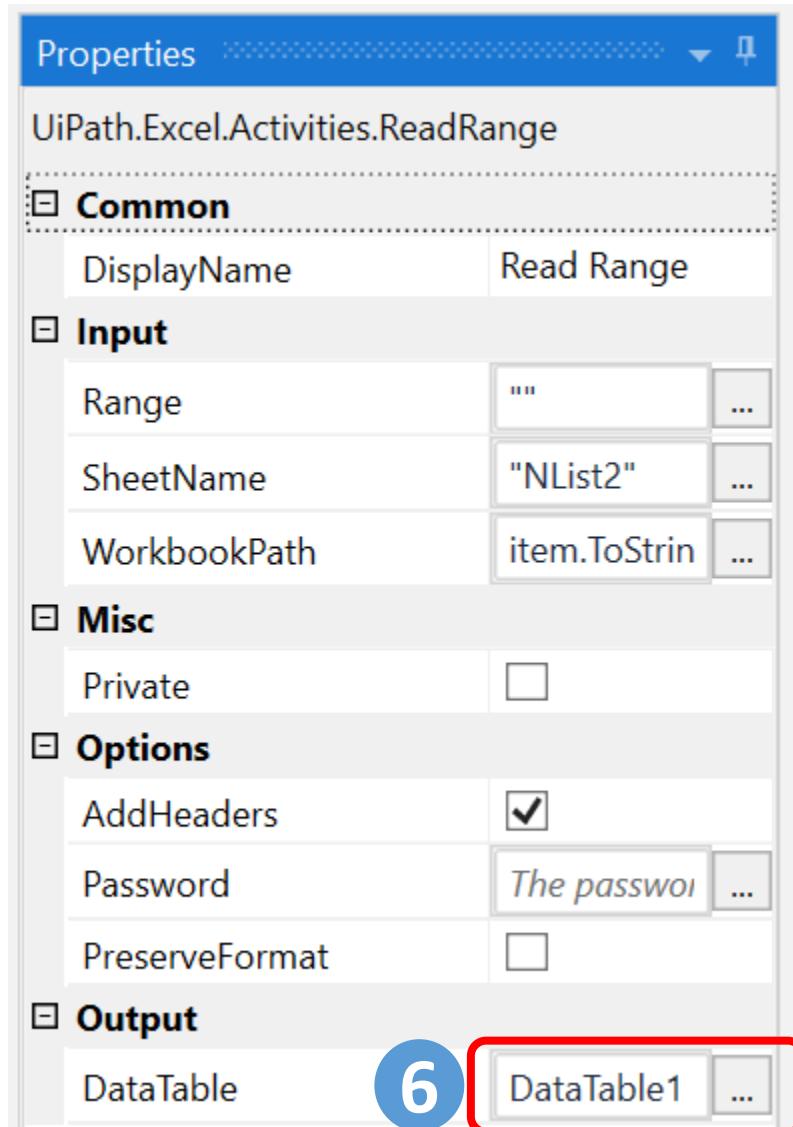
Exercise 7.1 (Step 8)

3. Input “**item.ToString**” in the box for the workbook path field (this is to reference the 5 Excel files, File_01 to File_05).
4. Change the sheet name from “Sheet1” from Sheet1 to “NList2”.
5. Change the range from “A1:A2” to “ ”.



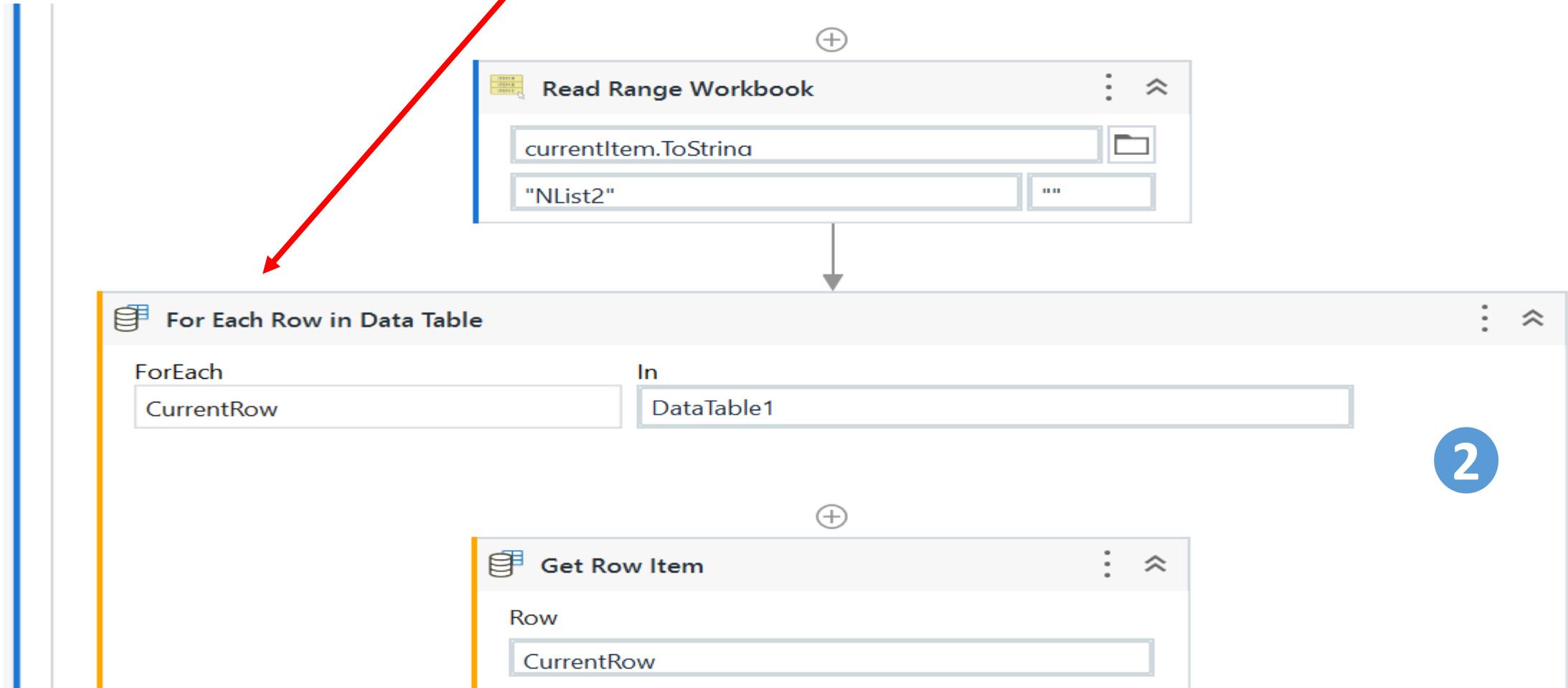
Exercise 7.1 (Step 8)

6. In the property panel of the “Read Range”, under the Output section, use the short-cut of “Ctrl + K” to create a DataTable variable “**DataTable1**”. Make sure the “AddHeaders” is Checked.



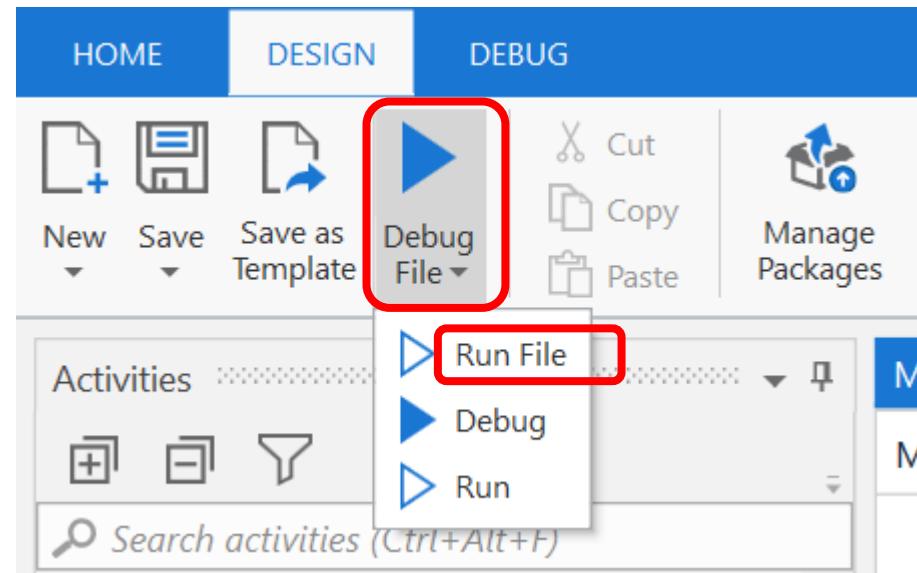
Exercise 7.1 (Step 9)

1. Copy “For Each Row in Data Table” activity in Ex 5.6 and paste it after Read Range



Exercise 7.1 (Step 10)

Click the **Debug File** button and select “Run File” to run your script.



Exercise 7.2

Processing Multiple Excel Files
Using “Append Range”

Exercise 7.2

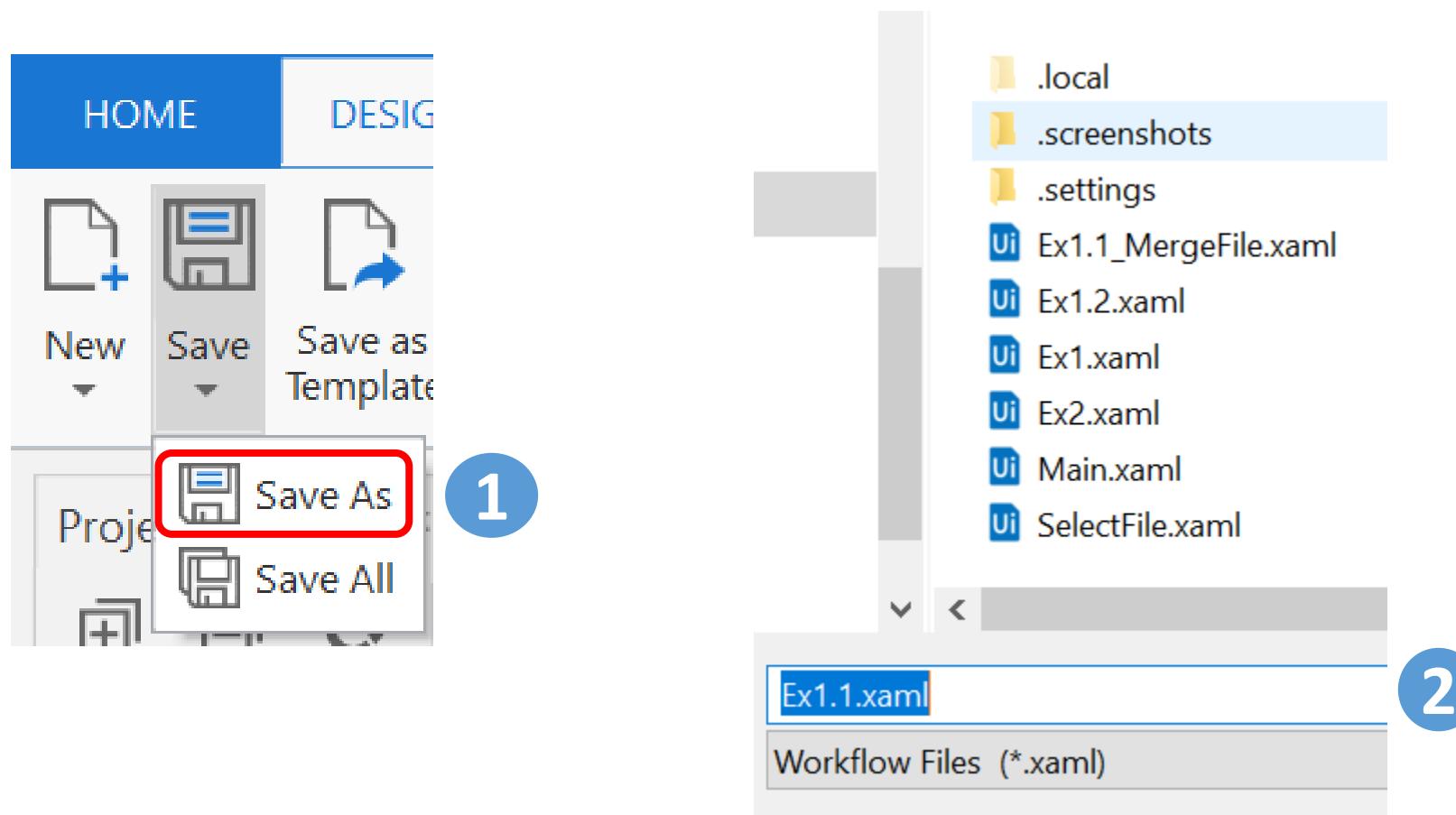
Exercise 7.1 can be scripted with some variations using:

- 1) Append Range OR
- 2) Write Range

Let's try it out by modifying Exercise 7.1!

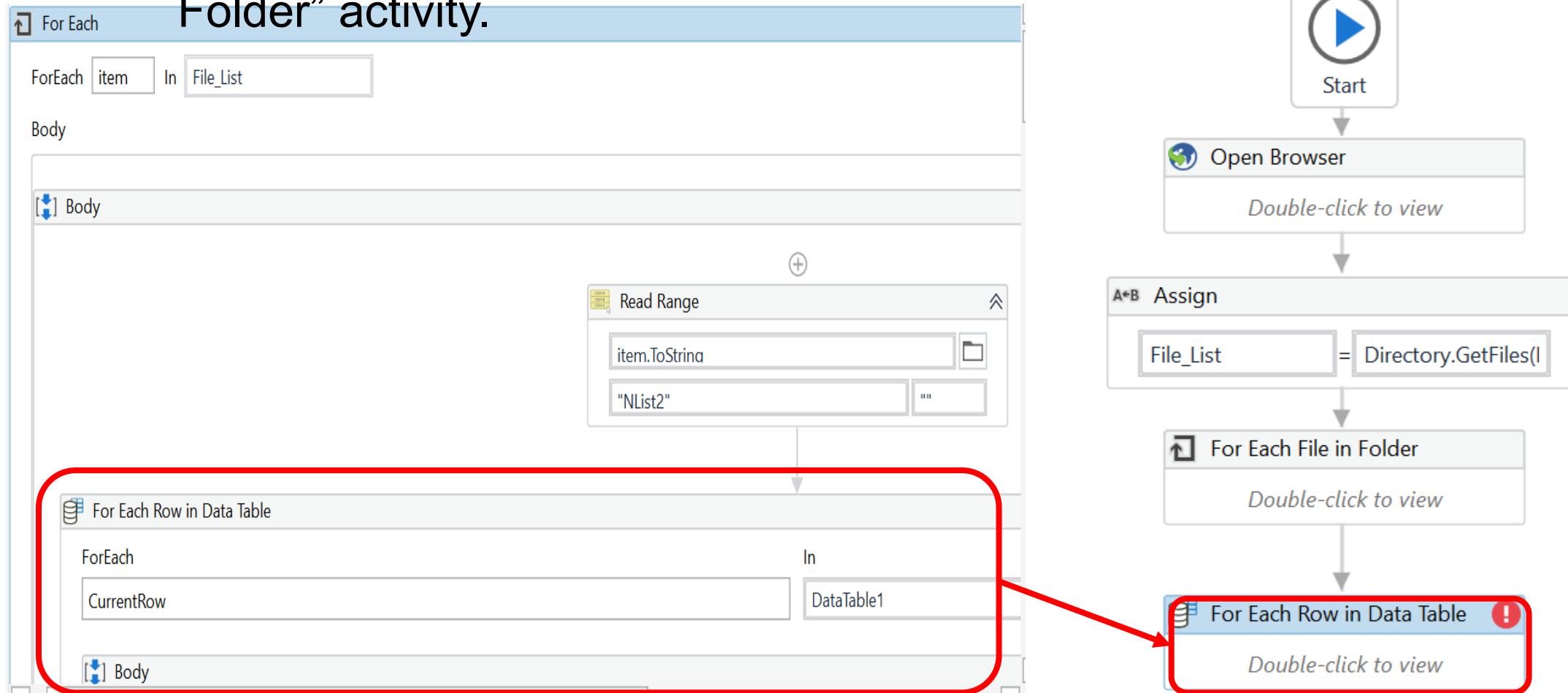
Exercise 7.2 (Step 1)

1. Click on “Save” and select “Save As”
2. Change the filename to “Ex7.2”



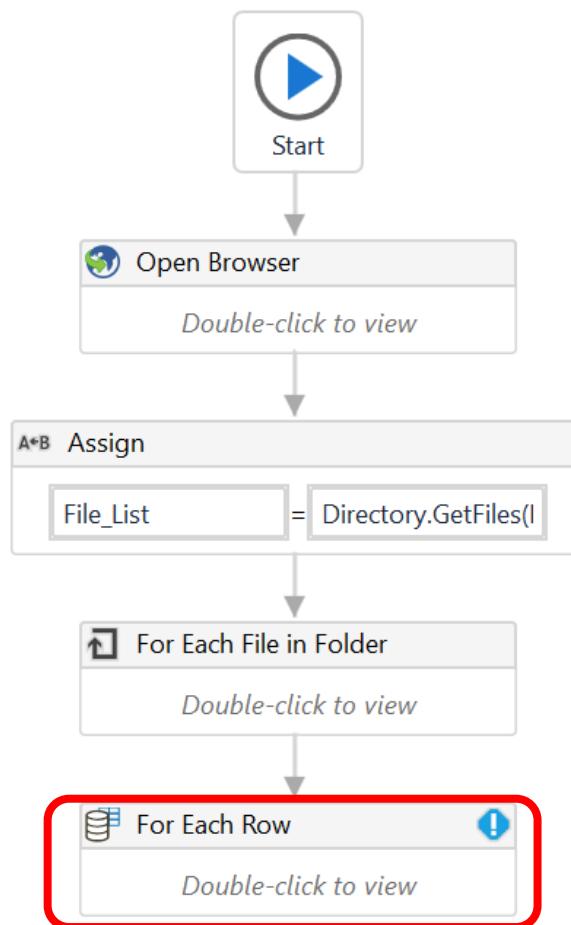
Exercise 7.2 (Step 2)

1. In the “For Each File in Folder” activity, cut and paste the “For Each Row in Data Table” activity outside the “For Each File in Folder” activity.



Exercise 7.2 (Step 2)

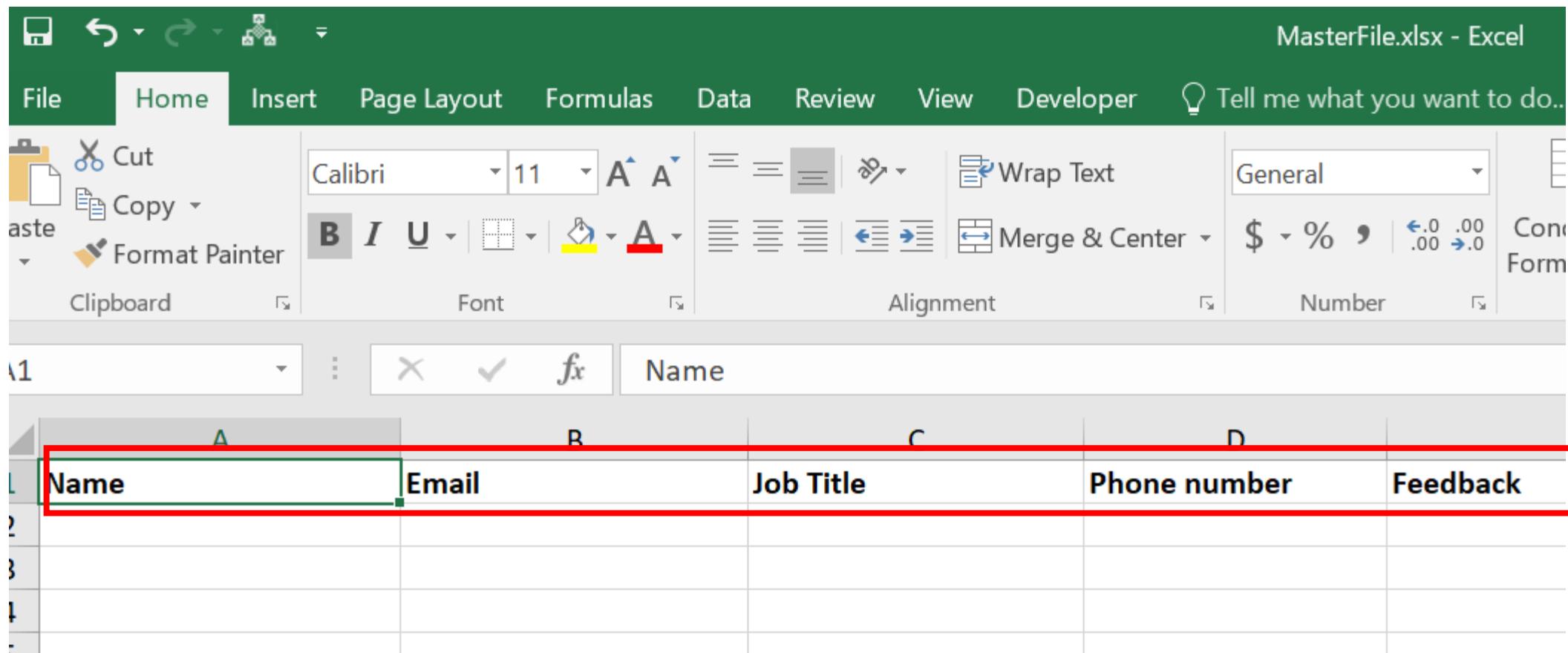
2. Correct the errors indicated in the “**For Each Row**” activity by changing the scope of all the variables accordingly.



Name	Variable type	Scope	Default
File_List	String[]	Ex7_1	Enter a VB expression
File_Directory	String	Ex7_1	"D:\DATA\EP0705 RPA\
Name	String	Ex7_1	Enter a VB expression
Email	String	Ex7_1	Enter a VB expression
Title	String	Ex7_1	Enter a VB expression
Phone	GenericValue	Ex7_1	Enter a VB expression
Feedback	String	Ex7_1	Enter a VB expression
DataTable1	DataTable	Ex7_1	Enter a VB expression

Exercise 7.2 (Step 3)

Download “MasterFile.xlsx” Excel file with the relevant headers.



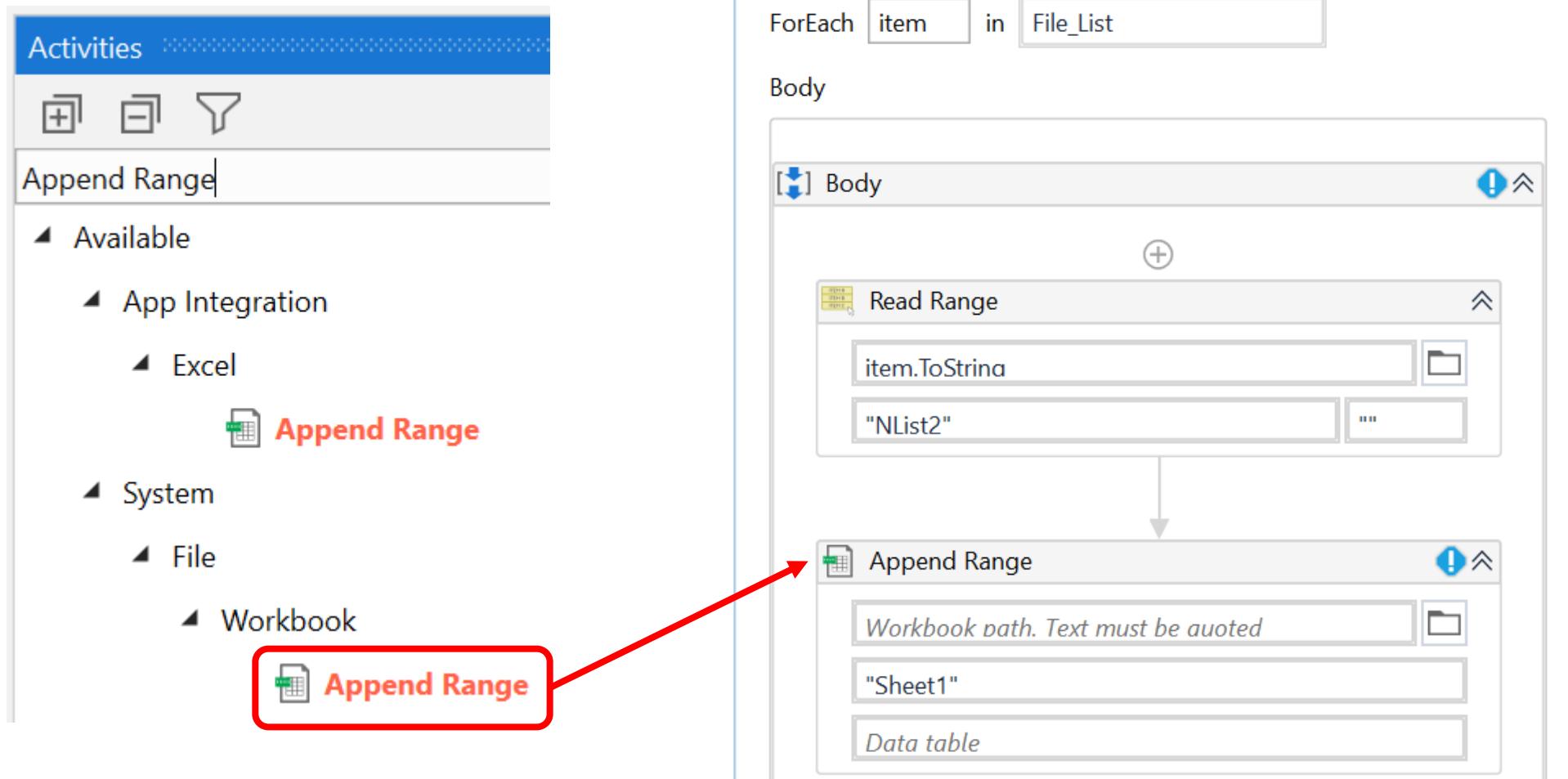
The screenshot shows a Microsoft Excel interface with the following details:

- File Tab:** The "File" tab is the active tab in the ribbon.
- Home Tab:** The "Home" tab is the currently selected tab in the ribbon.
- Clipboard Section:** Contains icons for Cut, Copy, Paste, and Format Painter.
- Font Section:** Includes Calibri font, font size 11, bold (B), italic (I), underline (U), and font color dropdowns.
- Alignment Section:** Includes horizontal alignment buttons (left, center, right) and vertical alignment buttons (top, middle, bottom).
- Number Section:** Includes a dropdown for number formats (General) and currency symbols (\$, %, , .).
- Cells:** The first row of the worksheet is highlighted with a red border. The columns are labeled A, B, C, D, and E. The cells contain the following header text:

Name	Email	Job Title	Phone number	Feedback
------	-------	-----------	--------------	----------

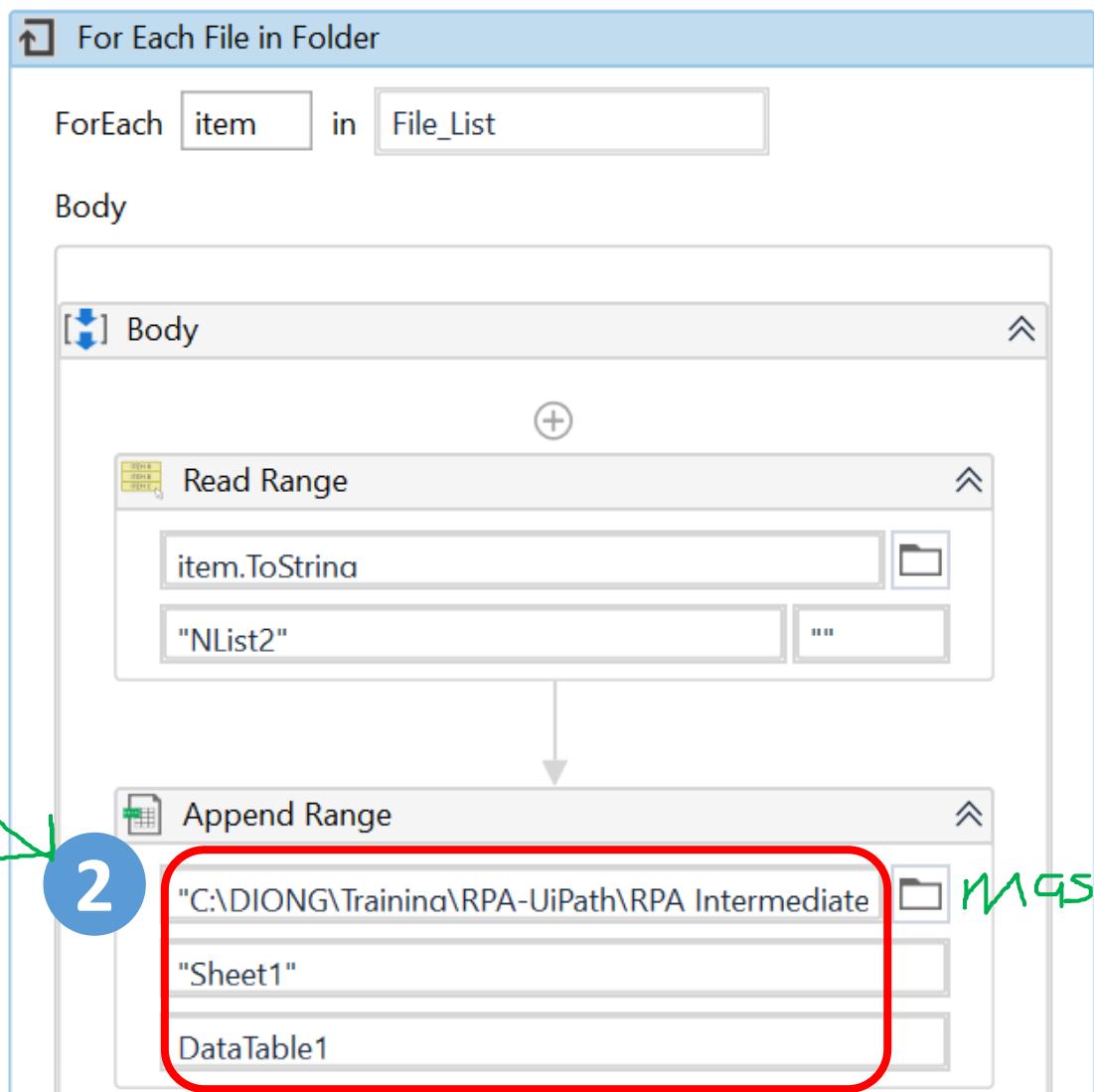
Exercise 7.2 (Step 4)

1. In the “For Each File in Folder” activity, insert an “Append Range” activity



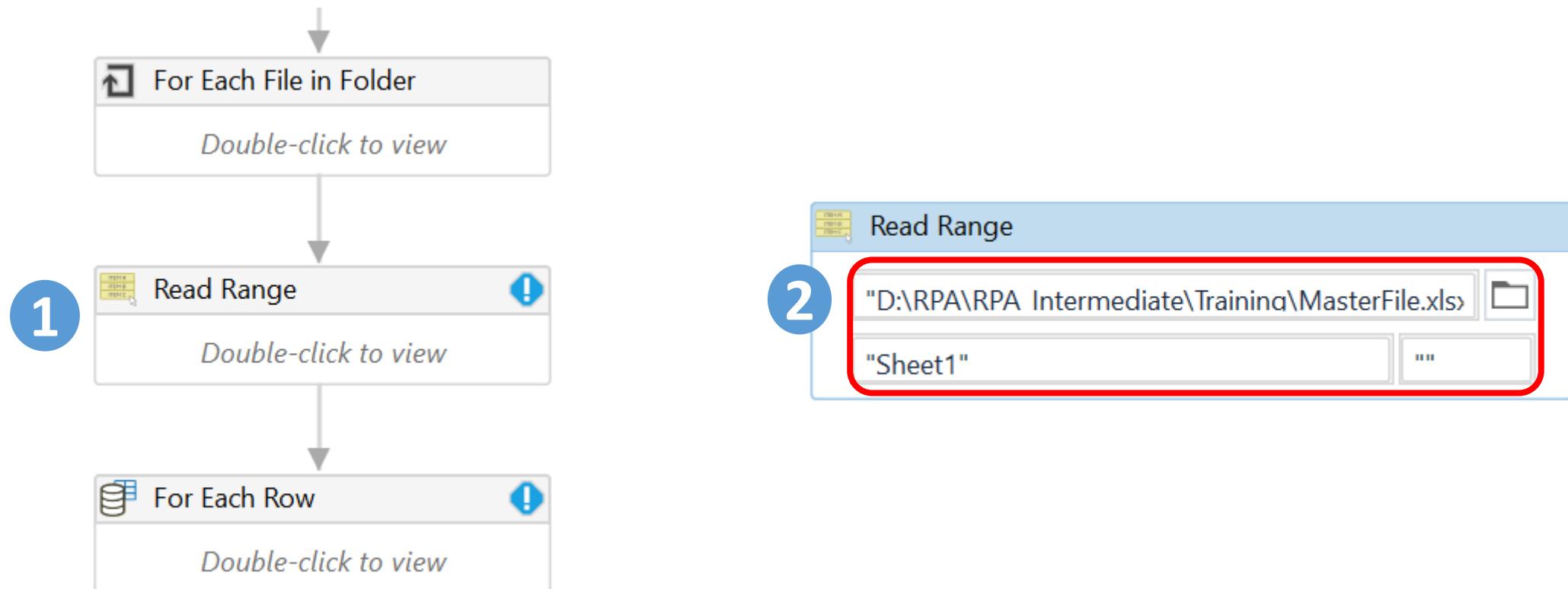
Exercise 7.2 (Step 4)

2. Input the path of the Excel file, "MasterFile.xlsx"
3. Update the DataTable to **DataTable1**.



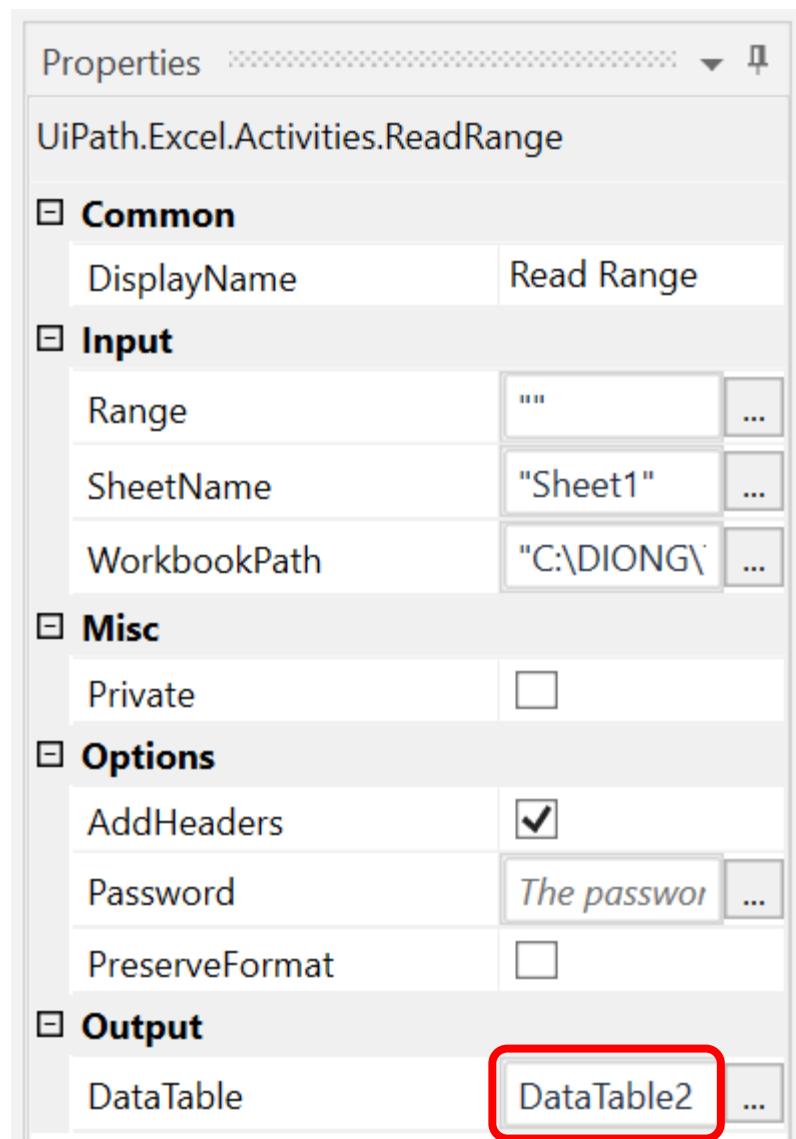
Exercise 7.2 (Step 5)

1. Insert a “Read Range” activity after “For Each File in Folder”.
2. In the “Read Range” box, input the path of the Excel file, “MasterFile.xlsx” and update the range to “ “.



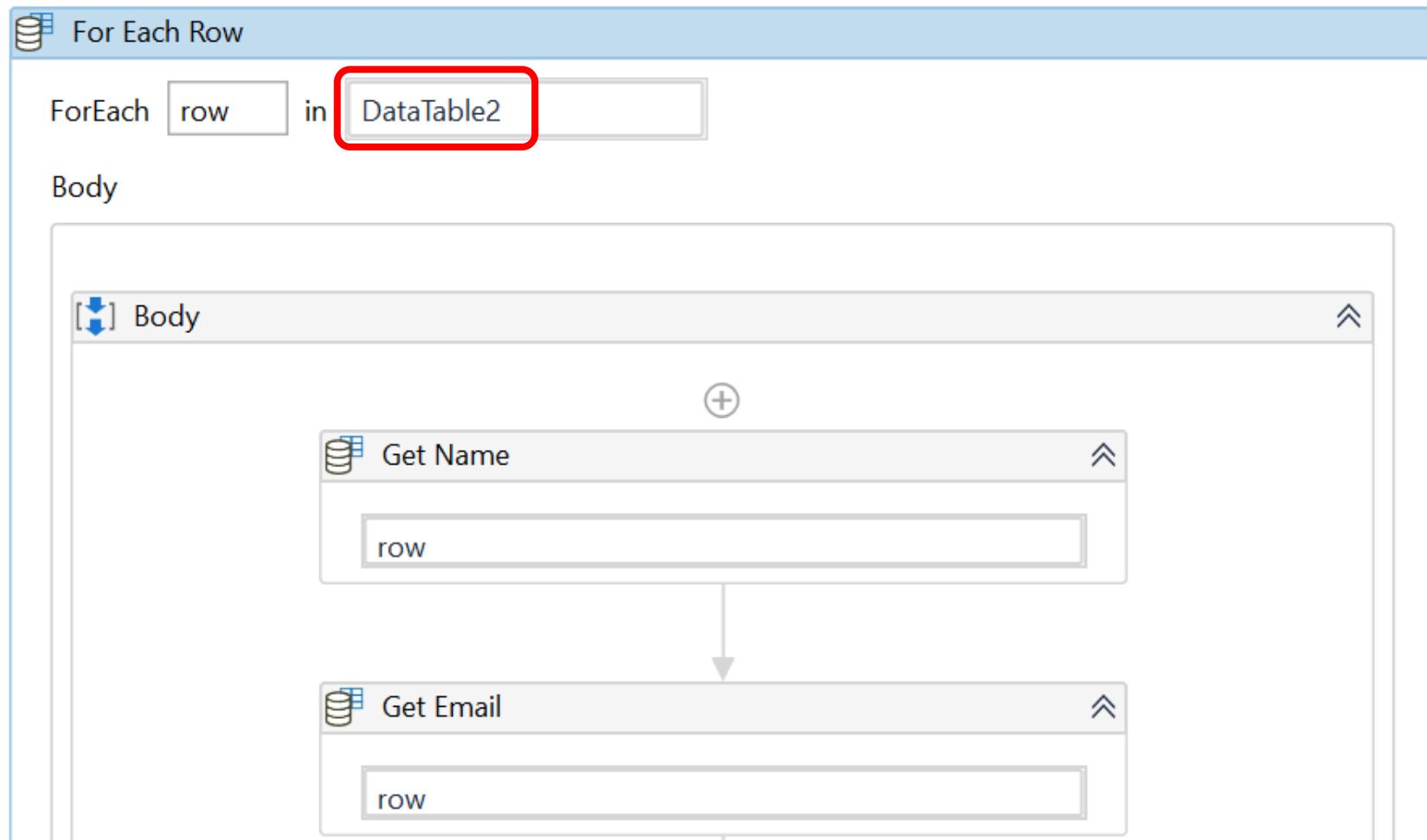
Exercise 7.2 (Step 5)

3. In the property panel of the “**Read Range**”, under the Output section, use the short-cut of “Ctrl + K” to create a DataTable variable “**DataTable2**”.



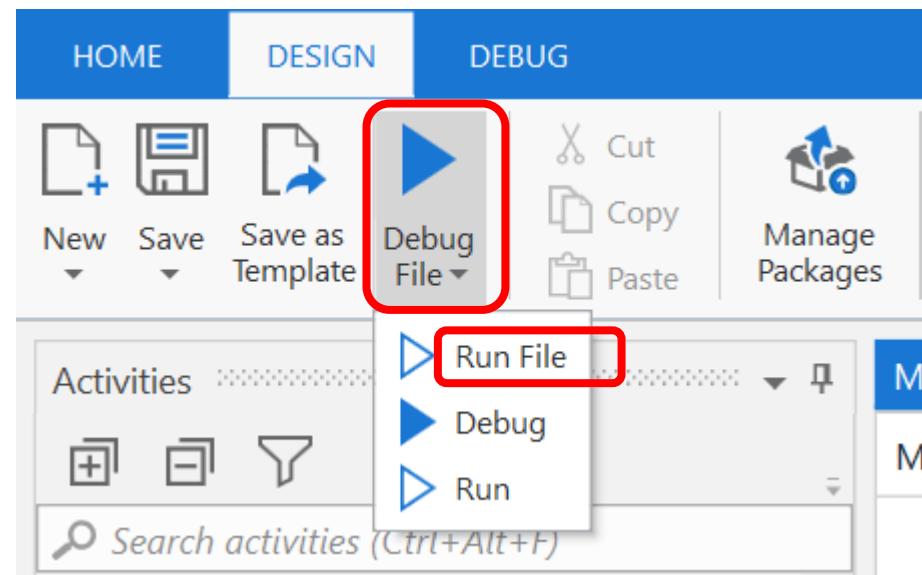
Exercise 7.2 (Step 6)

In the “For Each Row” activity, update the VB expression from “Datatable1” to “Datatable2”.



Exercise 7.2 (Step 7)

Click the **Debug File** button and select “Run File” to run your script.



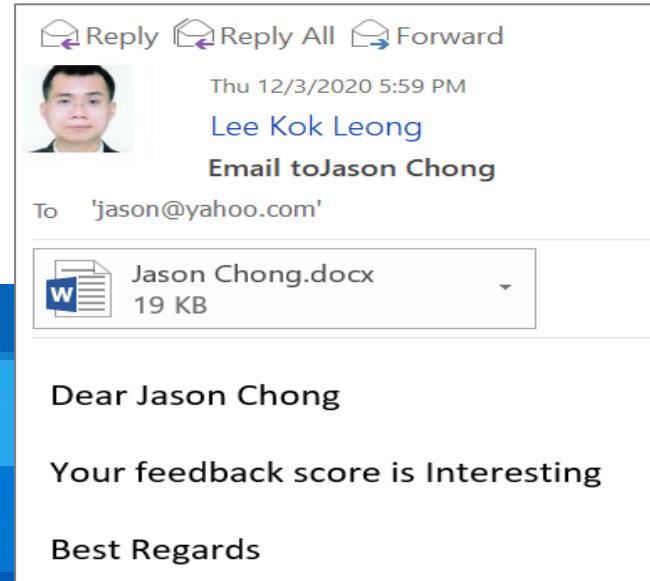


Ex8.1-8.2 Sending Multiple Emails

1. Continue using Robot4.

2. Script Robot to send multiple emails including attachment using “**Send Outlook Mail Message**” (ex 8.1) and using a template (Ex 8.2)

3. Refer to Ex8.1 to 8.2 in “RPA UiPath Hands-On Guide” for step-by-step instruction



Note to Student

Exercise 8.1 Sending Multiple Emails with Attachments

- Use the Microsoft Outlook Desktop version only, not the browser version.
- Please install before lesson
- Turn Outlook to WorkOffline mode while scripting

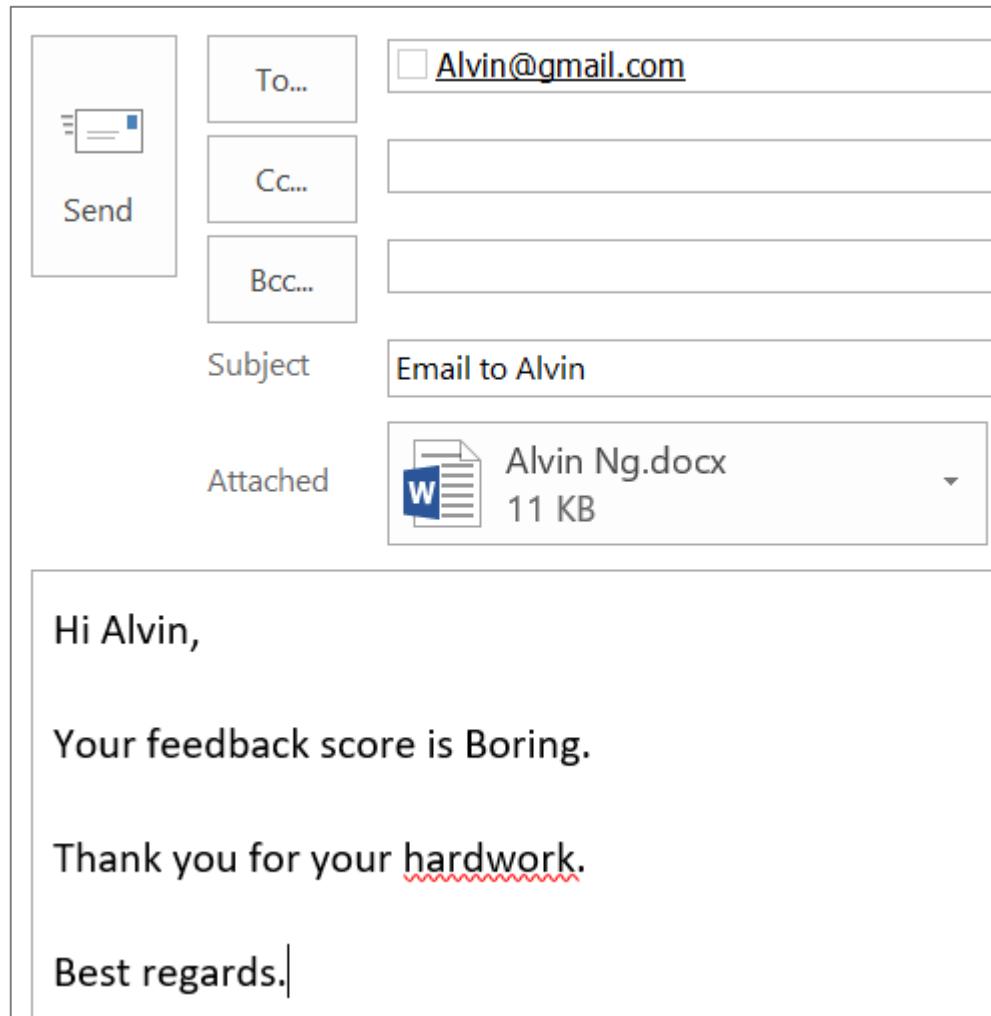
Exercise 8.1

Sending Multiple Emails
with Attachments

Exercise 8.1

This exercise shows how to send multiple emails with customised text and attachments.

Name	Email	Job Title	Phone number	Feedback
Alvin Ng	Alvin@gmail.com	Lecturer	63216781	Boring
Bee Leng	Beeleng@gmail.com	Finance Manager	90001892	Interesting
David	david@yahoo.com	Business Manager	56777777	Neutral
Jason Chong	jason@yahoo.com	Marketing Manager	33888999	Interesting
Kok Leong	kokleong@gmail.com	IT Manager	22266677	Boring





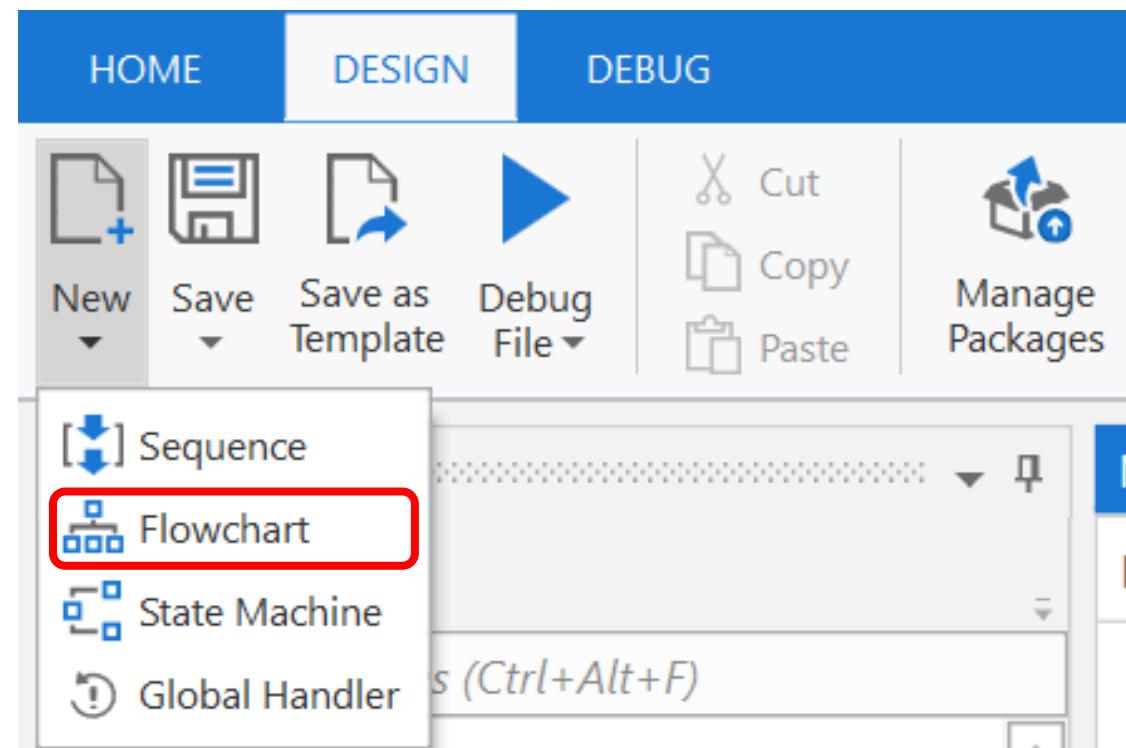
Exercise 8.1

What you need:

- [Excel files File_01](#)
- Folder with 5 Word files, “Alvin Ng”, “Bee Leng”, “David”, “Jason Chong” and “Kok Leong” (for use as email attachments)

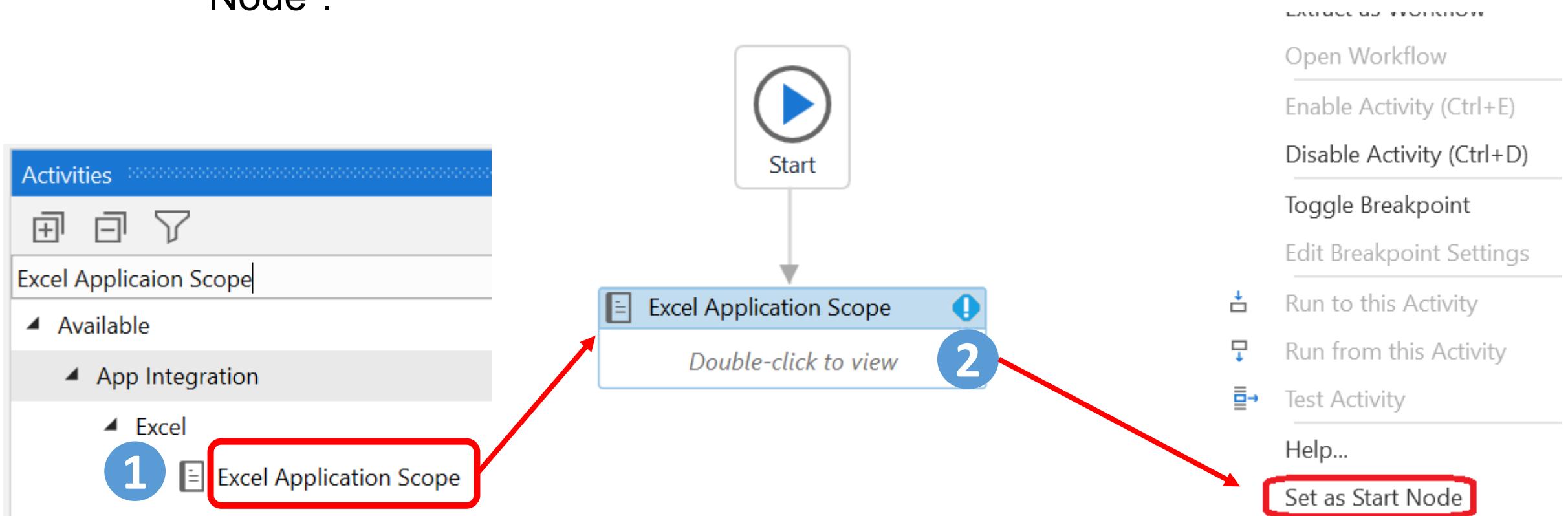
Exercise 8.1 (Step 1)

In UiPath Studio, create a new flowchart, and name it “Ex8.1”.



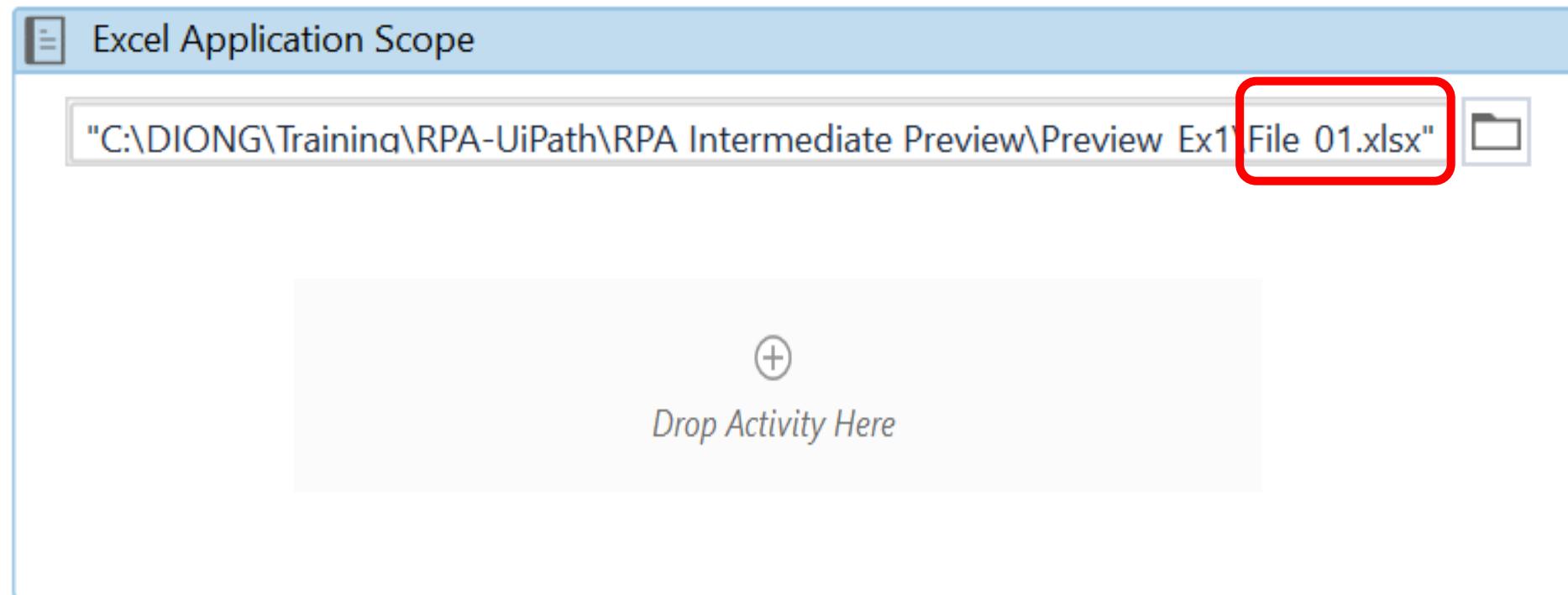
Exercise 8.1 (Step 2)

1. Search for “Excel Application Scope” and drag into the Designer Panel.
2. Right-click on “Excel Application Scope”, and select “Set as Start Node”.



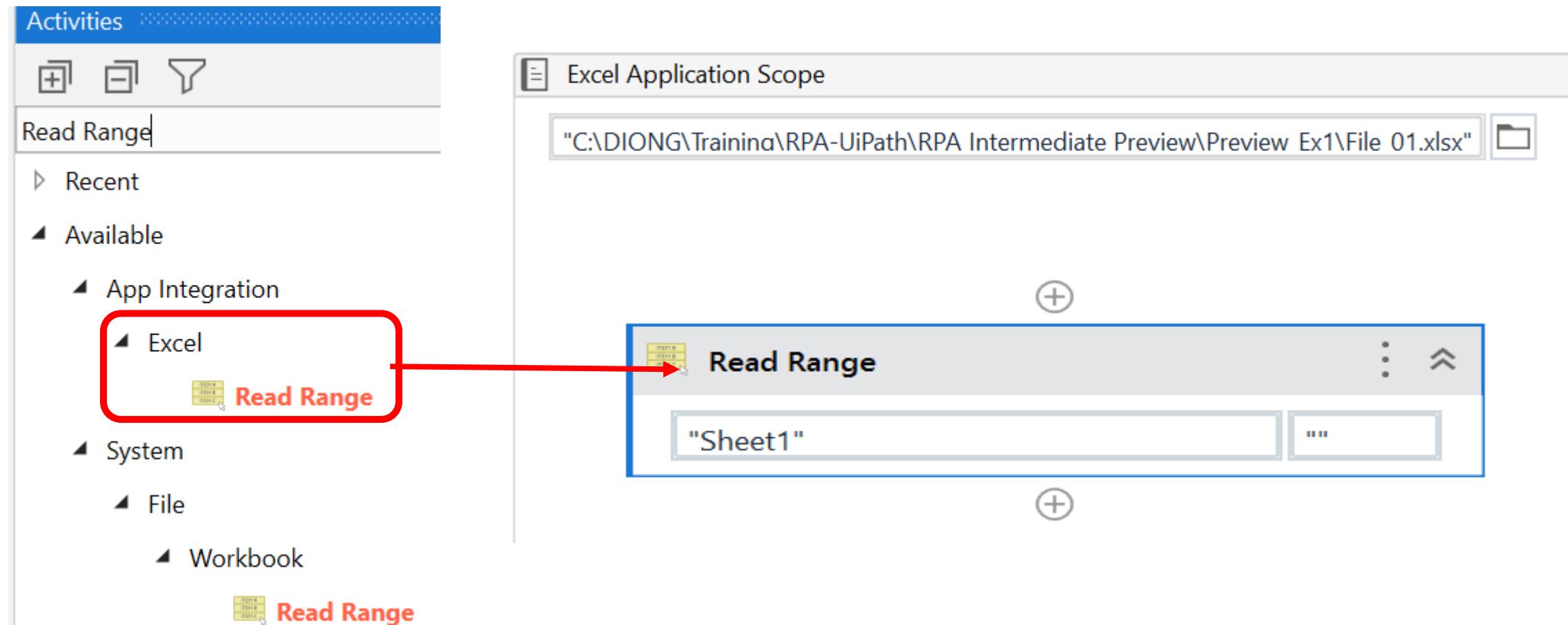
Exercise 8.1 (Step 3)

1. Double click to open “Excel Application Scope” box.
2. In the workbook path field, browse for and select the Excel file, “File_01.xlsx”.



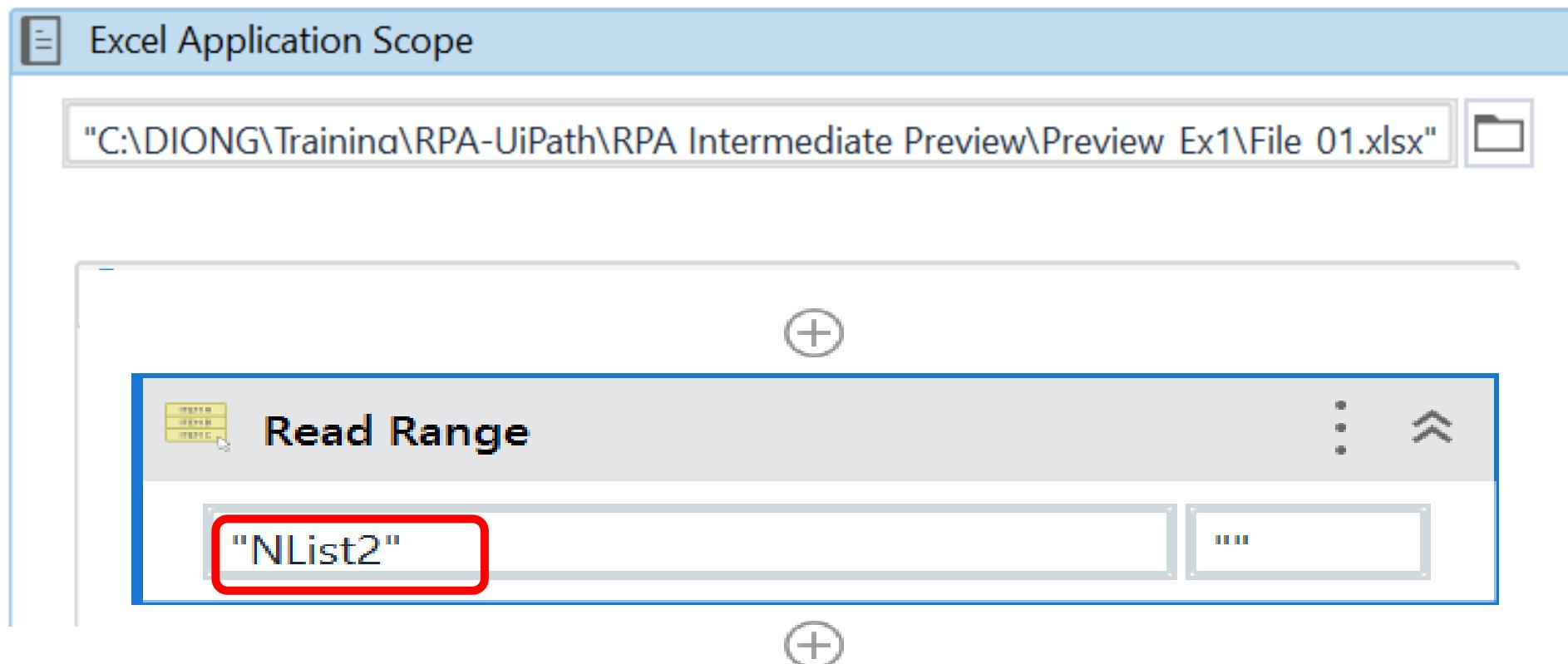
Exercise 8.1 (Step 4)

1. Search for “Read Range” and drag into the “Excel Application Scope” activity.



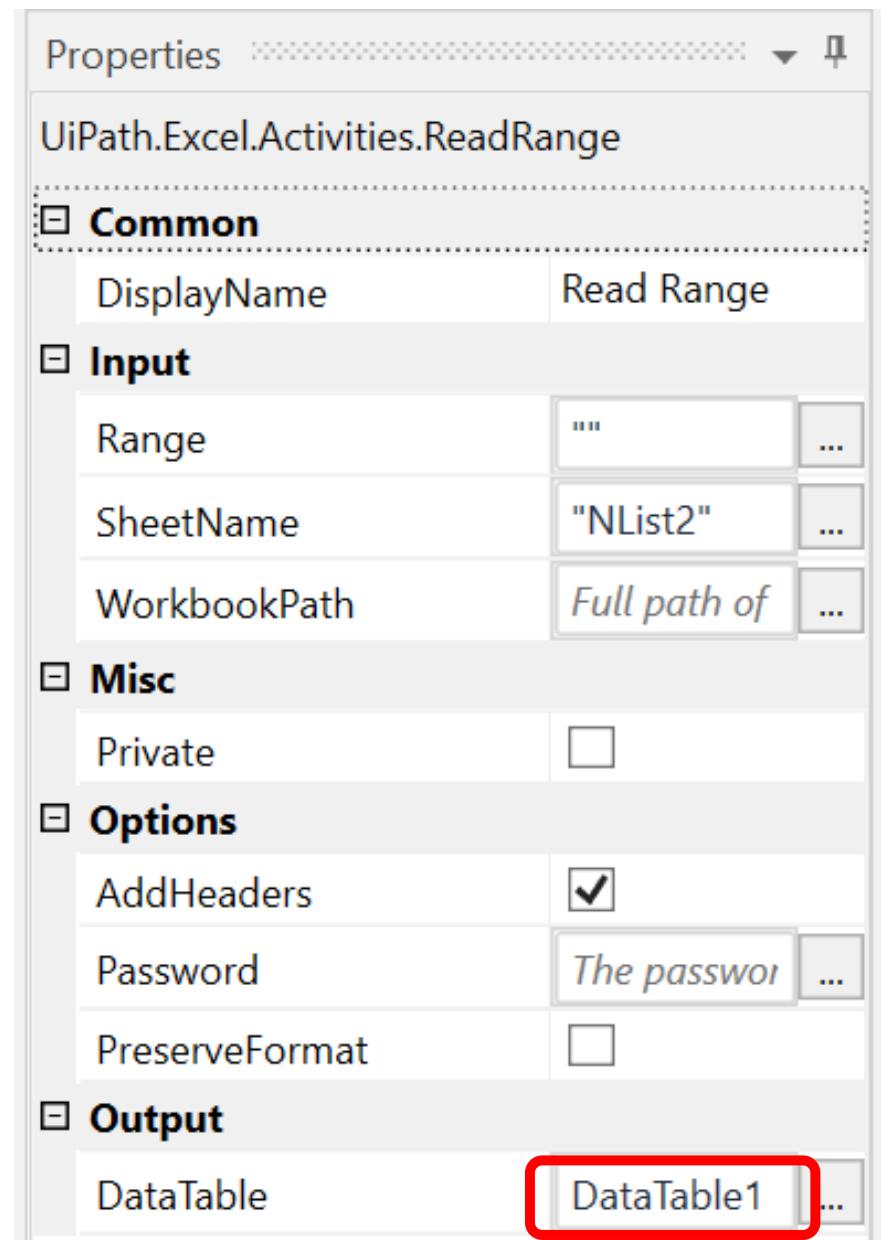
Exercise 8.1 (Step 4)

2. Change the worksheet from “Sheet1” to “NList2”.



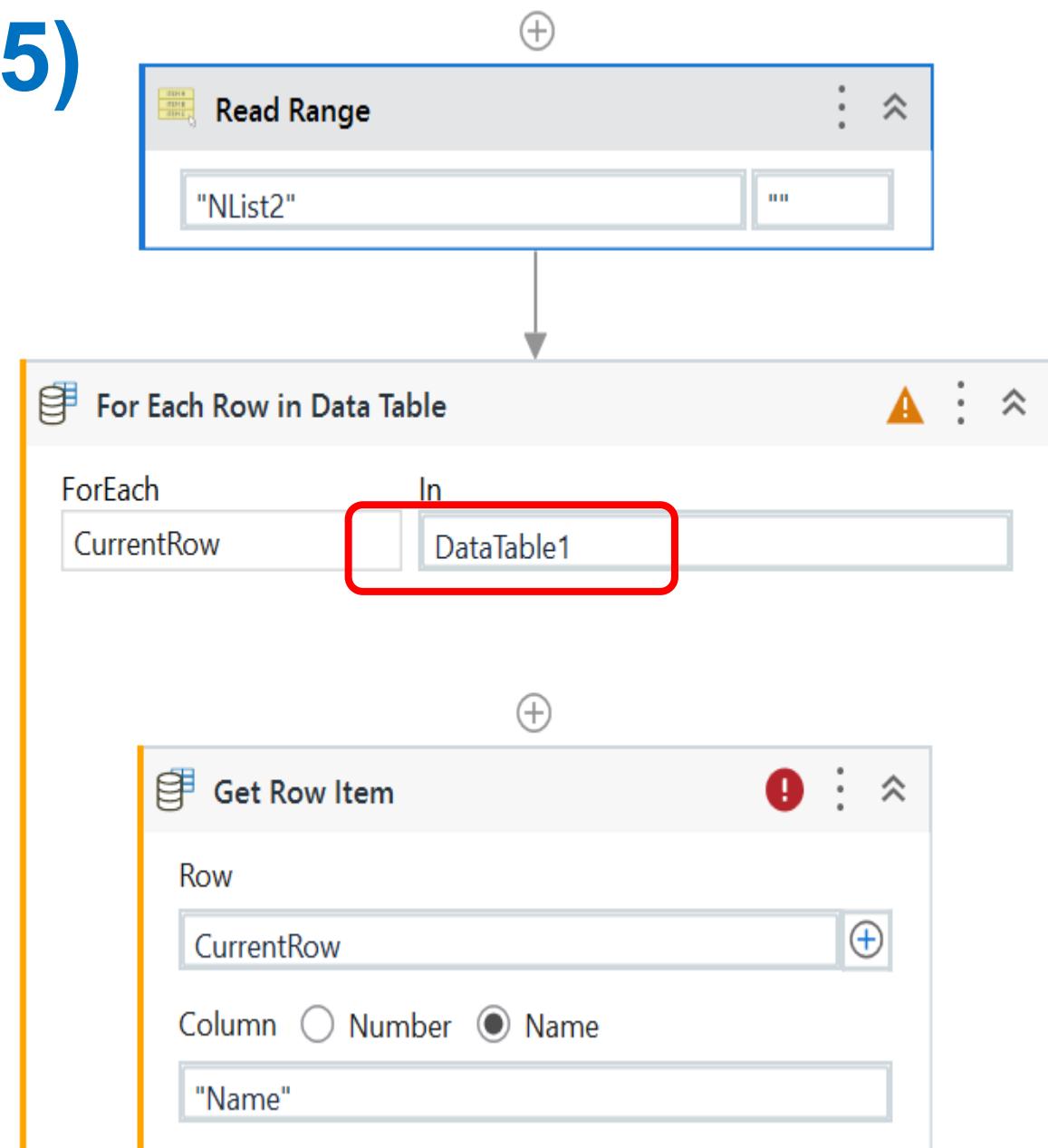
Exercise 8.1 (Step 4)

3. In the properties panel of the “**Read Range**”, under the Output section, use the short-cut of “Ctrl + K” to create a DataTable variable “**DataTable1**”.



Exercise 8.1 (Step 5)

1. Copy “**For Each Row in Data Table**” activity in Ex 7.2 and paste it after Read Range
2. Change DataTable2 to **DataTable1**
3. Del activities related to Form 2



Exercise 8.1 (Step 6)

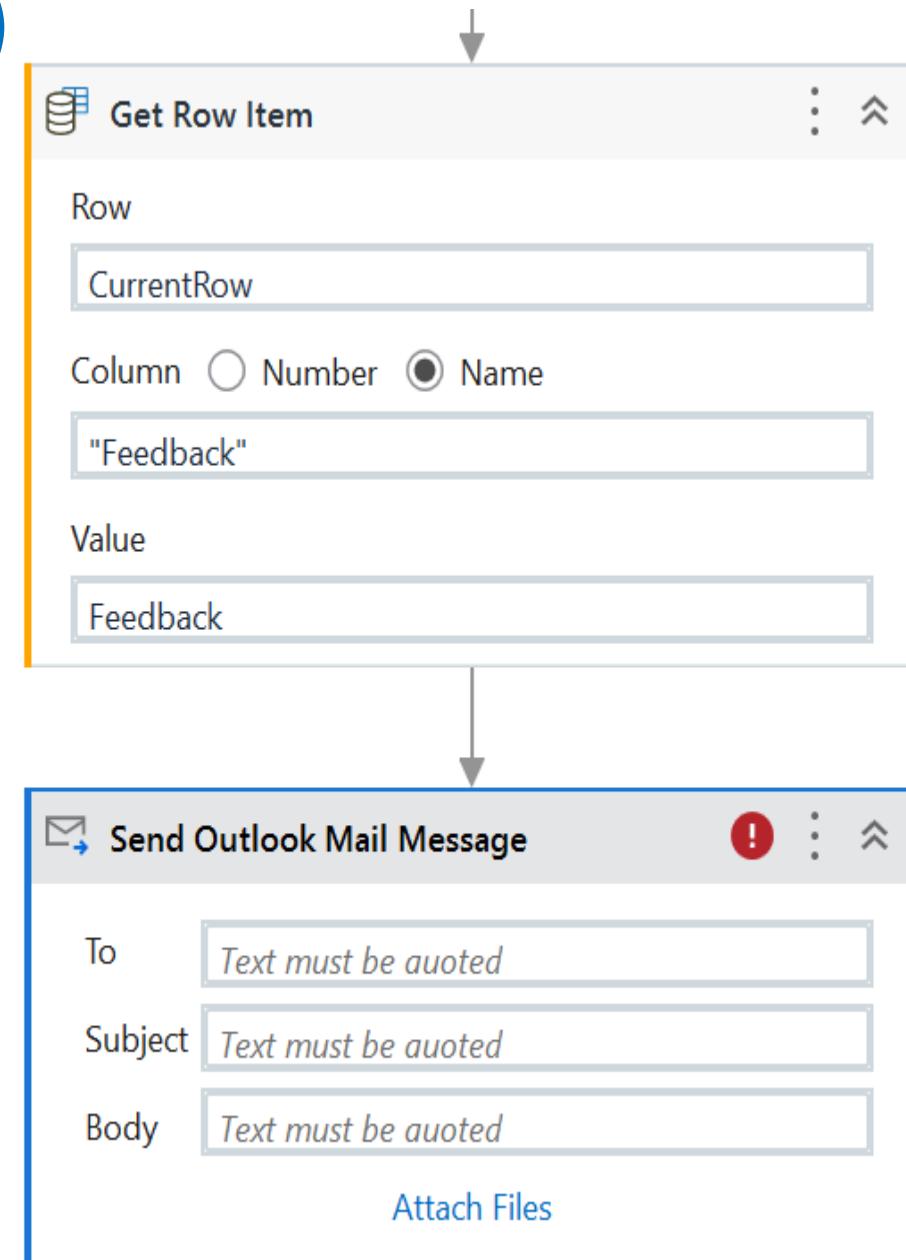
Create the following new variables.

Please note that the variable “Phone” must be of the variable type “GenericValue”

Name	Variable type	Scope	Default
Name	String	Body	<i>Enter a VB expression</i>
Email	String	Body	<i>Enter a VB expression</i>
Job	String	Body	<i>Enter a VB expression</i>
Phone	GenericValue	Body	<i>Enter a VB expression</i>
Feedback	String	Body	<i>Enter a VB expression</i>

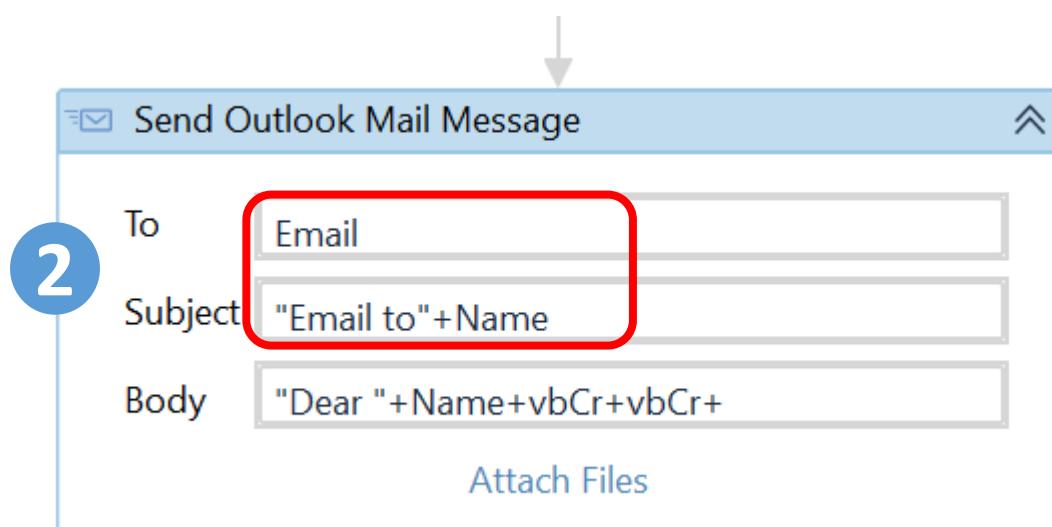
Exercise 8.1 (Step 8)

1. Insert the activity “Send Outlook Mail Message” after the “Get Feedback” activity.



Exercise 8.1 (Step 8)

2. Insert the activity “Send Outlook Mail Message”, input “To” and “Subject”.
3. The “Body” can be input in the Expression Editor in the Properties Panel (click on ... to open).



Properties

UiPath.Mail.Outlook.Activities.SendOutlookMail

- Attachments**
 - Attachments (Collection)
 - AttachmentsCollection
- Common**
 - DisplayName Send Outlook Mail I...
 - TimeoutMS
- Email**
 - Body "Dear "+Name (This field is highlighted with a red box)
 - Subject "Email to"+Na...

3

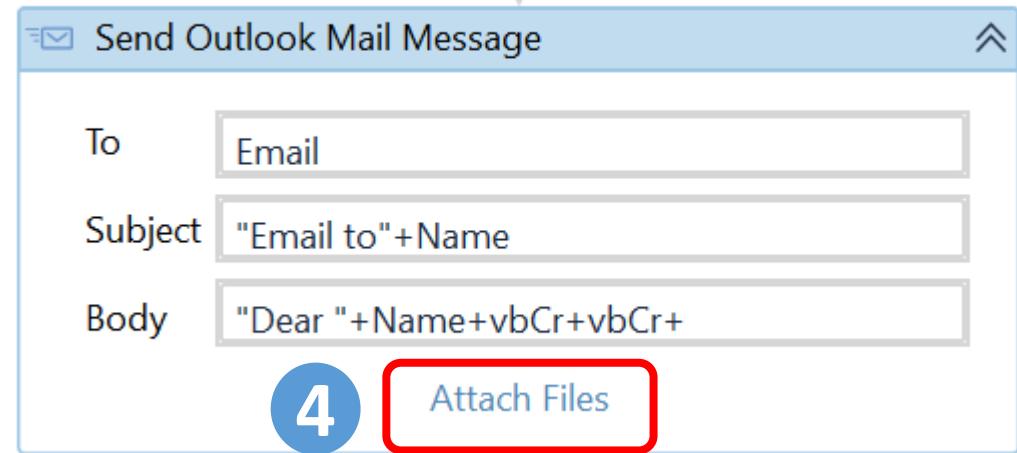
Expression Editor

Body (String)

```
"Dear "+Name+vbCr+vbCr+
"Your feedback score is "+Feedback+vbCr+vbCr+
"Best Regards"
```

Exercise 8.1 (Step 8)

4. Click on “Attach Files”.
5. Click on “Create Argument” and input the path of the attachments



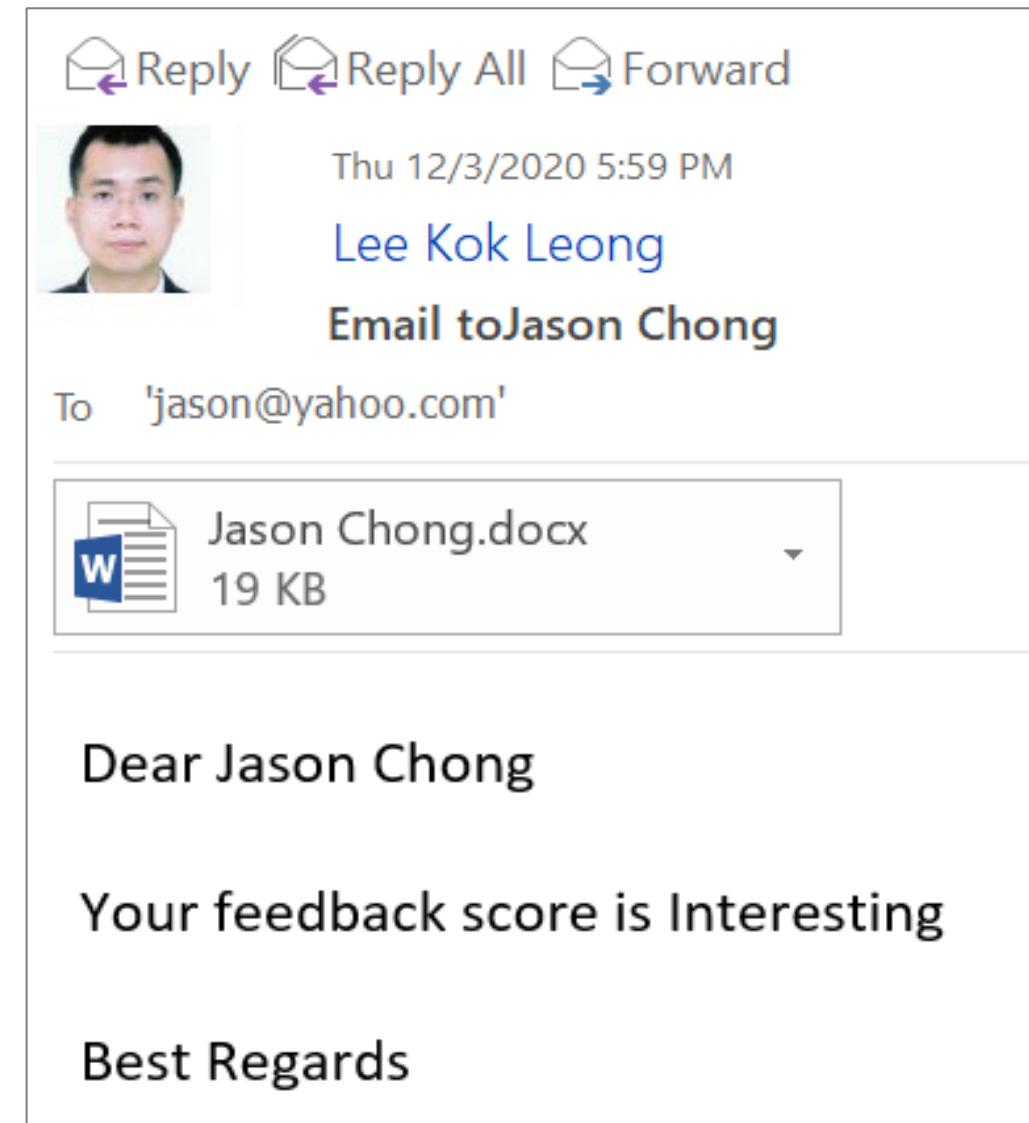
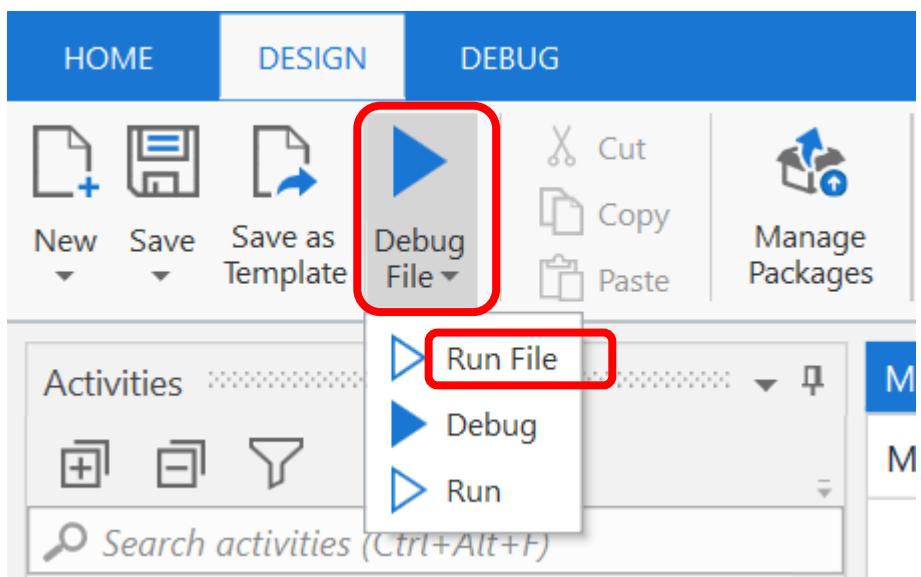
Direction	Type	Value
In	String	"C:\DIONG\Training\RPA-Uipath\RPA Intermediate Preview\Preview_Ex2.0\"+Name+".docx"
<i>Create Argument</i>		

OK Cancel

Exercise 8.1 (Step 9)

Click the **Debug File** button and select “Run File” to run your script.

Check your outlook email to confirm that the emails with attachments have been created.

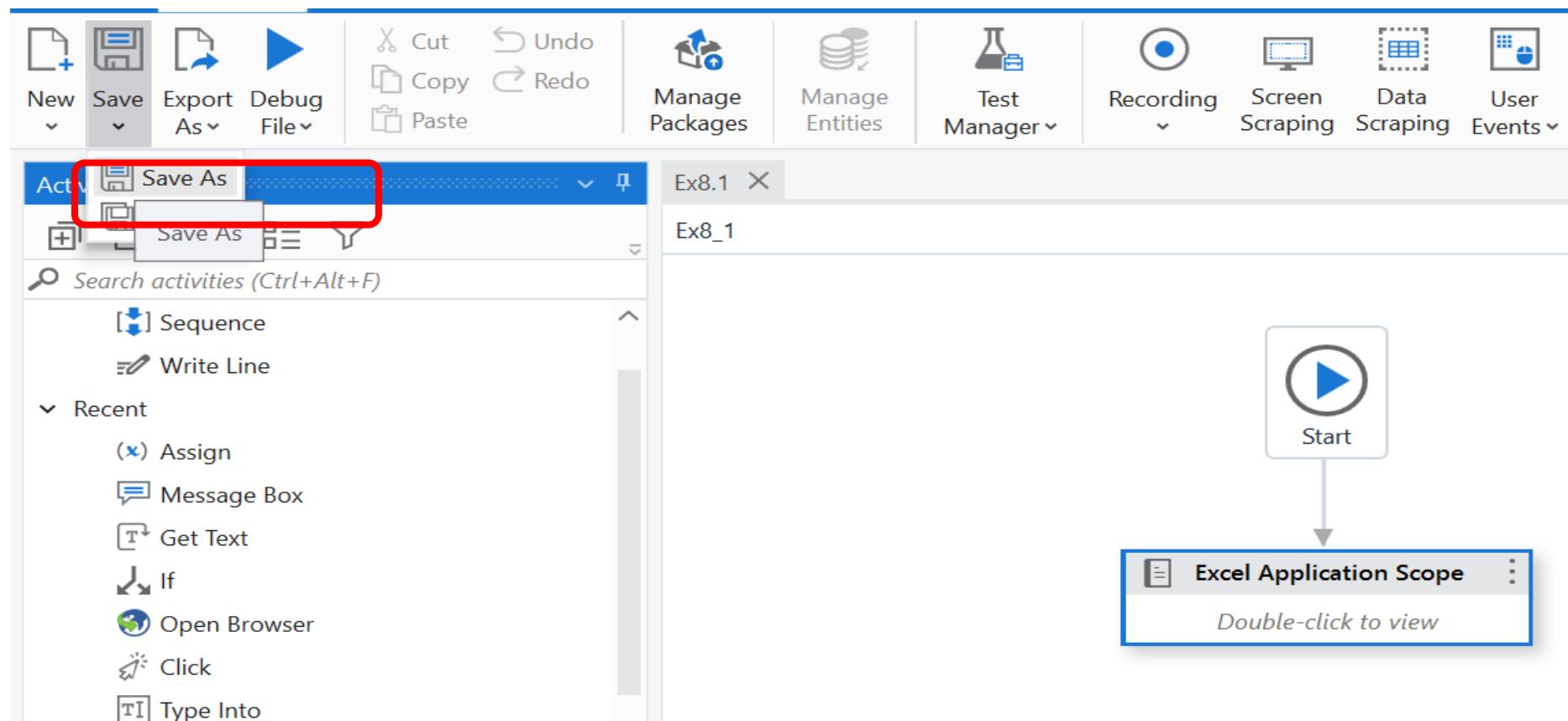


Exercise 8.2

Sending Email in Outlook
using a template

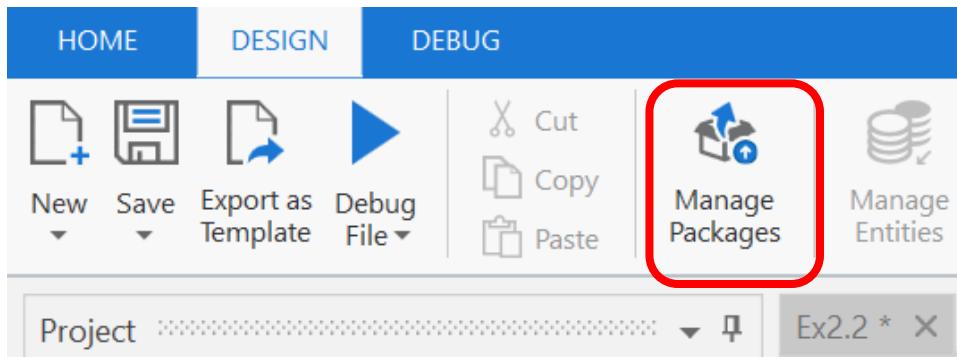
Exercise 8.2 (Step 1)

In UiPath Studio, save Ex8.1 and name it “Ex8.2”.

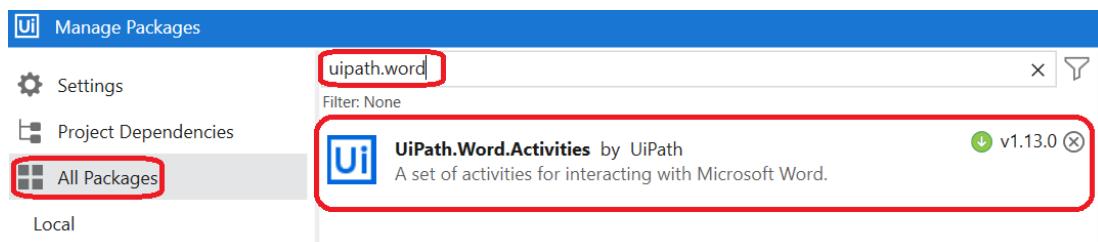


Exercise 8.2 (Step 2)

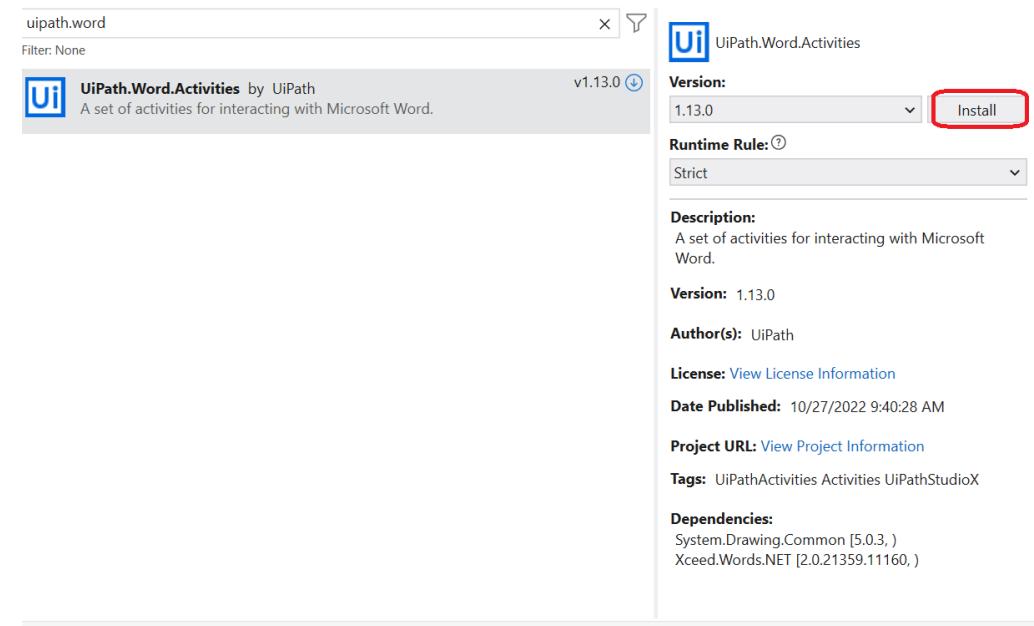
1. Select “Manage Packages”



2. Select “All Packages”, type “UiPath.Word” and click on the activity.

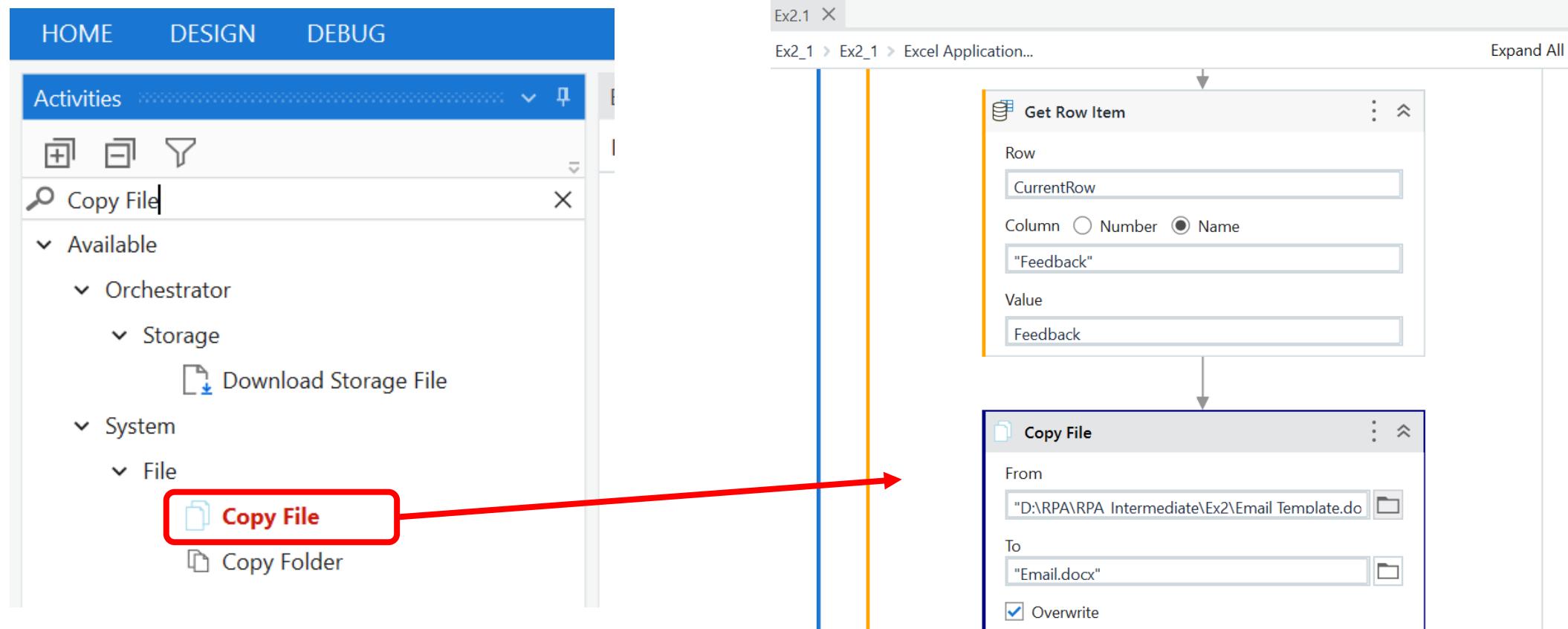


3. Click “Install” follow by “Save”



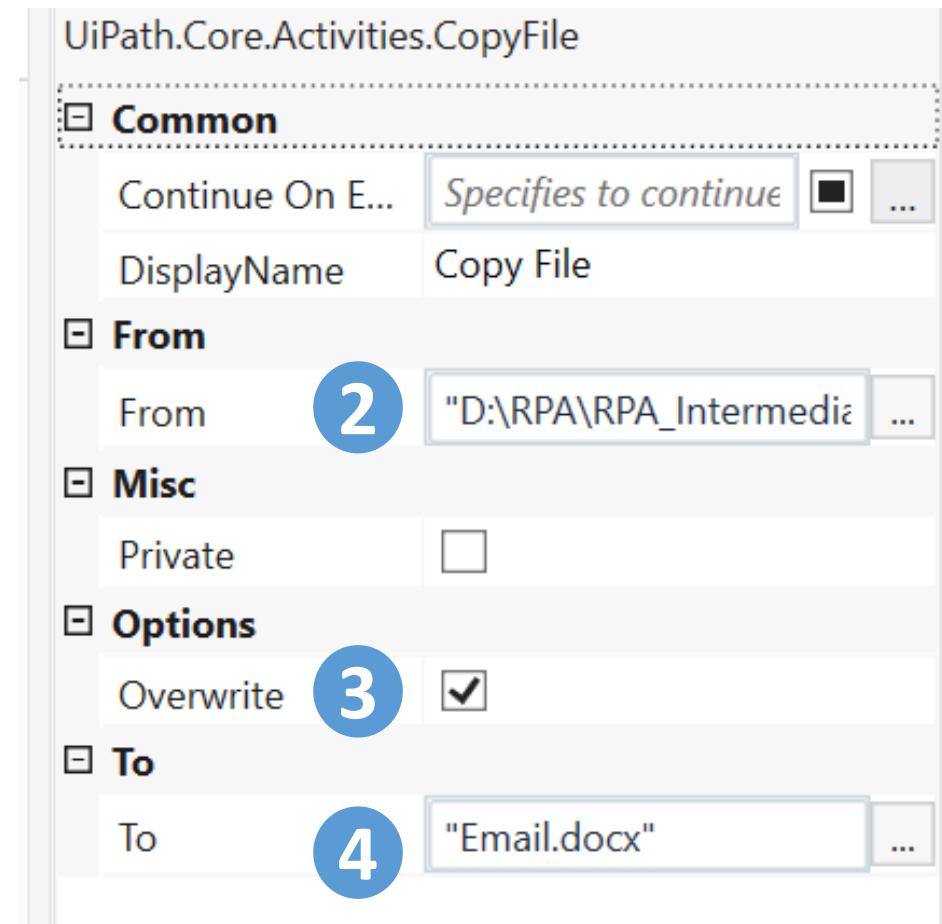
Exercise 8.2 (Step 3)

1. Remove “Send Outlook email”
2. Search and insert “Copy File” within the “For Each Row...” loop, below the last Get Row Item



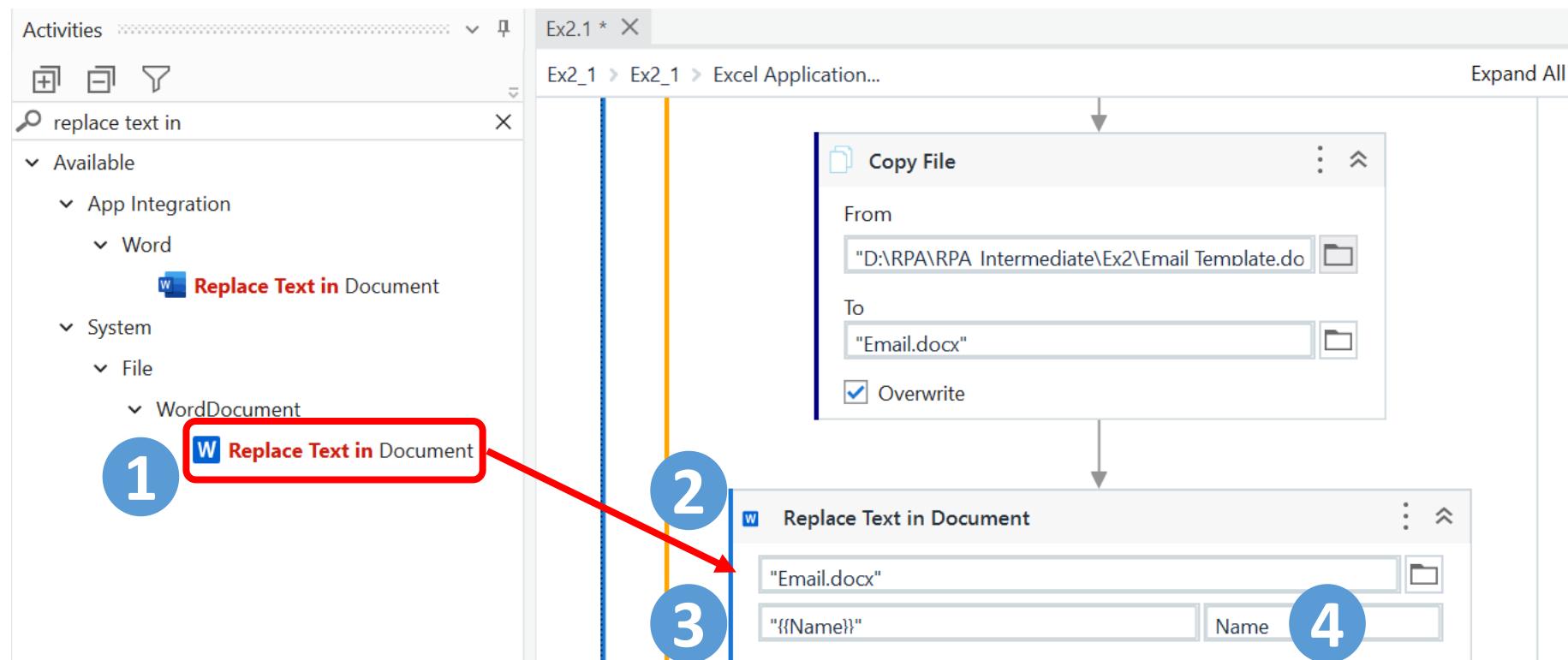
Exercise 8.2 (Step 3)

2. Use the Word email template (provided) as the source in the “Copy File” activity.
3. Set “Overwrite” to “Enabled”.
4. Use the output to “Email.docx” (or any name you prefer). This will be a temporary file, used for customising the email.



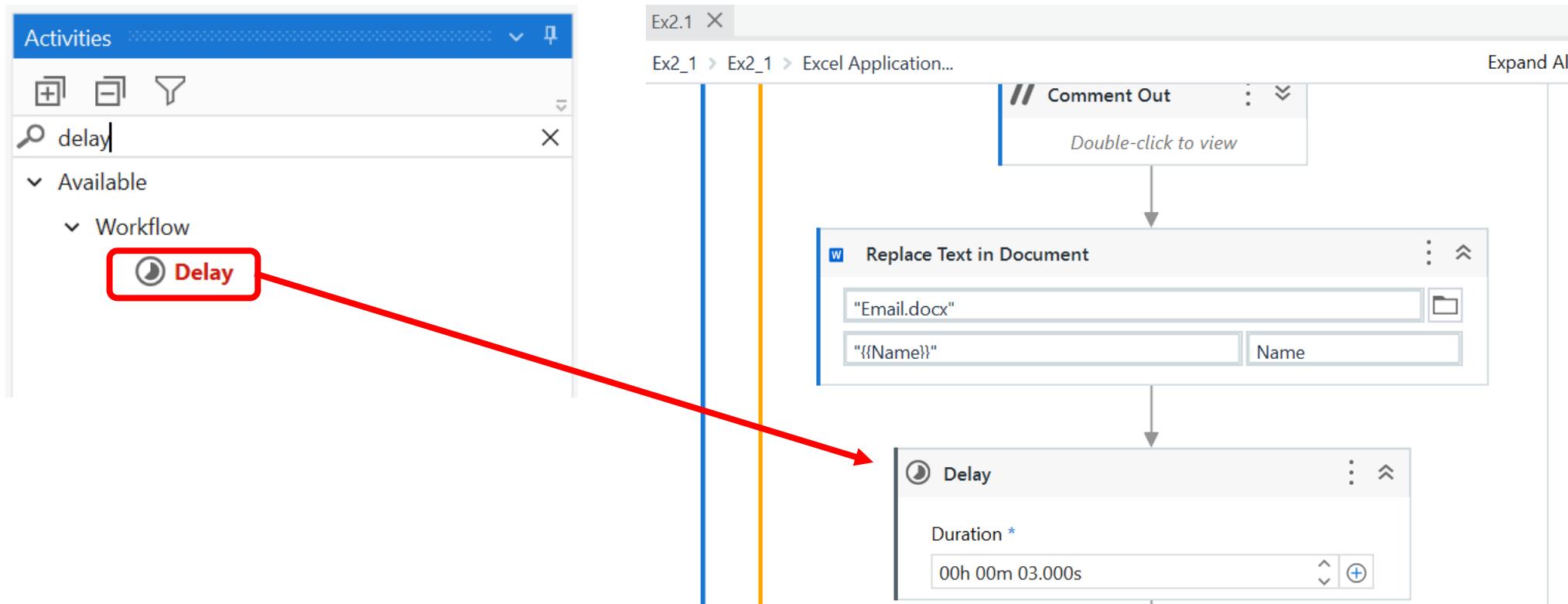
Exercise 8.2 (Step 4)

1. Look for “Replace Text in Document” and add it below “Copy File”
2. In “Replace Text...”, specify that “Email.docx” is what we are working with
3. The activity should look for “{{Name}}”
4. ... and replace that with the content of the variable Name.



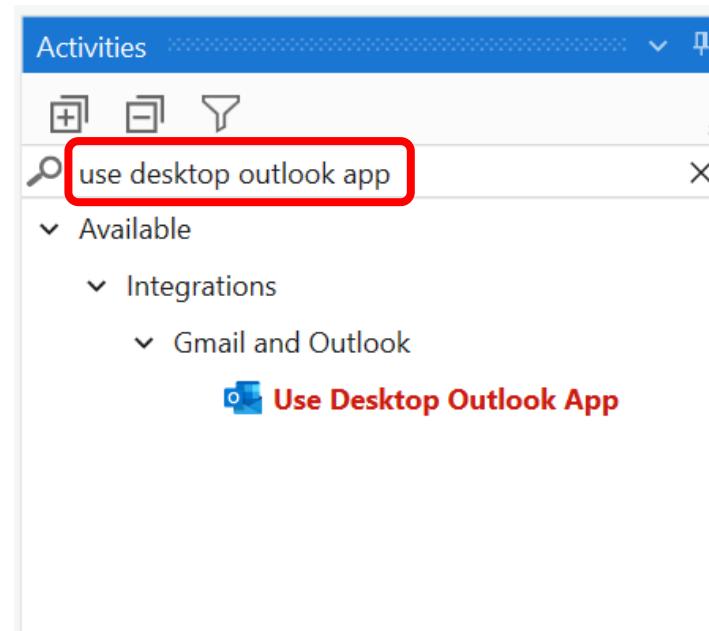
Exercise 8.2 (Step 5)

1. Insert an appropriate delay so that UiPath can save the Word file.



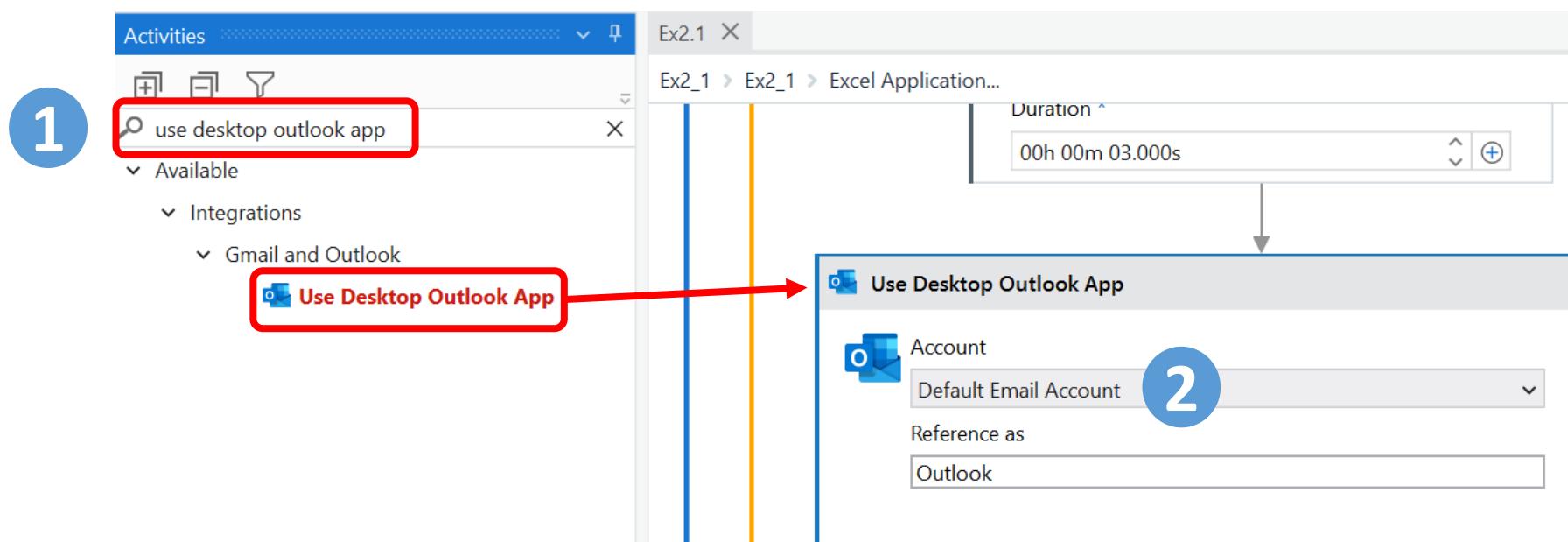
Exercise 8.2 (Step 6)

1. Look for “Use Desktop Outlook App” in Search



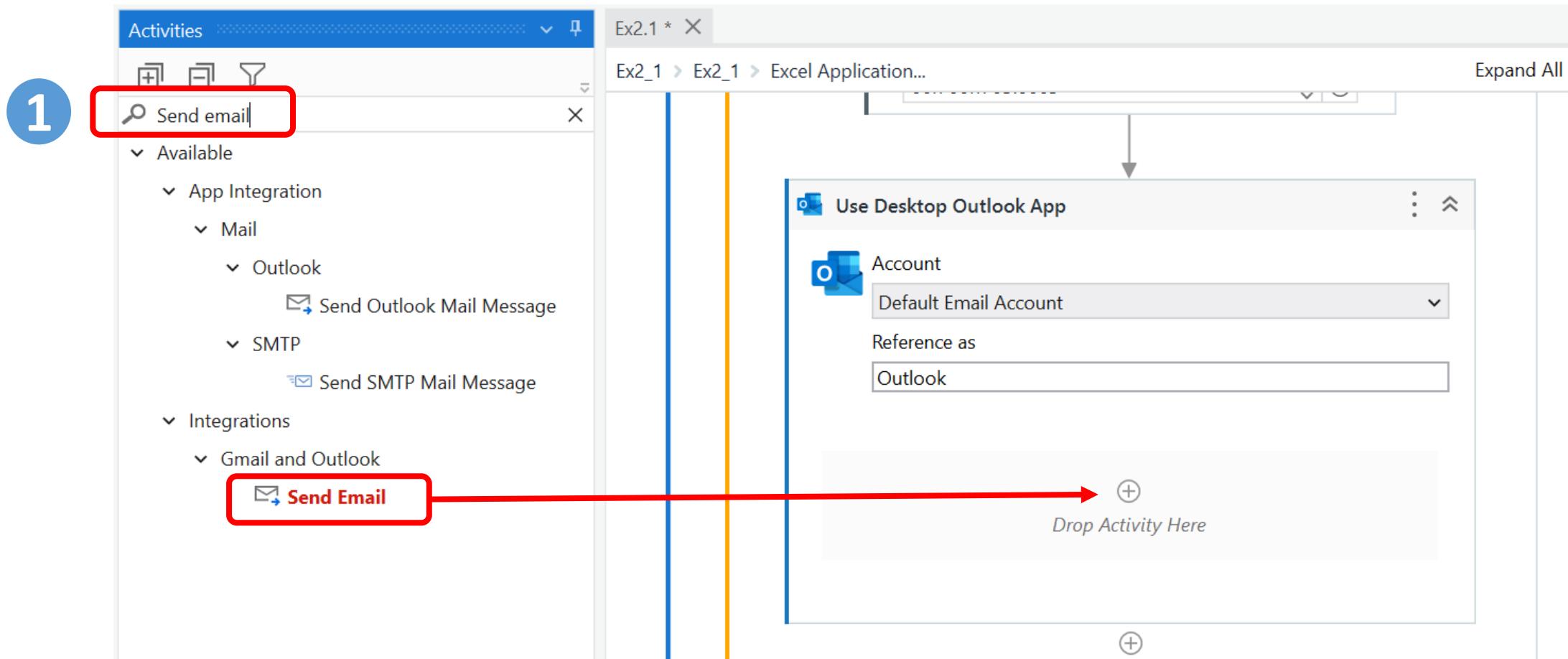
Exercise 8.2 (Step 7)

1. Add “Use Desktop Outlook App” below “Delay”.
2. Unless you have other email accounts connected to your Outlook client, it is recommended that “Default Email “Account” be used.



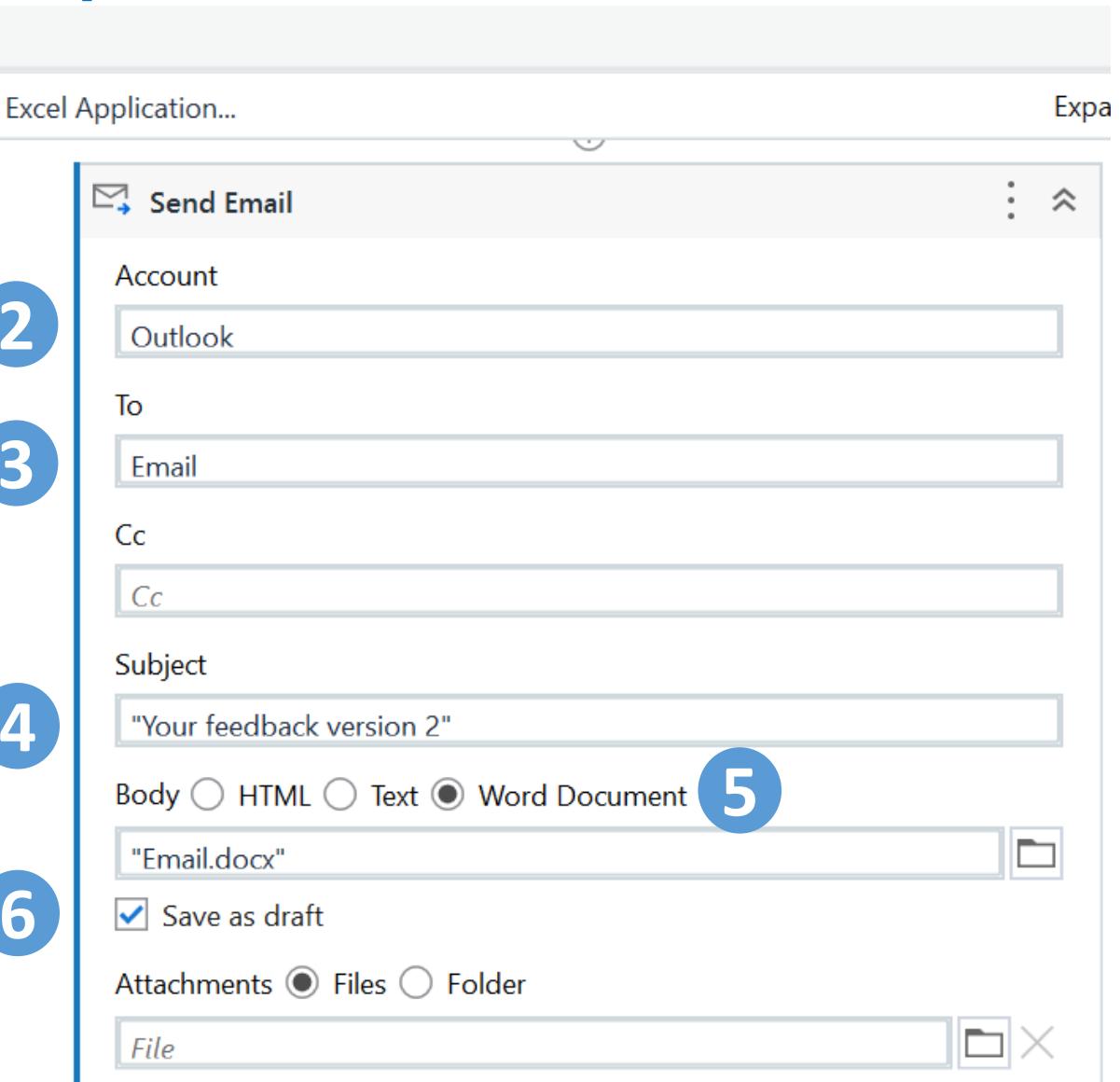
Exercise 8.2 (Step 8)

1. Look for “Send email” and add it into “Use Desktop Outlook App”



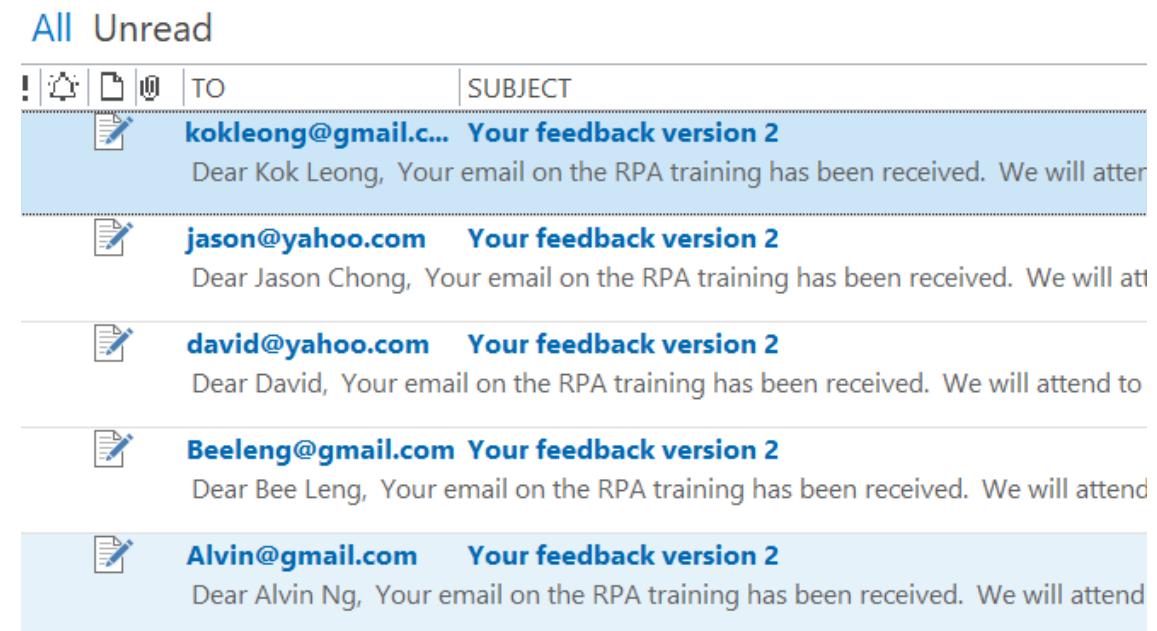
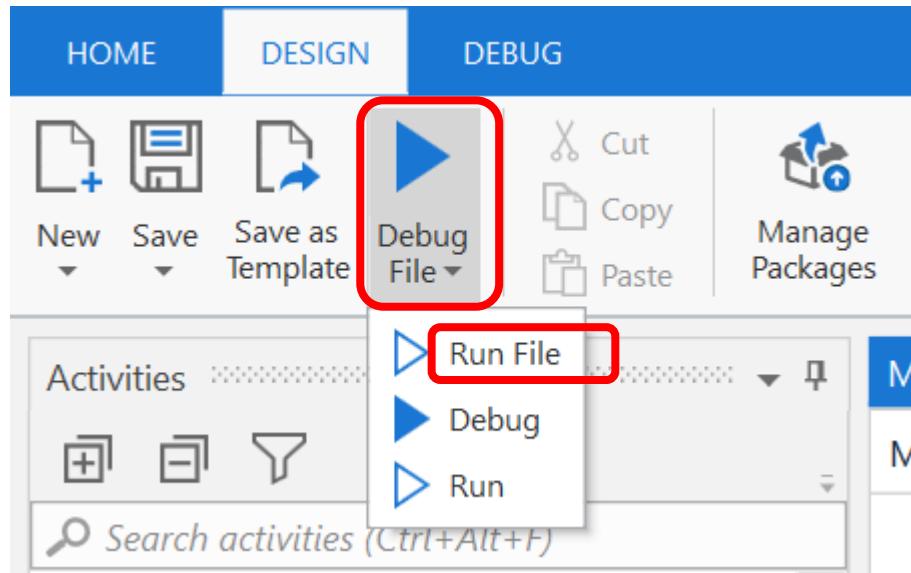
Exercise 8.2 (Step 8)

2. Set "Outlook" as the Account.
3. Use the content of the Email variable as the destination email.
4. Specify the subject
5. Use the Word document "Email.docx" for the email body.
6. Enable "Save as Draft". This means the email will be in Draft folder. If you do not tick this, it will be in Outbox (if you work offline in Outlook), or in Sent Items (if you did not work offline in Outlook).



Exercise 8.2 (Step 9)

Click the **Debug File** button and select “Run File” to run your script. Check your outlook email to confirm that the email has been created.





Ex9.1 Excel with VLookUp

1. Continue to use Robot4
2. Script Robot to perform an Excel VLOOKUP
3. Use “Join Data Tables” command with “Left join” or “Full join”
4. Use “Filter Tables”
5. Refer to Ex9.1 in “RPA UiPath Hands-On Guide” for step-by-step instruction



	A	B	C	D
1	Salesperson ID	Q1 Sales (\$ in thou)	Salesperson	Sales Team
2	63216781	81	63216781	Science
3	90001892	93	90001892	Business
4	56777777	99	56777777	Engineering
5	22266677	76	22266677	Science
6	63216779	95	63216779	Business
7	90001890	96	90001890	Engineering
8	22266675	82	22266675	Science
9	63216778	92	63216778	Science
10	90001889	90	90001889	Business
11	56777774	79	56777774	Engineering
12	63216776	89	63216776	Science
13	90001887	79	90001887	Science
14	56777772	77	56777772	Business
15	33888994	94	33888994	Engineering



Exercise 9.1

Performing VLOOKUP with RPA

Exercise 9.1

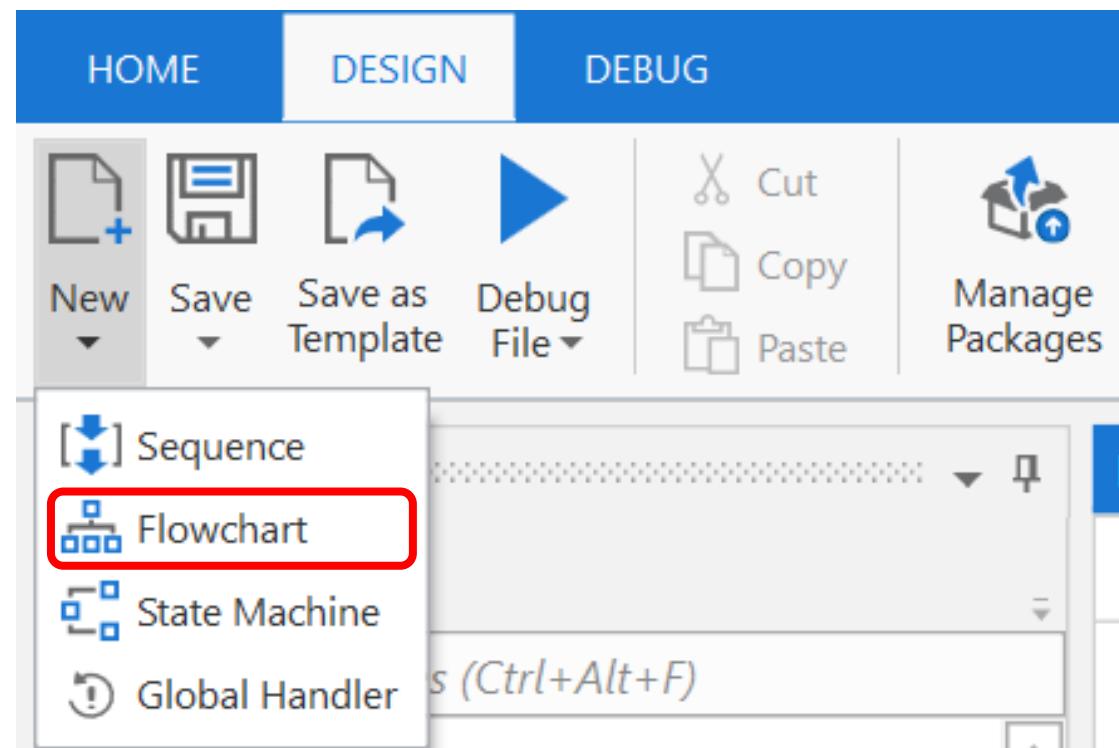
This exercise shows how to use RPA to perform the equivalent of VLOOKUP in Excel.

What you need:

- Excel file, “**Textbook_Sales.xlsx**”
- Excel file, “**Joined.xlsx**” (blank file)

Exercise 9.1 (Step 1)

In UiPath Studio, create a new flowchart, and name it “Ex9.1”.



Exercise 9.1 (Step 2)

Create the following variables.

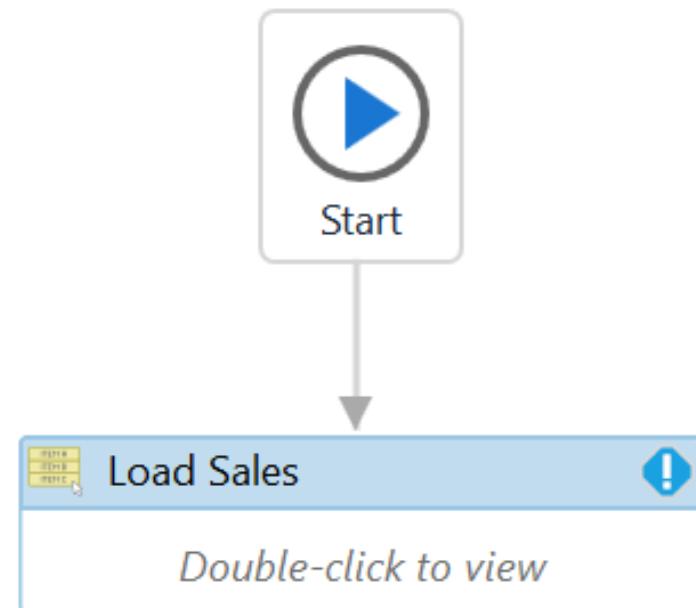
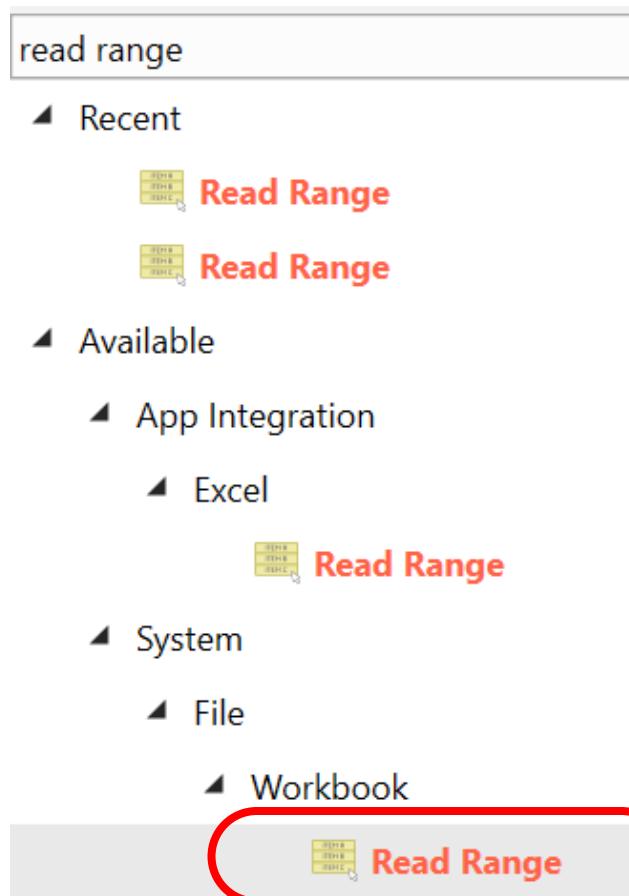
Name	Variable type	Scope
SalesTable	DataTable	Ex3_3
SalesTeamTable	DataTable	Ex3_3
SalesByTeamTable	DataTable	Ex3_3

Remark

1. SalesTable shows the sales made by each Salesperson (input)
2. SalesTeamTable shows the SalesTeam that each Salesperson belongs to (input)
3. SalesByTeamTable shows the sales made by each SalesTeam (output)

Exercise 9.1 (Step 3)

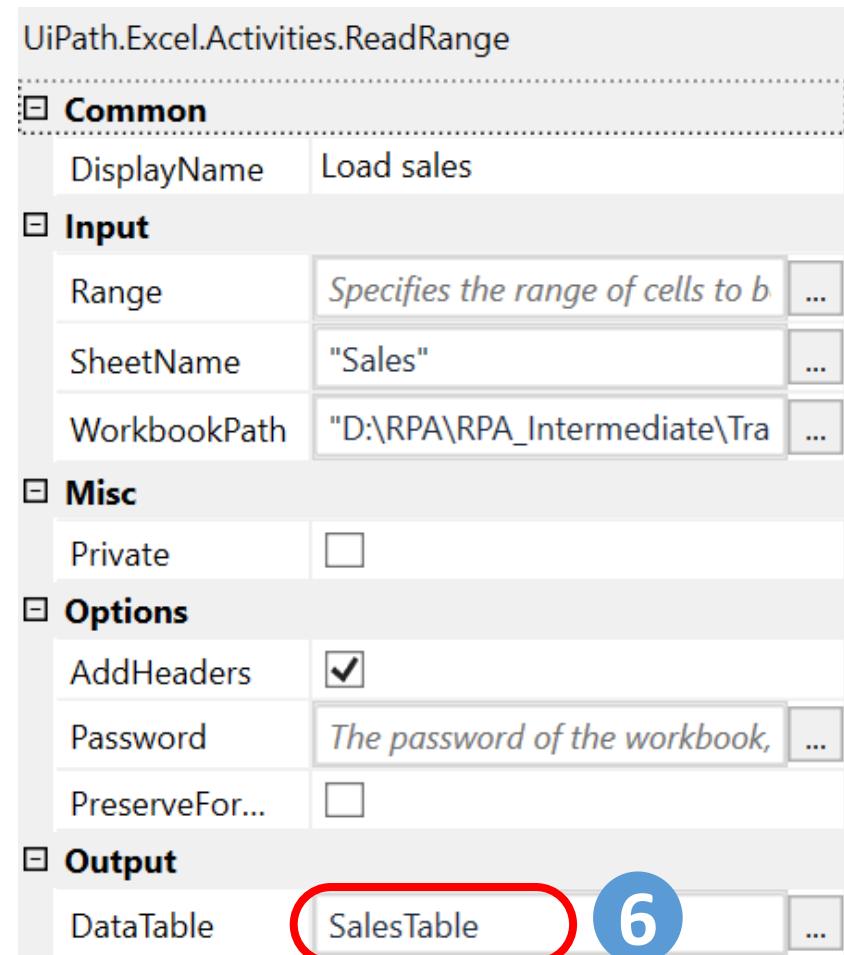
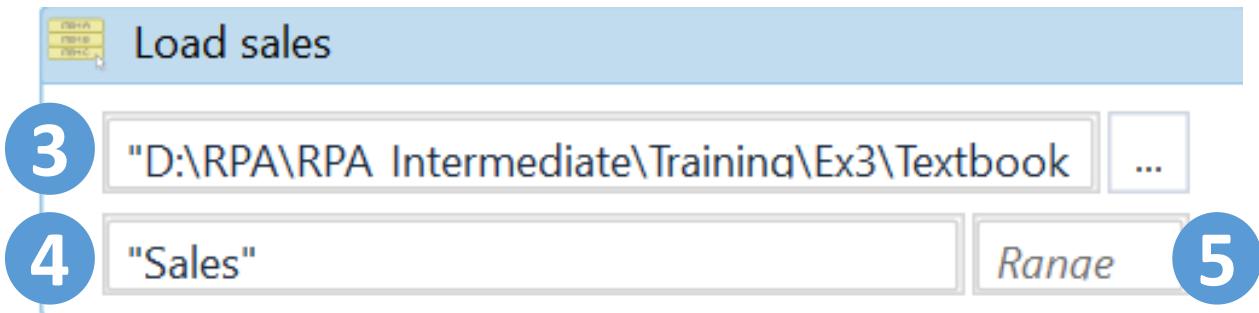
1. Search for “**Read Range**” and drag into the Designer Panel. Rename as “Load Sales”.
2. Right-click on “**Read Range**”, and select “Set as Start Node”.



To read sales value *from the “Sales” worksheet* into “**SalesTable**”

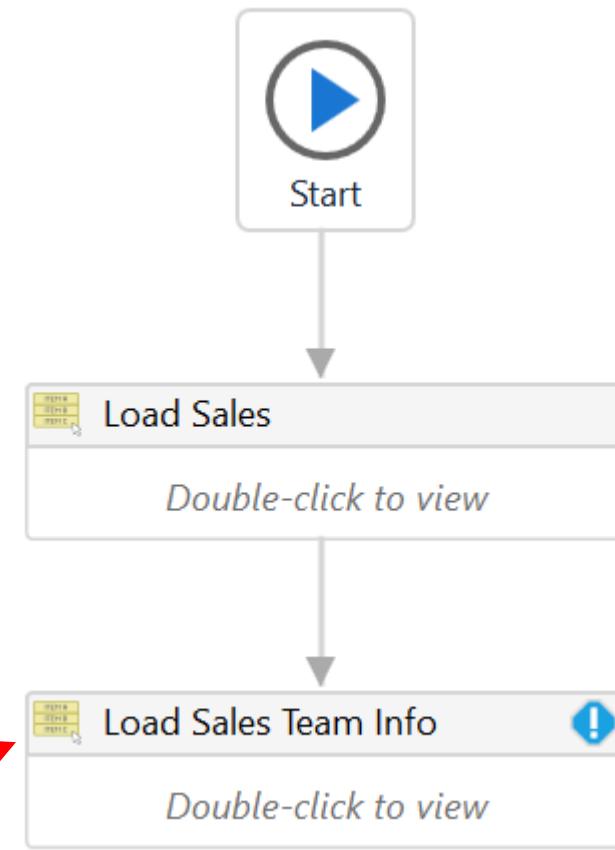
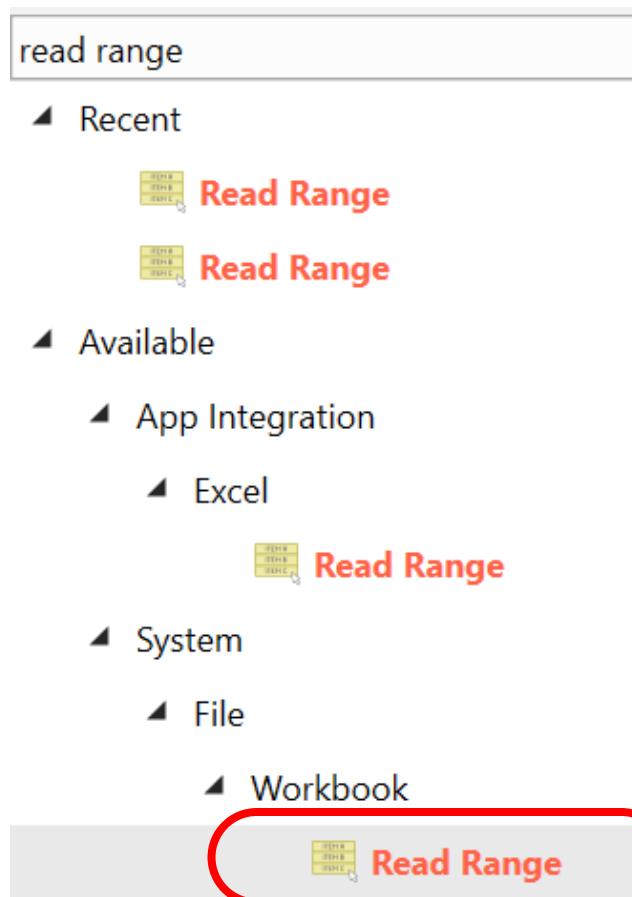
Exercise 9.1 (Step 3)

3. Input workbook path of Excel file, “Textbook_Sales.xlsx”.
4. Change “Sheet1” to “Sales”.
5. Leave range empty.
6. In the properties panel, under the Output section, specify ”SalesTable”.



Exercise 9.1 (Step 4)

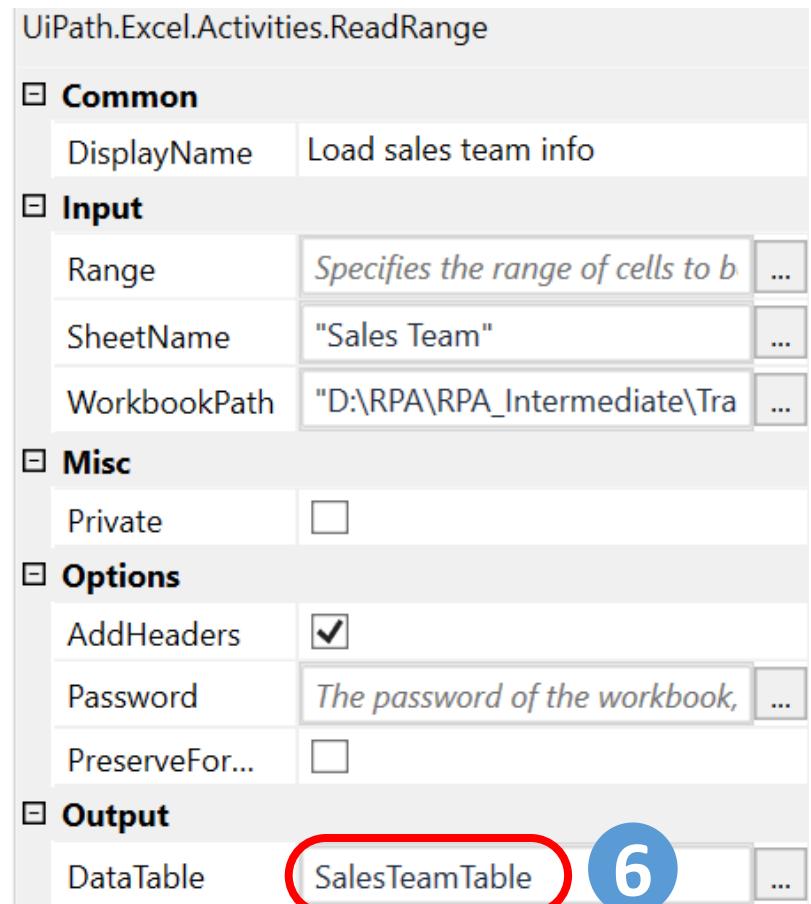
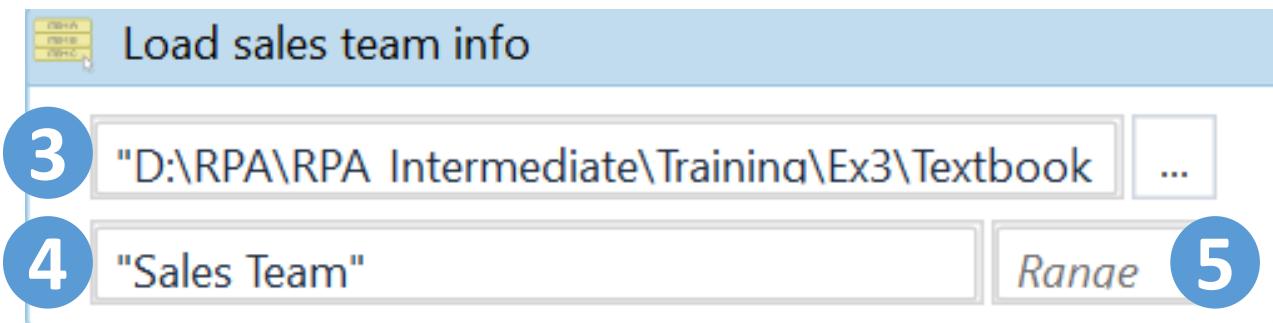
1. Search for “Read Range” and drag into the Designer Panel. Rename as “Load Sales Team Info”.
2. Join to “Load Sales”.



To read sales team info *from the “Sales Team” worksheet* into **“SalesTeamTable”**

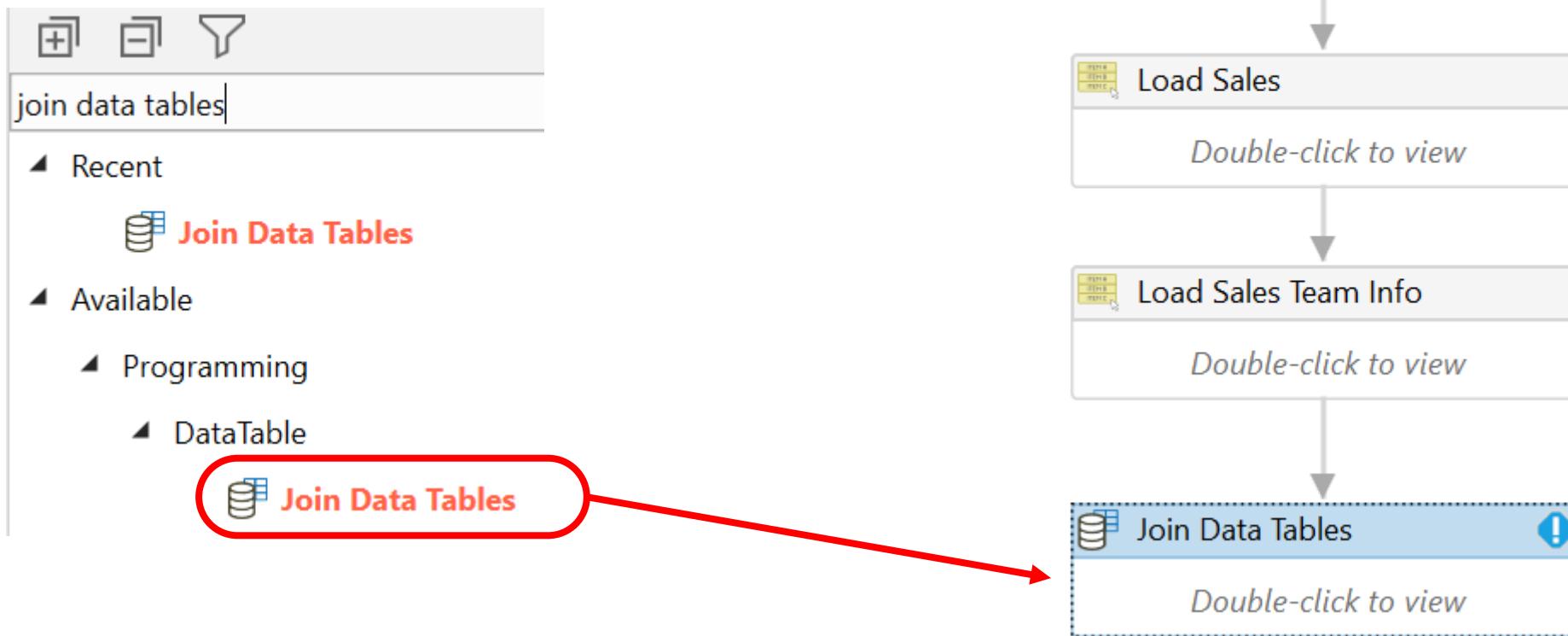
Exercise 9.1 (Step 4)

3. Input workbook path of Excel file, “**Textbook_Sales.xlsx**”.
4. Change “Sheet1” to “Sales Team”.
5. Leave range empty.
6. In the properties panel, under the Output section, specify **”SalesTeamTable”**.



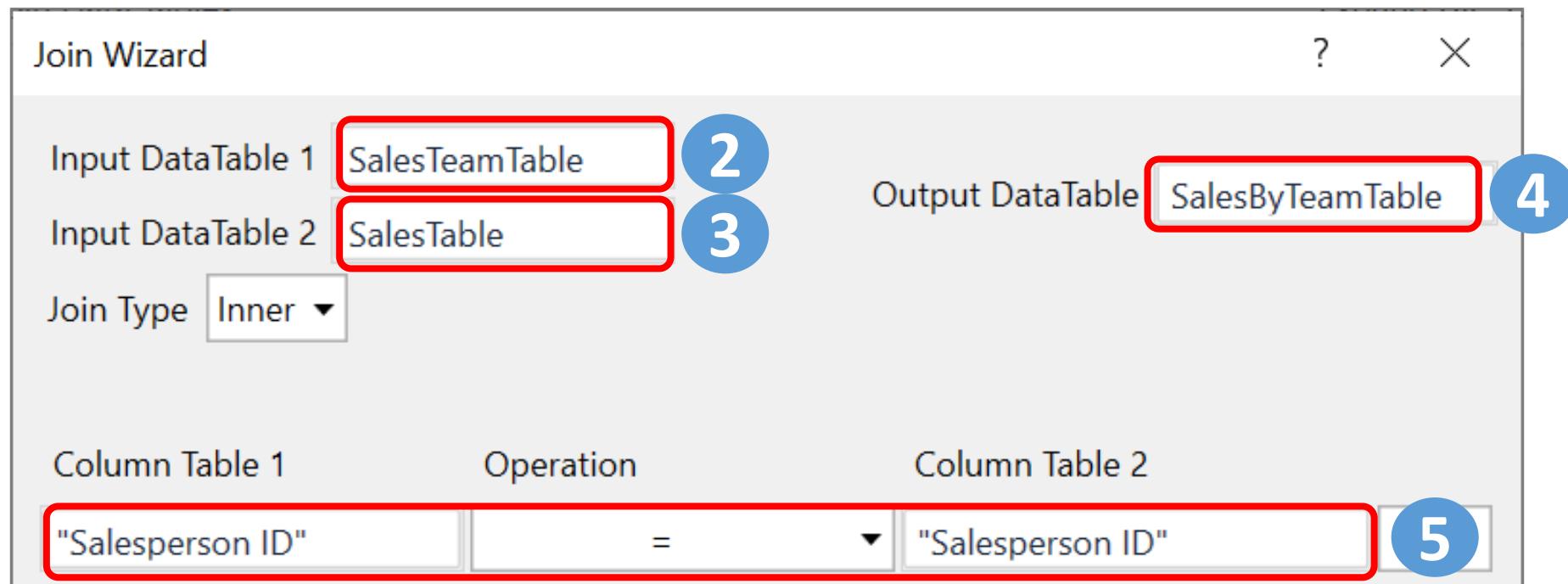
Exercise 9.1 (Step 5)

1. Search for “Join Data Tables” and drag into the Designer Panel.
2. Join to “Load Sales Team Info”.



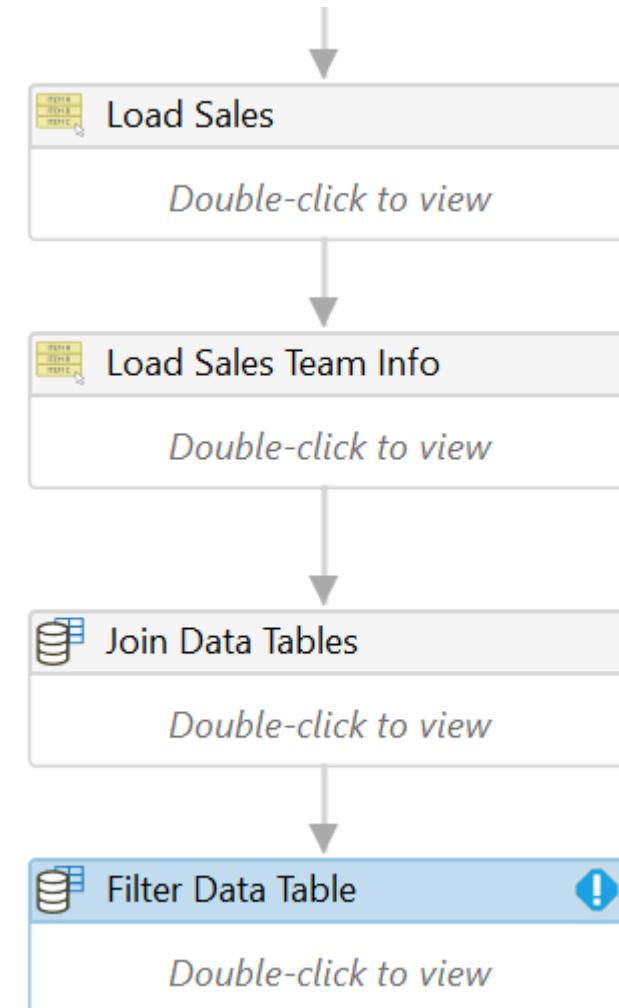
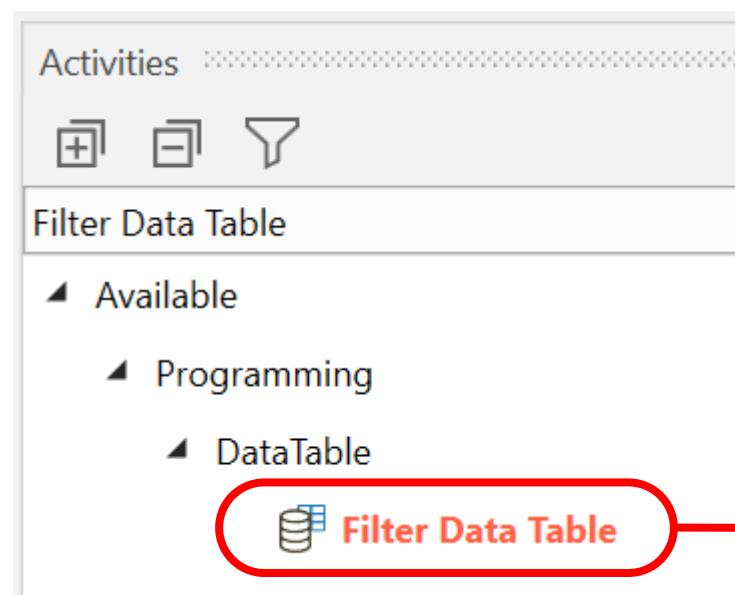
Exercise 9.1 (Step 6)

1. Double-click to view “Join Data Tables” activity and click “Join Wizard”.
2. Update “Input DataTable 1” as “SalesTeamTable”.
3. Update “Input DataTable 2” as “SalesTable”.
4. Update “Output DataTable” as “SalesByTeamTable”.
5. Update “Salesperson ID” in Column Table 1 as “=” to “Salesperson ID” in Column Table 2. Click OK to apply settings.



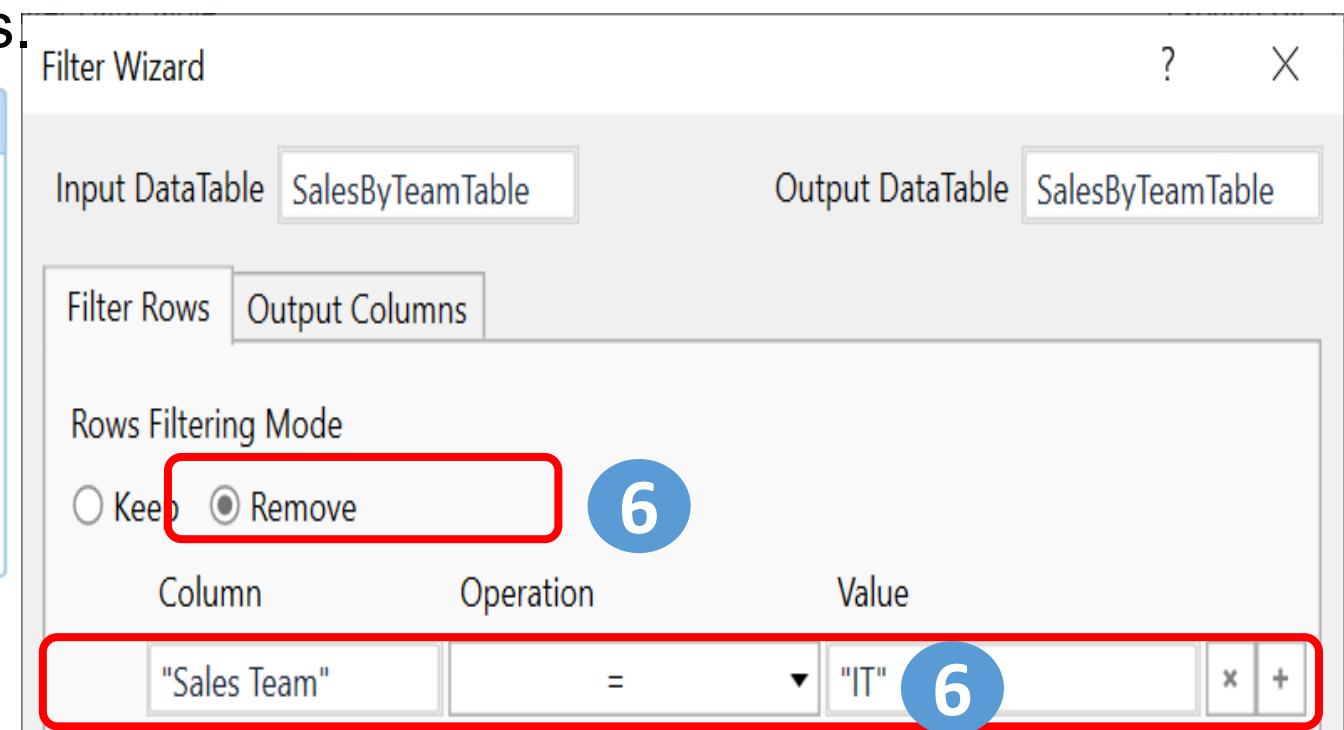
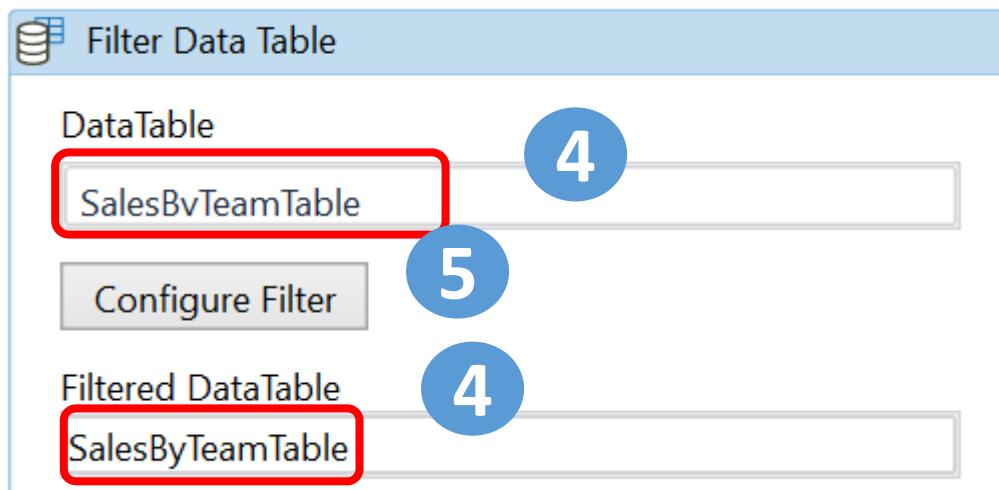
Exercise 9.1 (Step 7)

1. Search for “Filter Data Table” and drag into the Designer Panel.
2. Join to “Join Data Tables”.



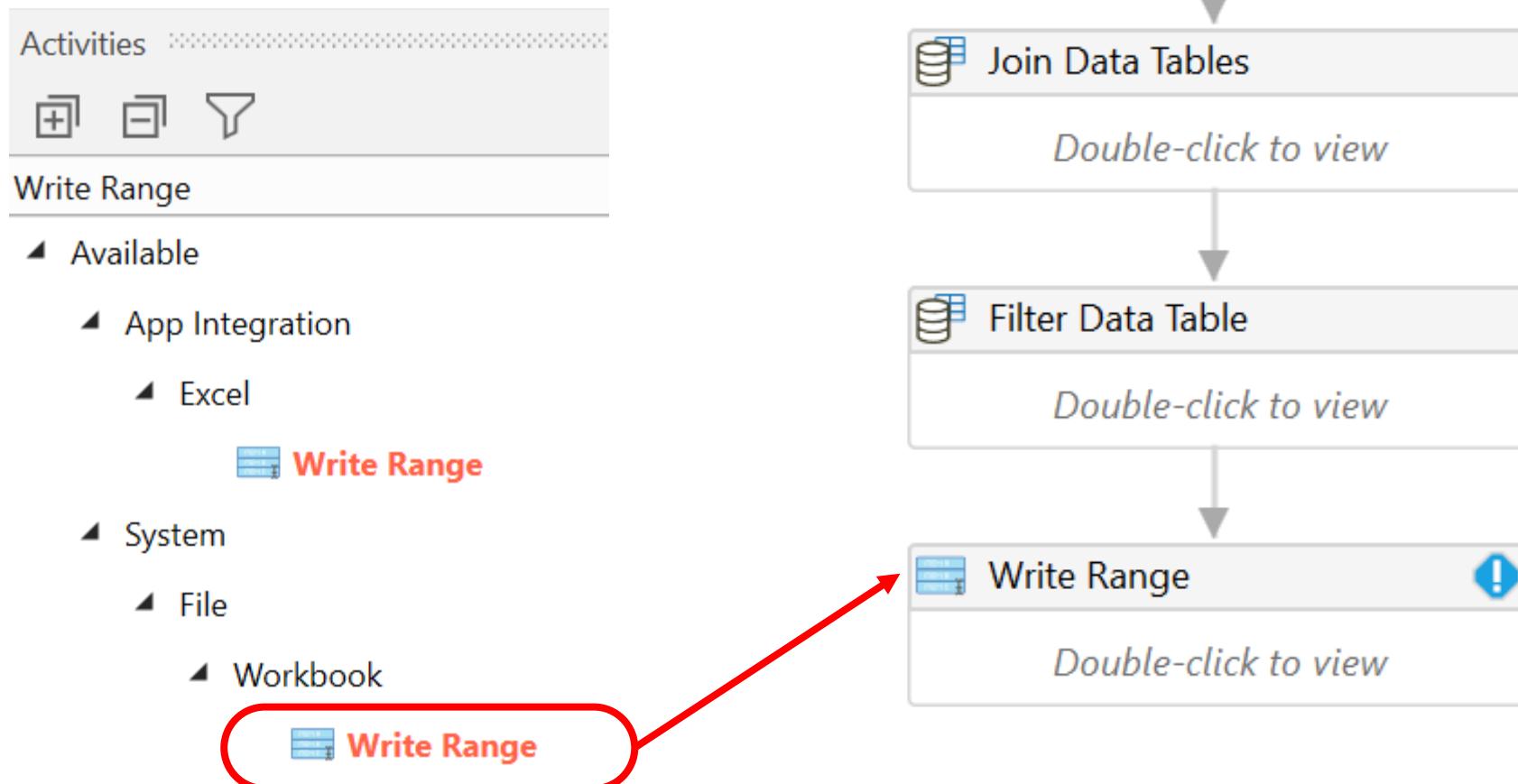
Exercise 9.1 (Step 7)

3. Double-click to view “**Filter Data Table**” activity.
4. Update both “**DataTable**” and “**Filtered Data Table**” as “**SalesByTeamTable**”.
5. Click “**Configure Filter**”.
6. Under “**Filter Rows**”, click “**Remove**” and update “**Sales Team**” = “**IT**”.
Click OK to apply settings.



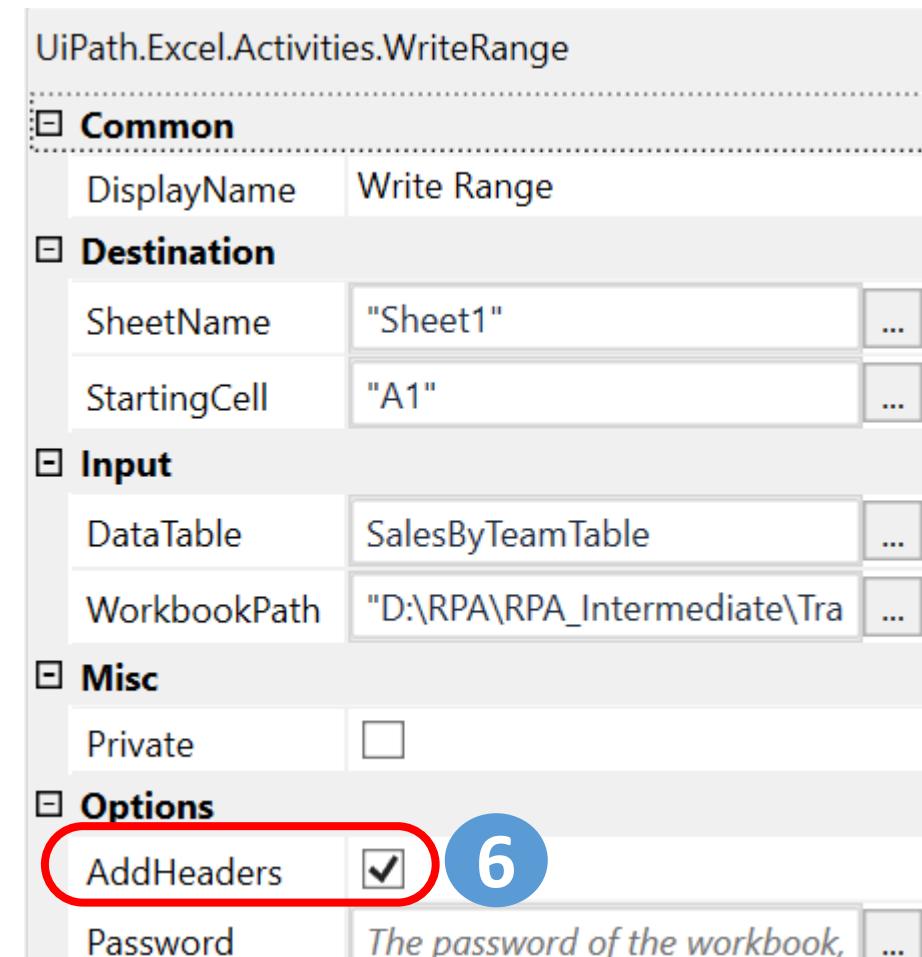
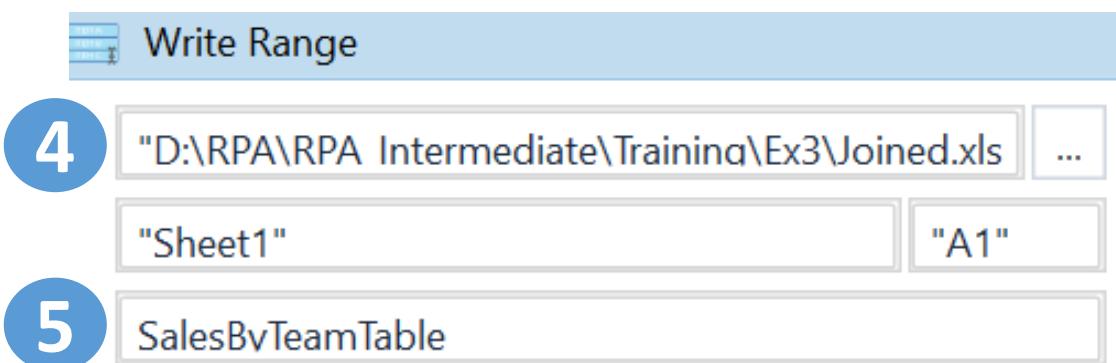
Exercise 9.1 (Step 8)

1. Search for “Write Range” and drag into the Designer Panel.
2. Join to “Filter Data Table”.



Exercise 9.1 (Step 8)

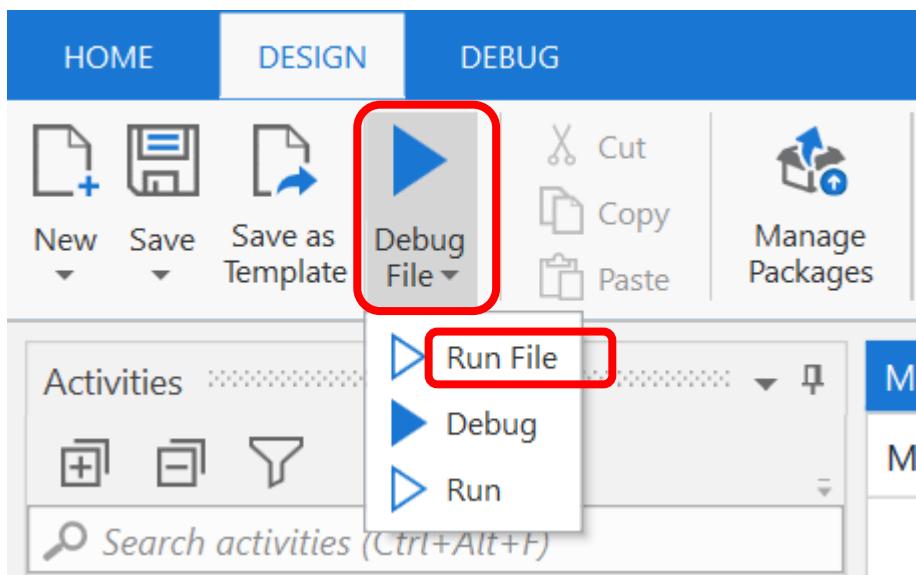
3. Double-click to view “**Write Range**” activity.
4. Input workbook path of Excel file, “**Joined.xlsx**”.
5. Update “**DataTable**” to “**SalesByTeamTable**”.
6. In the properties panel, under the Options section, check off “**AddHeaders**”



Exercise 9.1 (Step 9)

Click the **Debug File** button and select “Run File” to run your script.

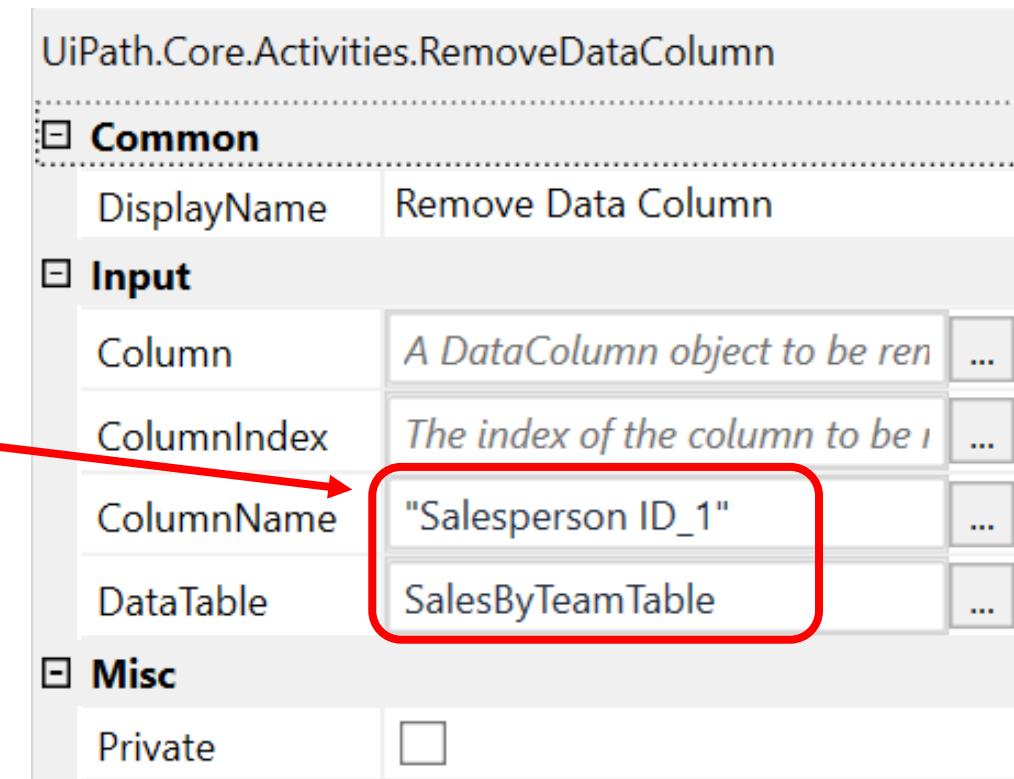
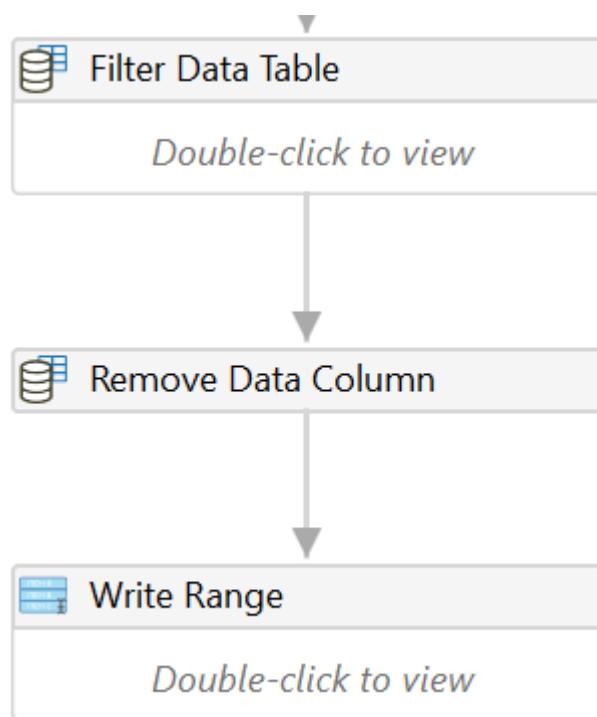
Check your output file “Joined.xlsx” to confirm the file is correct.



A	B	C	D
1 Salesperson ID	Sales Team	Salesperson ID_1	Q1 Sales (\$ in thousands)
2 63216781	Science	63216781	81
3 90001892	Business	90001892	93
4 56777777	Engineering	56777777	99
5 22266677	Science	22266677	76
6 63216779	Business	63216779	95
7 90001890	Engineering	90001890	96
8 22266675	Science	22266675	82
9 63216778	Science	63216778	92
10 90001889	Business	90001889	90
11 56777774	Engineering	56777774	79
12 63216776	Science	63216776	89
13 90001887	Science	90001887	79
14 56777772	Business	56777772	77
15 33888994	Engineering	33888994	94

Exercise 9.1 (Additional Steps)

1. Use “Remove Data Column” to delete “Salesperson ID_1” before using “Write Range” to output the results to Excel.
2. In the properties panel, under the Input section, update the “ColumnName” and “DataTable” accordingly.



Exercise 9.1 (Review)

1. Try using the “**Join Data Tables**” command with “Left join” or “Full join”
2. Experiment with “**Filter Tables**” using different settings.

Exercise 9.2 (optional)

Creating an Excel Pivot Table with RPA

Exercise 9.2

This exercise shows how to use RPA to create an Excel pivot table.

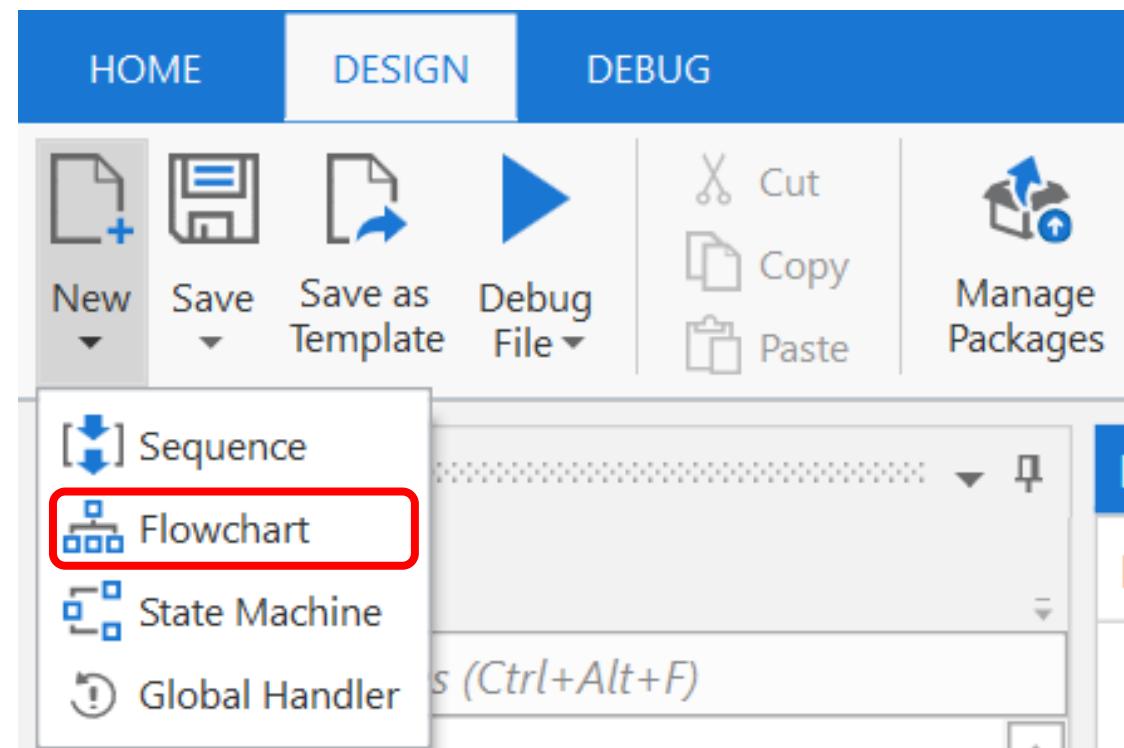
What you need:

- Excel file, “**Joined.xlsx**” (created in Ex 3.1) – Open this file and delete the column “Salesperson ID”.

A	B	C	D	E	F
1	Salesperso	Sales Tear	Q1 Sales (\$ in thousands)		
2	63216781	Science	81		
3	90001892	Business	93		
4	56777777	Engineerir	99		
5	22266677	Science	76		
6	63216779	Business	95		
7	90001890	Engineerir	96		
8	22266675	Science	82		
9	63216778	Science	92		
10	90001889	Business	90		
11	56777774	Engineerir	79		
12	63216776	Science	89		
13	90001887	Science	79		
14	56777772	Business	77		
15	33888994	Engineerir	94		
16					
17					
18					

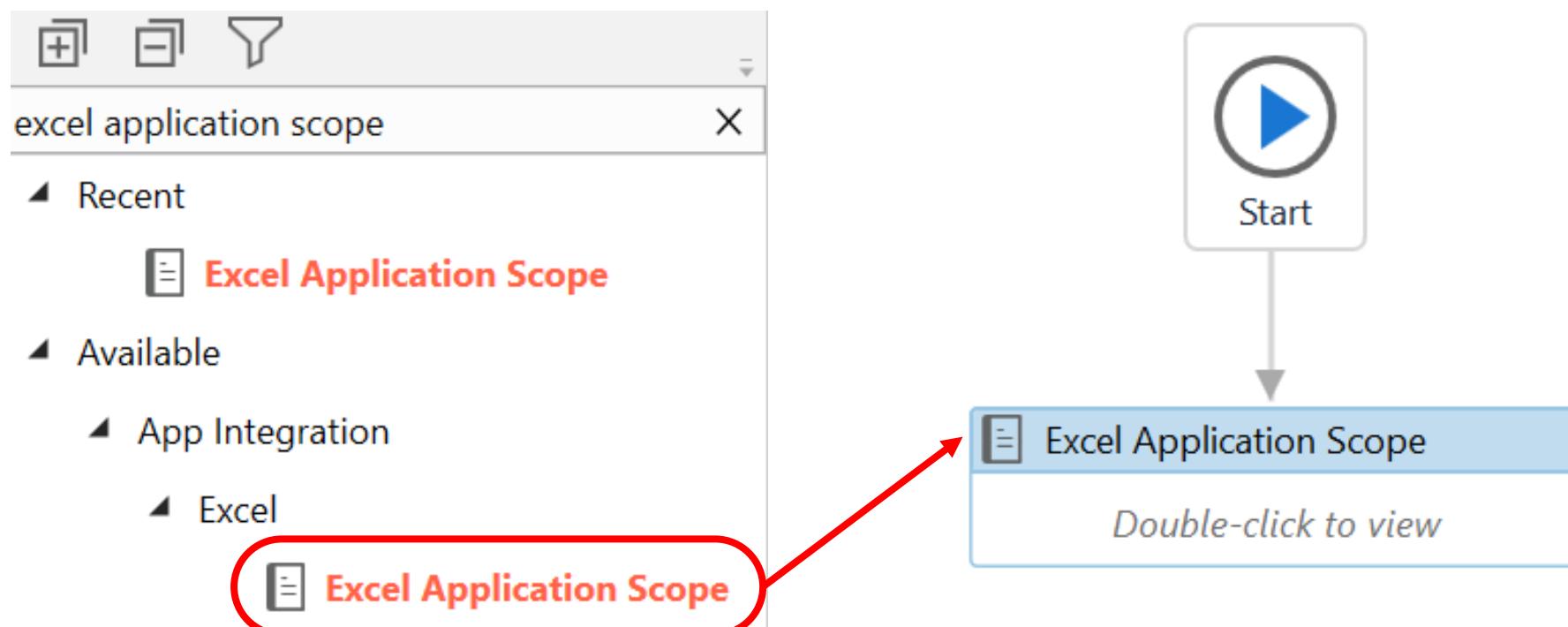
Exercise 9.2 (Step 1)

In UiPath Studio, create a new flowchart, and name it “Ex9.2”.



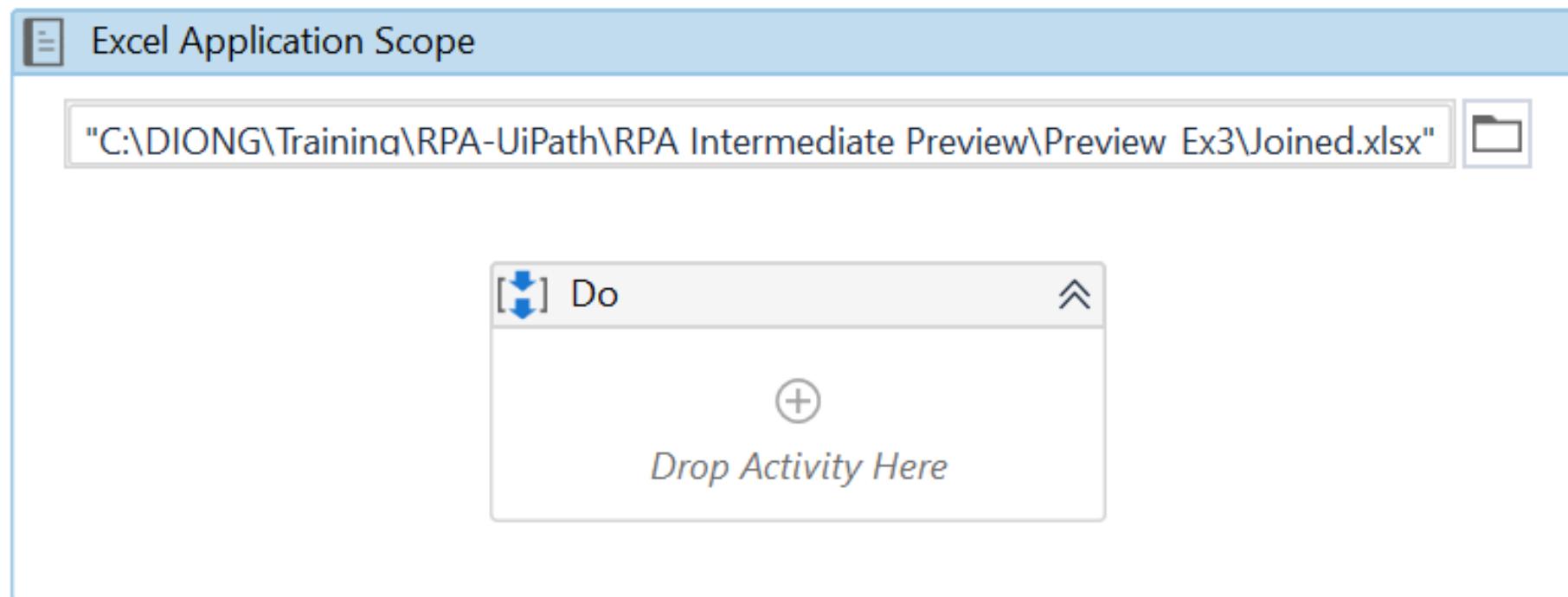
Exercise 9.2 (Step 2)

1. Search for “Excel Application Scope” and drag into the Designer Panel.
2. Right-click on “Excel Application Scope”, and select “Set as Start Node”.



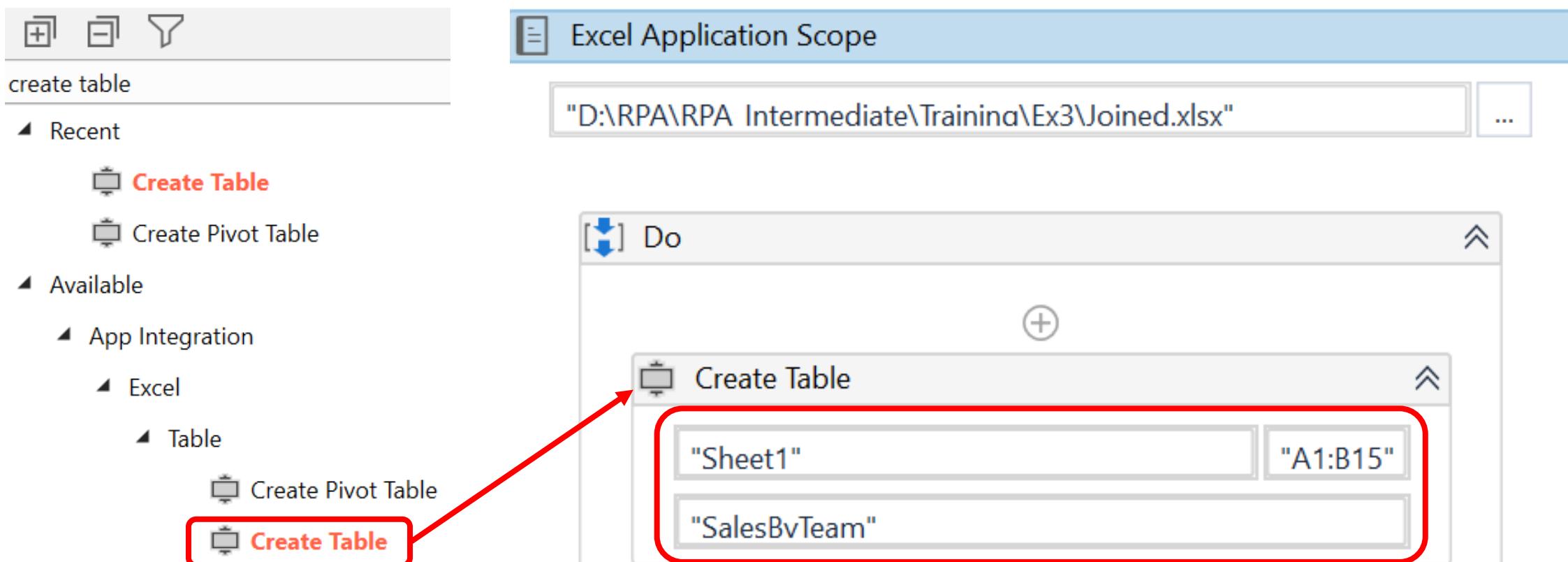
Exercise 9.2 (Step 3)

1. Double-click to view the “**Excel Application Scope**” activity.
2. Input workbook path of Excel file, “**Joined.xlsx**” (created in Ex 9.1).



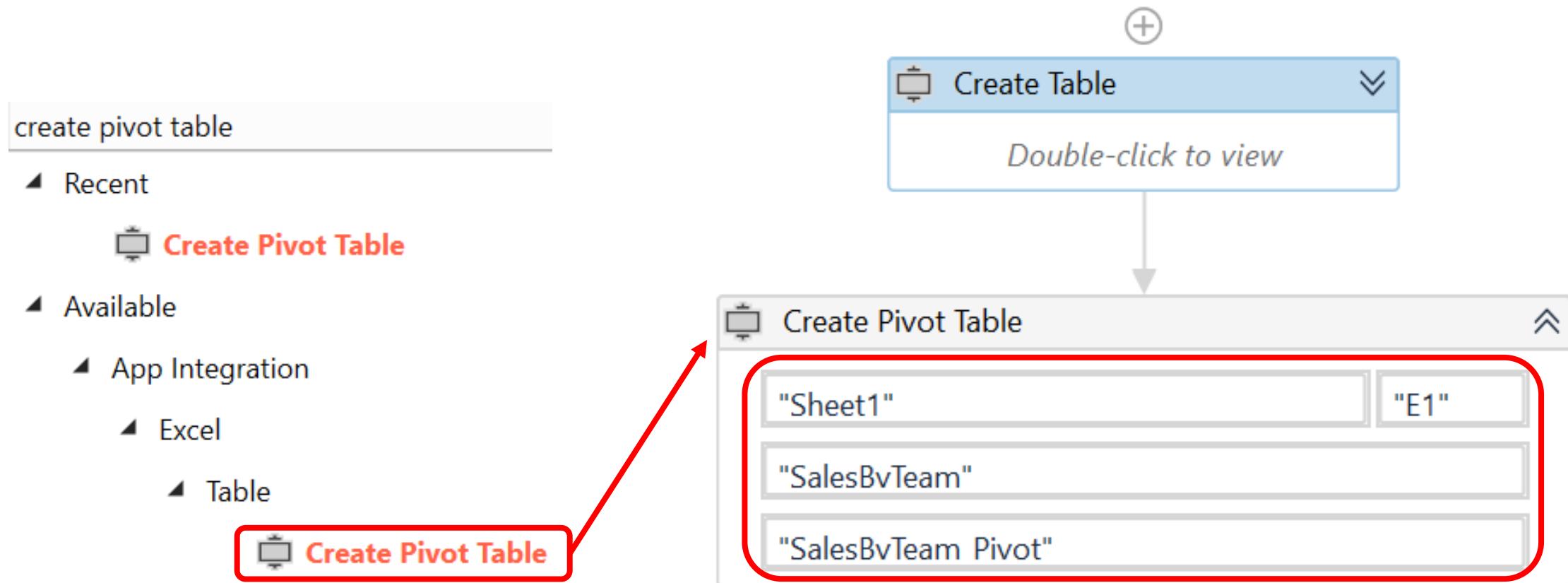
Exercise 9.2 (Step 3)

3. In the “Do” window of the “Excel Application Scope” activity, insert a “Create Table” activity to create an Excel table named “SalesByTeam”.



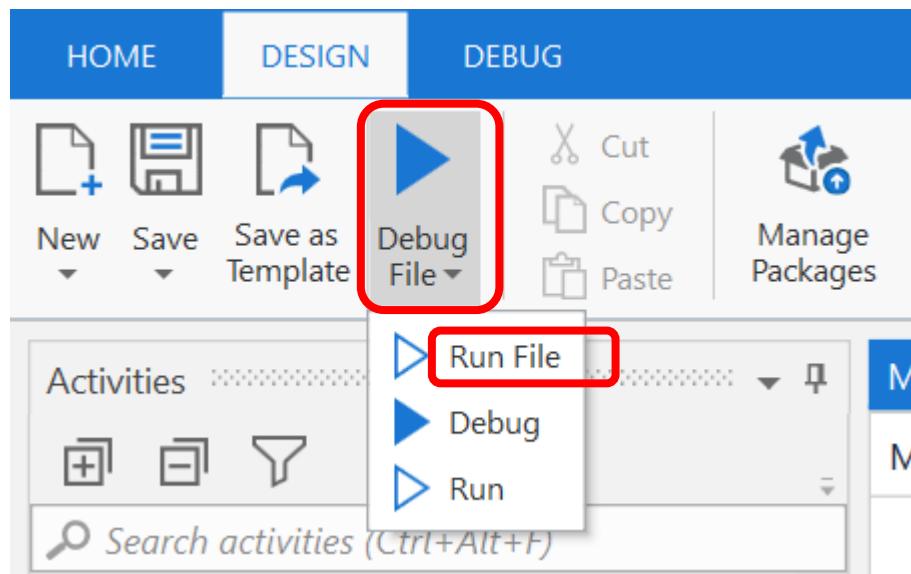
Exercise 9.2 (Step 3)

3. Insert a “Create Pivot Table” activity and join to “Create Table” to create an Excel table named “SalesByTeam_Pivot”.



Exercise 9.2 (Step 4)

Click the **Debug File** button and select “Run File” to run your script. Check your output file “Joined.xlsx” to confirm the file is correct.



Exercise 9.2 (Step 4)

Your output should look like this:

The screenshot shows a Microsoft Excel spreadsheet with a green header bar containing the ribbon tabs: File, Home, Insert, Page Layout, Formulas, Data, Review, View, Developer, Design, and Tell me what you want. The 'Design' tab is selected. The active cell is A3, and the formula bar shows 'Business'. The main area contains a pivot table with the following data:

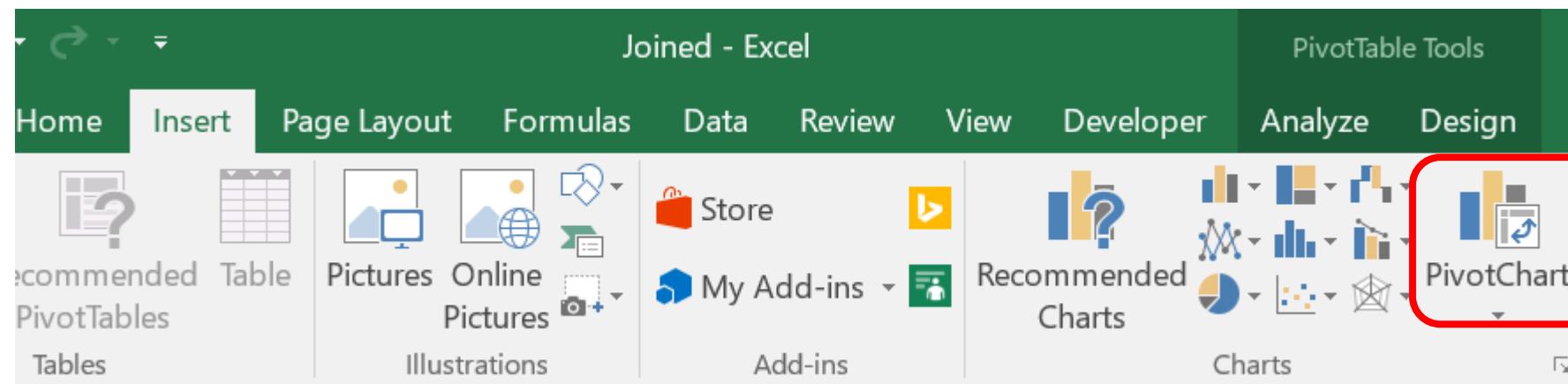
	A	B	C	D	E	F
1	Sales Team	Q1 Sales (\$ in thousands)		Row Labels	Sum of Q1 Sales (\$ in thousands)	
2	Science		81	Business		355
3	Business		93	Engineering		368
4	Engineering		99	Science		499
5	Science		76	Grand Total		1222
6	Business		95			
7	Engineering		96			
8	Science		82			
9	Science		92			
10	Business		90			
11	Engineering		79			
12	Science		89			
13	Science		79			
14	Business		77			
15	Engineering		94			

Exercise 9.2 (Additional Steps)

1. Click on the pivot table.
2. Click “Insert” and then “Pivot Chart”.
3. Use the wizard to produce the desired chart.

1 Row Labels ▾ Sum of Q1 Sales (\$ in thousands)

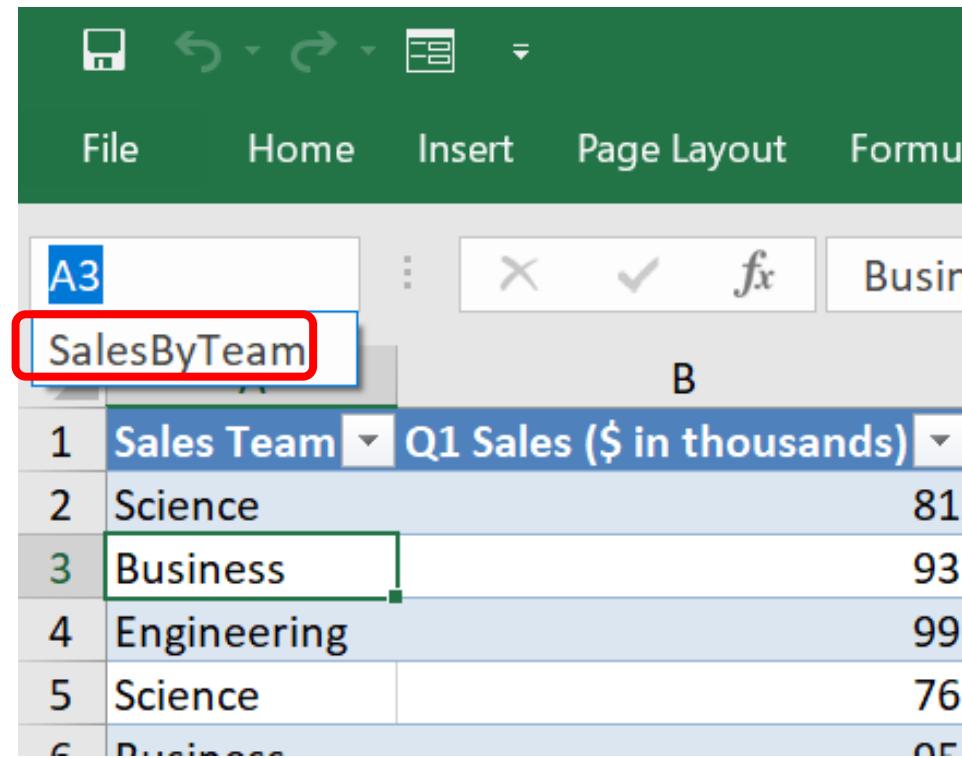
Business	355
Engineering	368
Science	499
Grand Total	1222



Exercise 9.2 (Review)

The “**Create Table**” step was needed to create an Excel variable “SalesByTeam”.

Excel will pick inputs for the pivot table using this variable name.



A screenshot of the Microsoft Excel ribbon interface. The ribbon tabs visible are File, Home, Insert, Page Layout, and Formulas. The active cell is A3, which contains the text "SalesByTeam". This cell is highlighted with a red rectangular border. Below the ribbon, the formula bar shows the same text "SalesByTeam". The main workspace displays a PivotTable with the following data:

	Sales Team	Q1 Sales (\$ in thousands)
1	Science	81
2	Business	93
3	Engineering	99
4	Science	76
5	Business	05

Exercise 10a

Introduction to Strings



Exercise 10a

- A string is an sequence of alphabets, digits or special characters such as commas or dashes
- We will learn how to do string manipulation such as
 - a) Joining a few string together
 - b) Counting the length of the string
 - c) Identifying the position of each substring within a string
 - d) Extracting the desired portion of the substring
 - e) Replacing the string
 - f) Splitting the string



Ex10a – Ex 10e PDF – extracting info

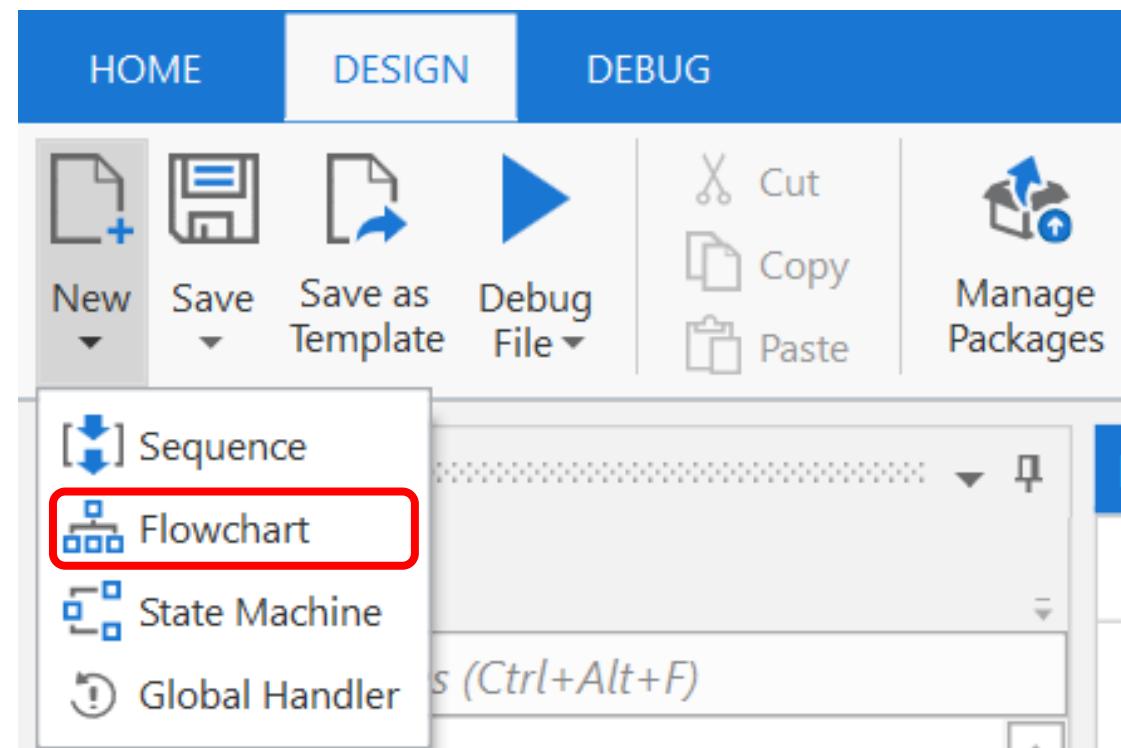
1. Create a new Process and name it “*Robot5_XXXXXX*” with description as “*RPA Lesson7 Ex10a to Ex10e*
XXXXXX is your first name, e.g. *Jun_Hao* or *Jeremy* etc
2. Script Robot to read PDF files.
3. Refer to Ex10a to 10e in “RPA UiPath Hands-On Guide” for step-by-step instruction



	A	B	C	D	E	F	G
1	Seller	Invoice_Ref	Invoice_Date	Amt_Due	Due_Date	Bill_To	Invoice_Purpose
2	Acme Inc	A100	7/3/2020	\$200.00	21/3/2020	John Tan	Catered Food
3							

Exercise 10a (Step 1)

In UiPath Studio, create a new flowchart, and name it “Ex10a”.



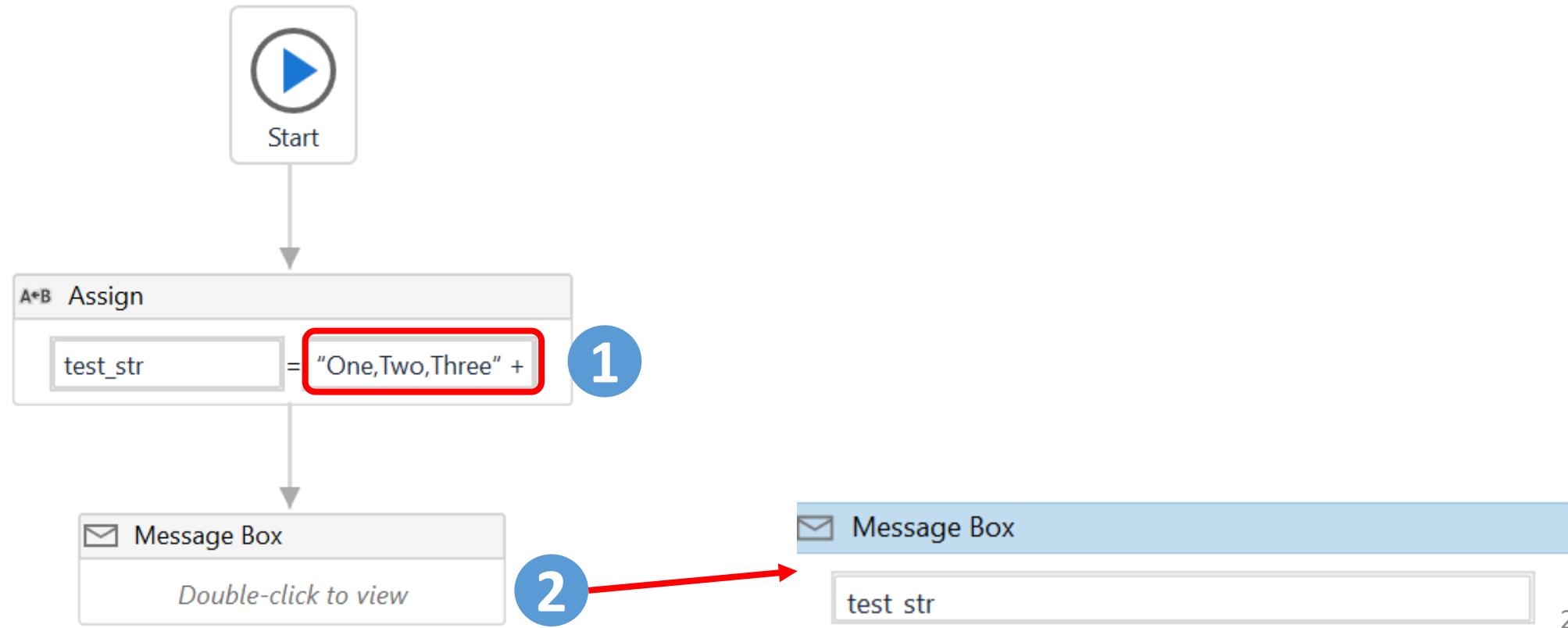
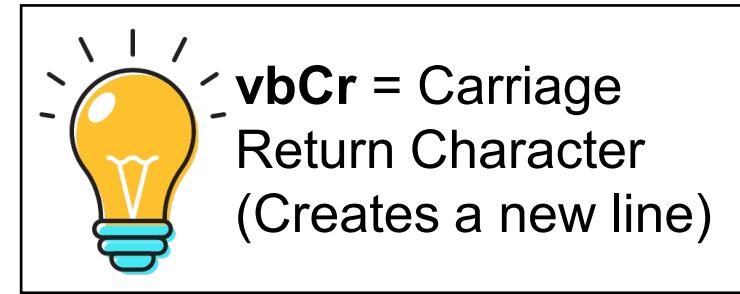
Exercise 10a (Step 2)

Create the following variables:

Name	Variable type	Scope
test_str	String	Ex4_0
int_count	Int32	Ex4_0

Exercise 10a (Step 3)

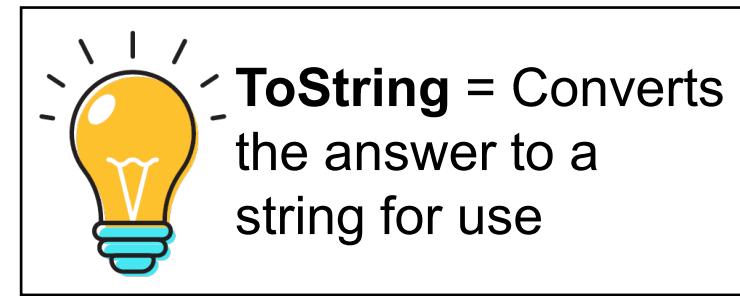
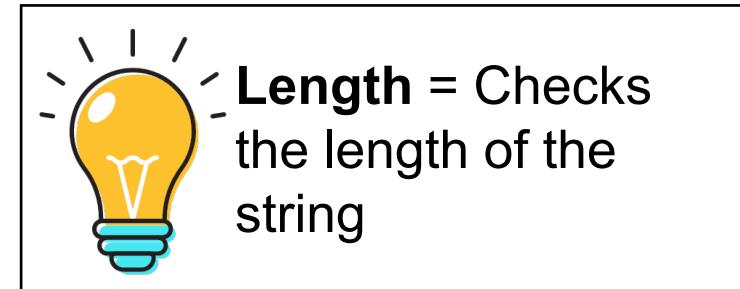
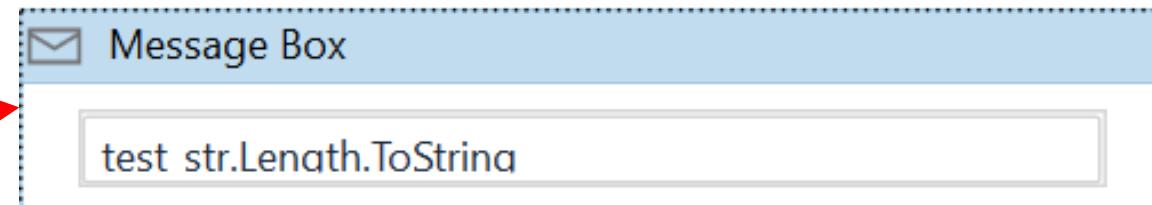
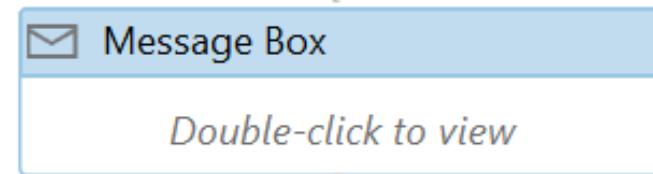
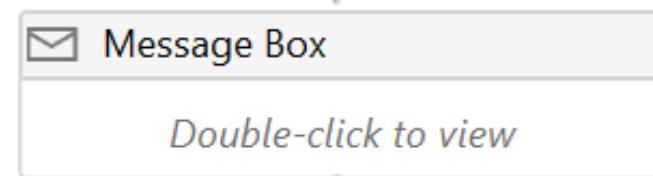
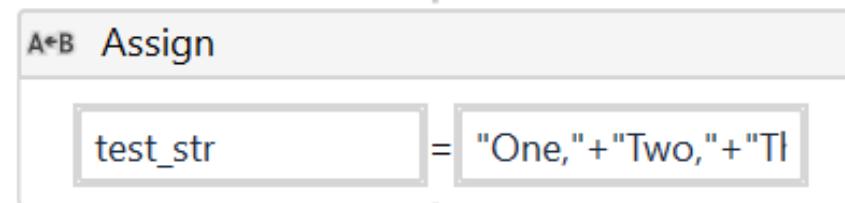
1. Use 'Assign' to set test_str to
“One,Two,Three” + vbCr + “Four,Five”
(‘+’ is used to join the two strings)
2. Use a Message Box to inspect the contents of test_str.





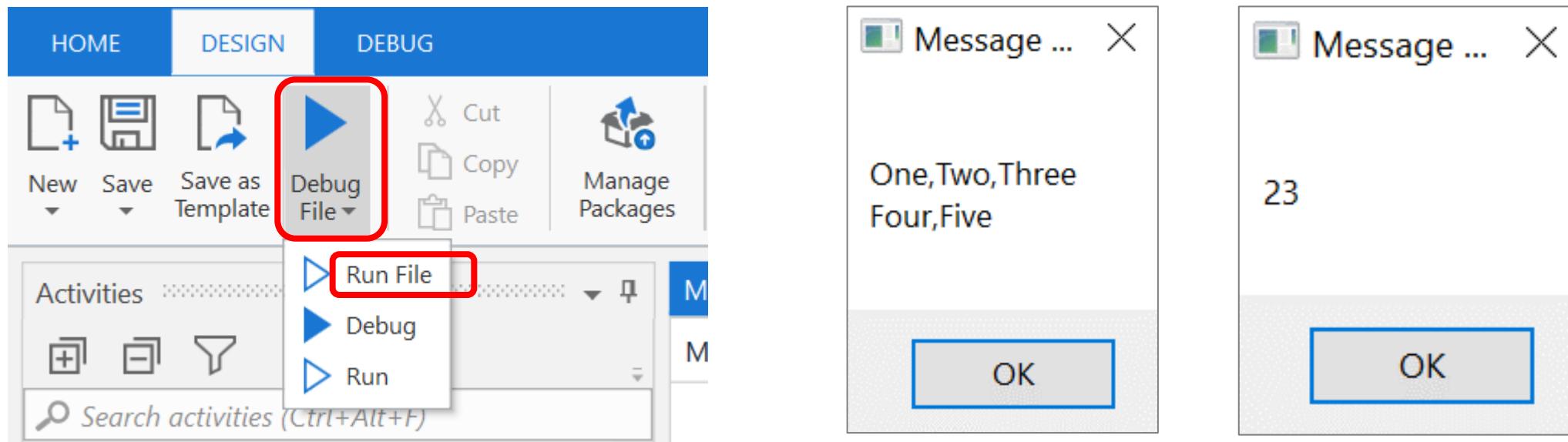
Exercise 10a (Step 4)

1. Add another Message Box to check the length of test_str



Exercise 10a (Step 5)

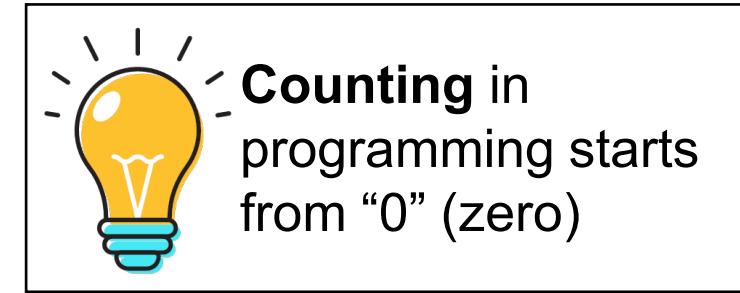
Click the **Debug File** button and select “Run File” to run your script.
Inspect the contents of the message boxes.



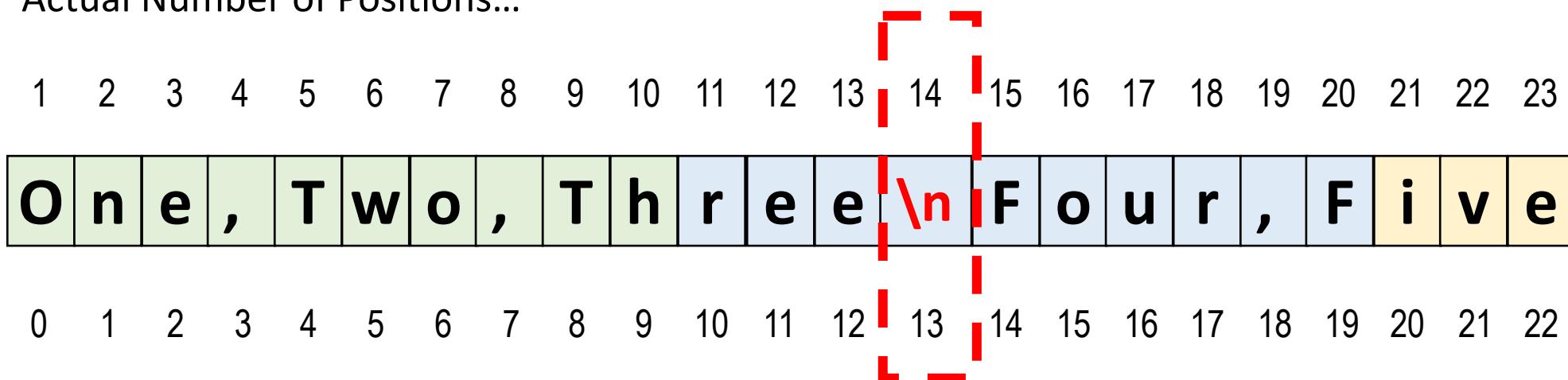
Exercise 10a

Why is the length of the string ‘23’?

New line (vbCr) is at position 13 and takes up one spot within the string.



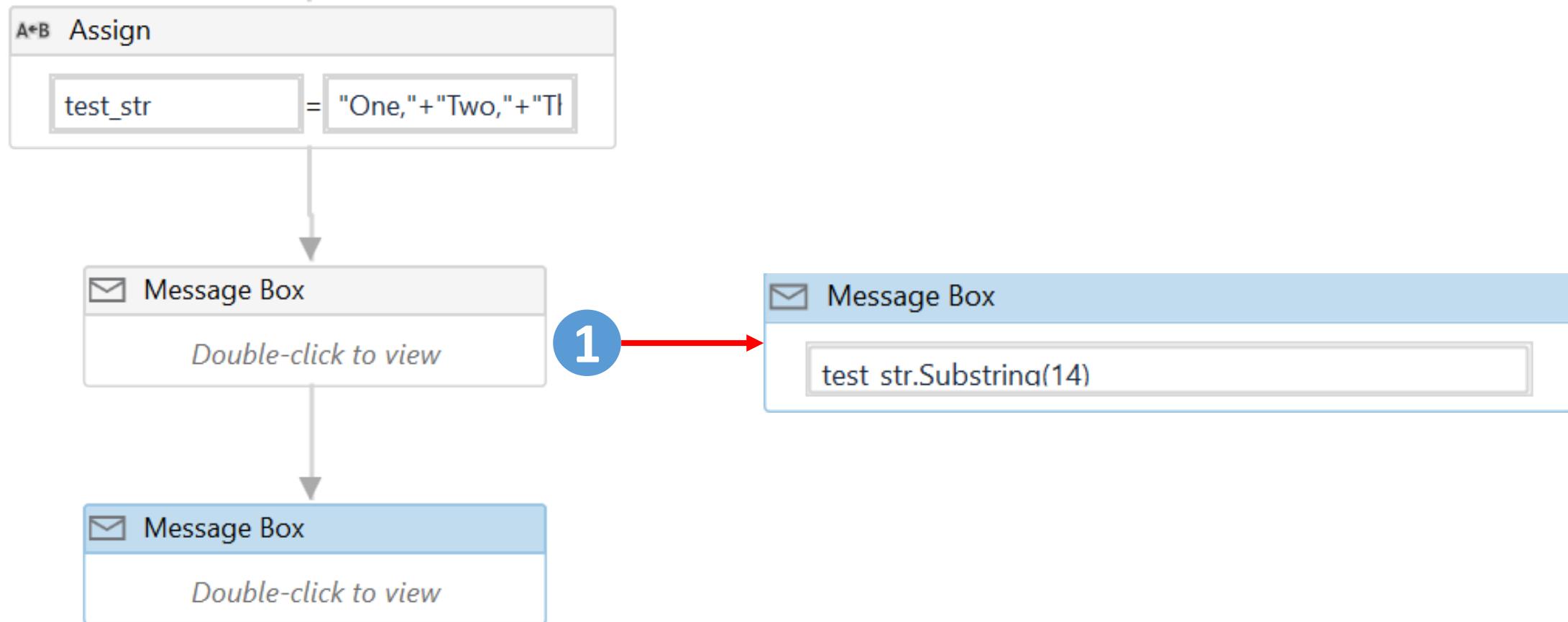
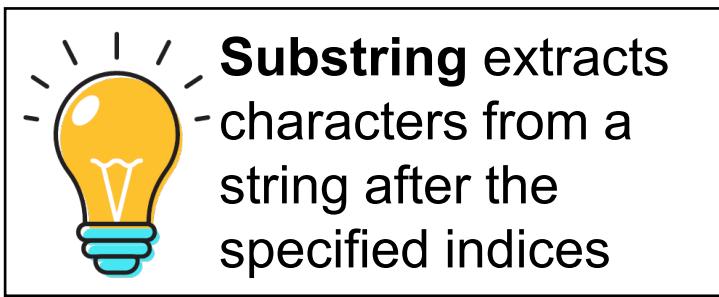
Actual Number of Positions...



Counting in programming...

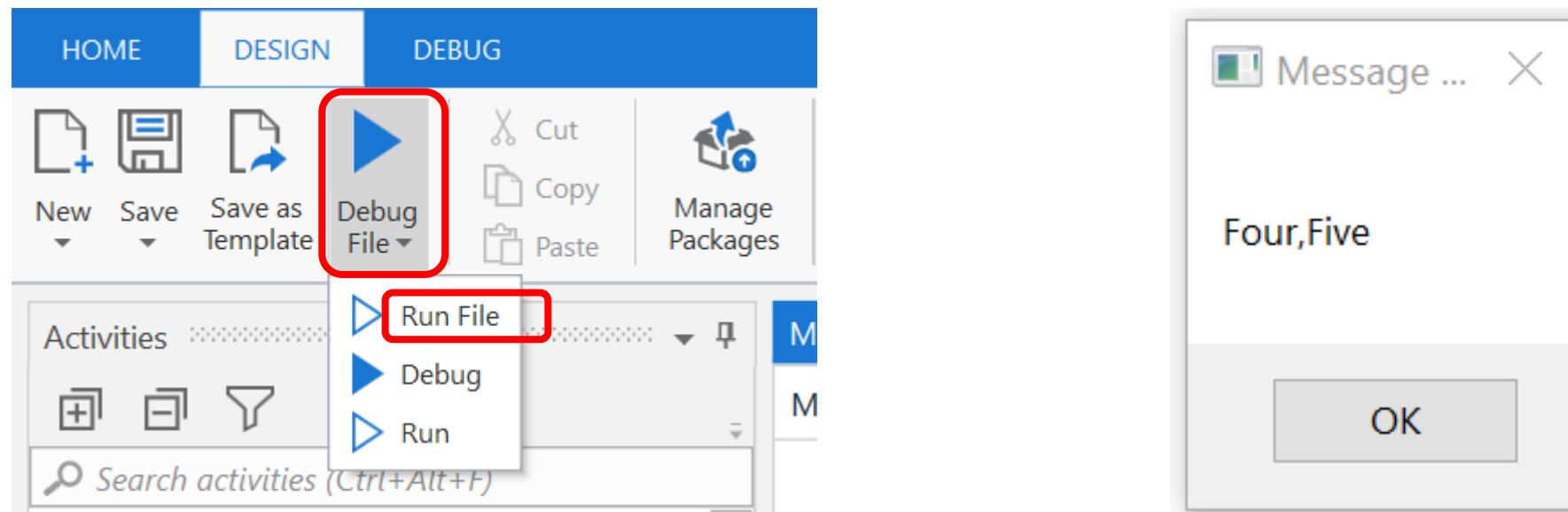
Exercise 10b (Step 1)

1. Use **Substring** to work with parts of a string directly



Exercise 10b (Step 2)

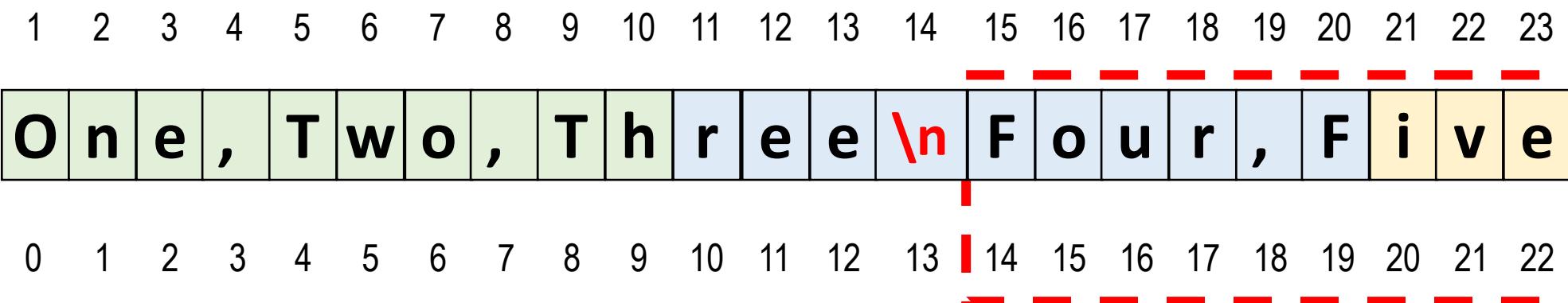
Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message boxes.



Exercise 10b

Substring(14) asks UiPath to identify the content of test_str, starting from position 14. Hence, the output '**Four,Five**'.

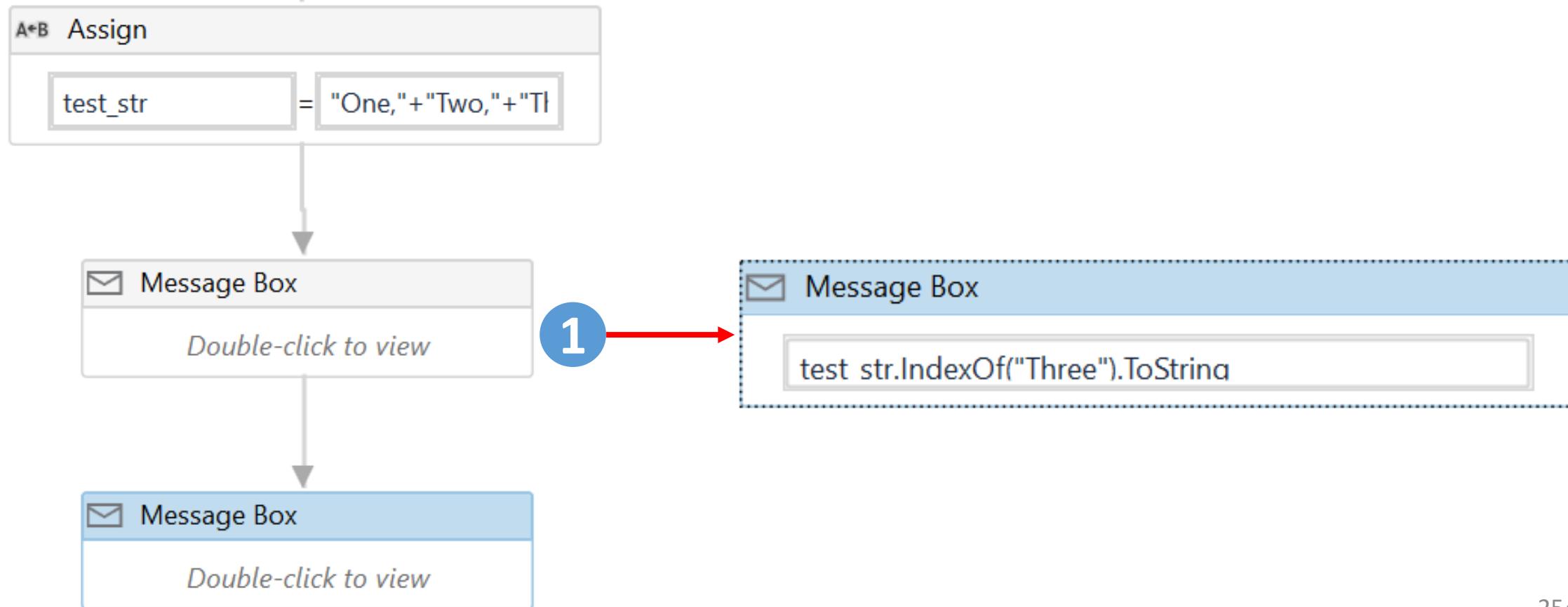
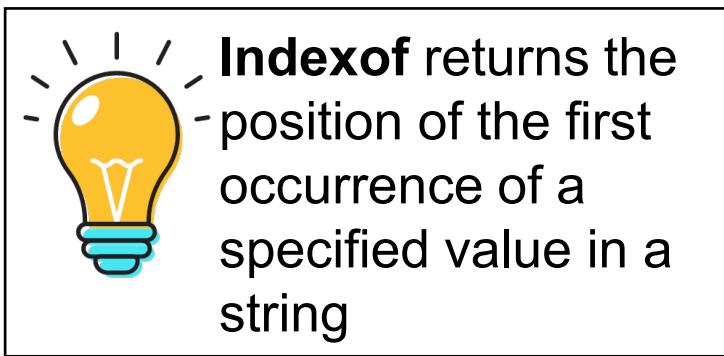
Actual Number of Positions...



Counting in programming...

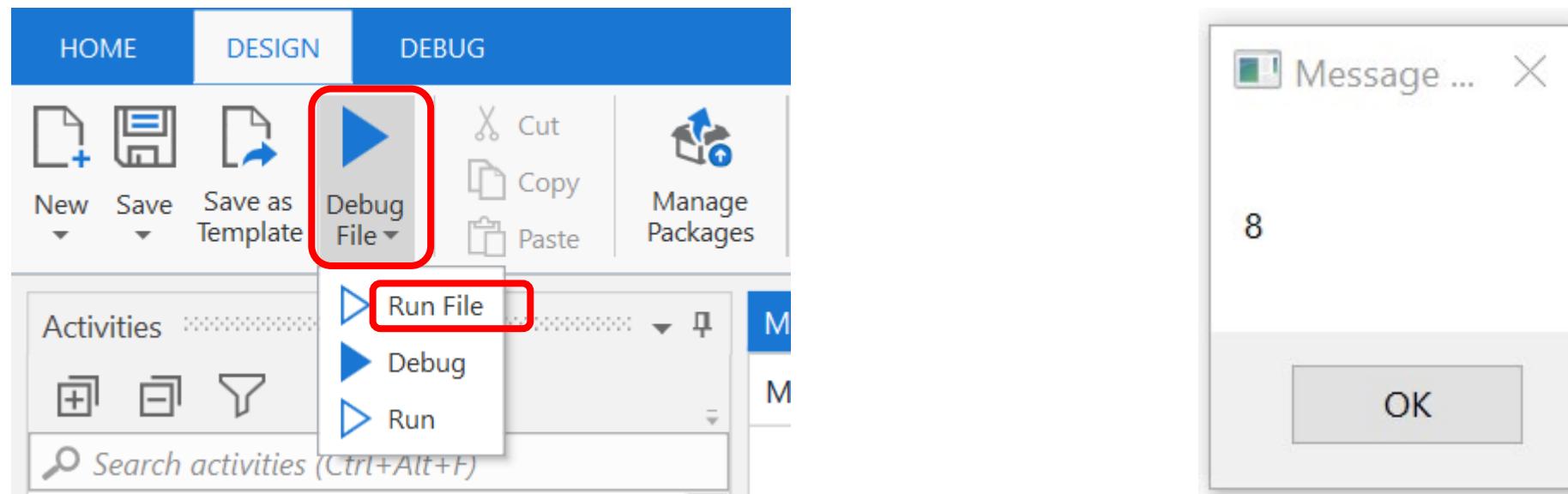
Exercise 10c (Step 1)

1. Use **Indexof** to identify the position of a substring within a source



Exercise 10c (Step 2)

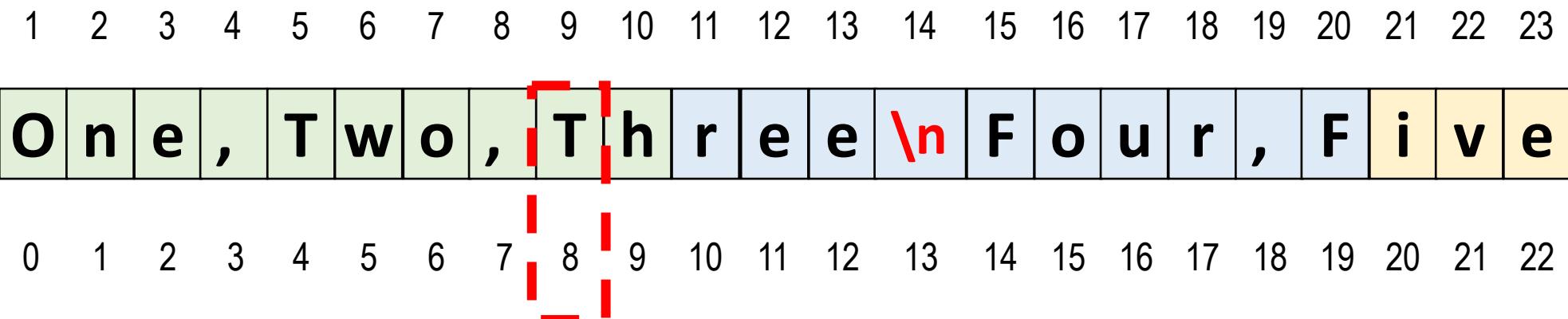
Click the **Debug File** button and select “Run File” to run your script.
Inspect the contents of the message boxes.



Exercise 10c

Indexof("Three") asks UiPath to identify the starting position of "Three". Hence, the output of 8.

Actual Number of Positions...



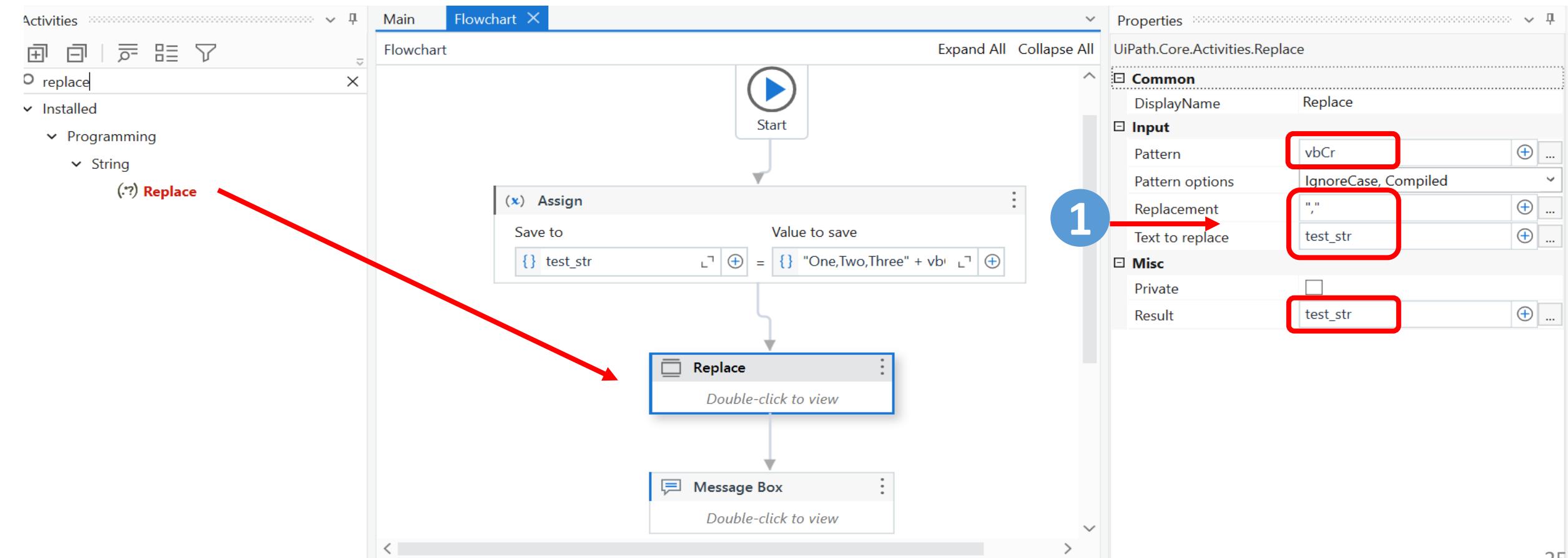
Counting in programming...

Exercise 10d (Step 1)

1. Use **Replace** to substitute text within a string with other text.

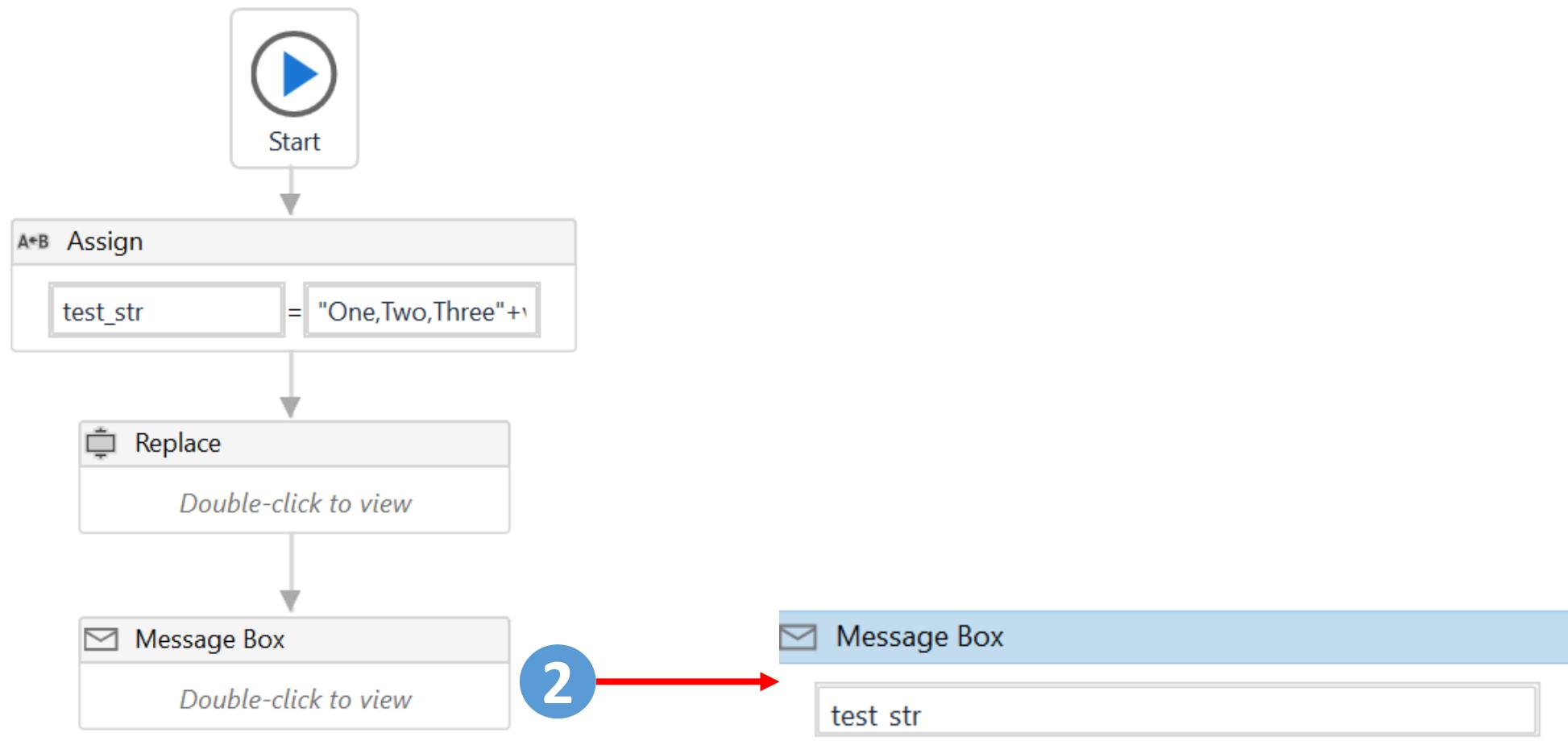
Here, we will replace **vbCr** (new line) with "," (comma) within **test_str**.

Initially, **test_str** = "One,Two,Three" + vbCr + "Four,Five"



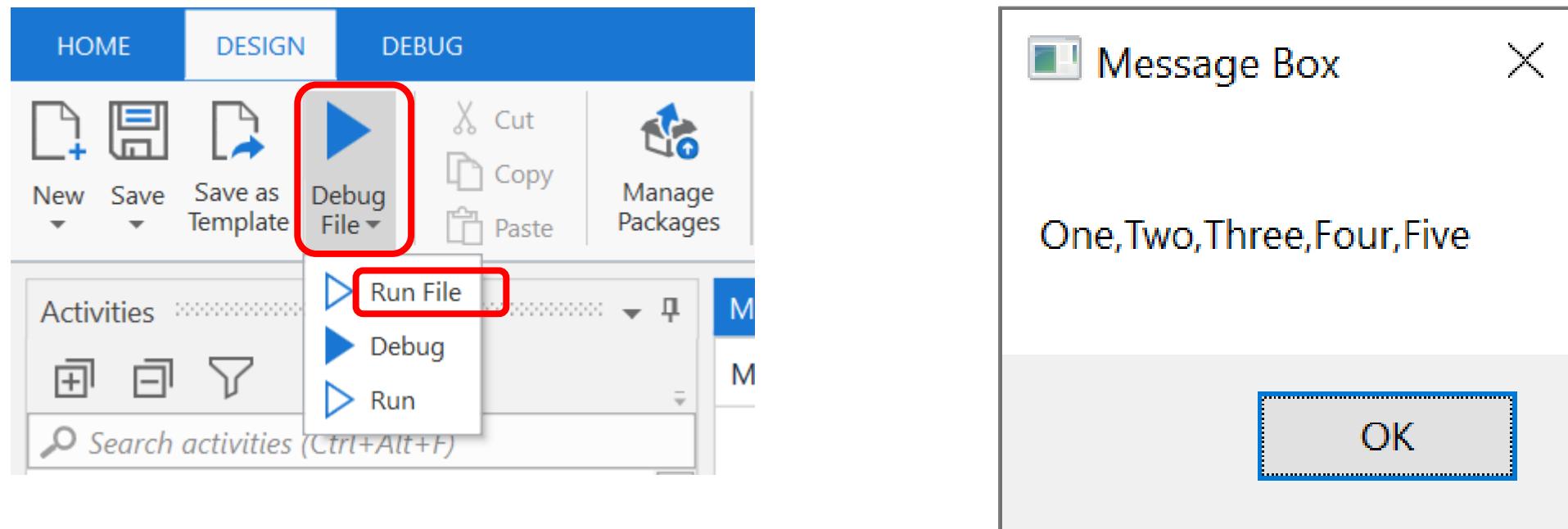
Exercise 10d (Step 2)

2. Edit the Message Box to display test_str



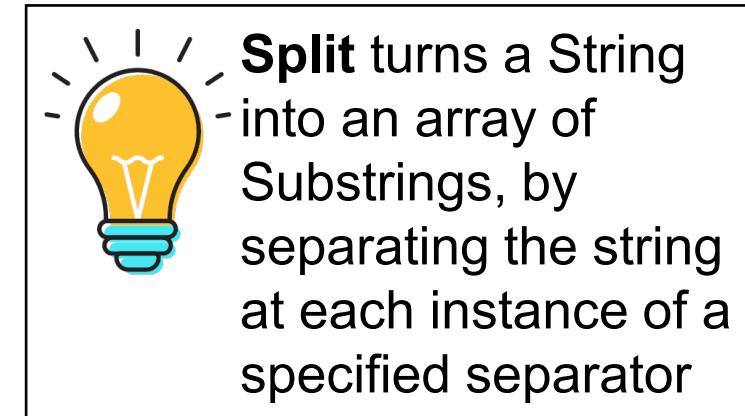
Exercise 10d (Step 3)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message boxes.



Exercise 10e (Step 1)

Use **Split** to split a string into an array of substrings

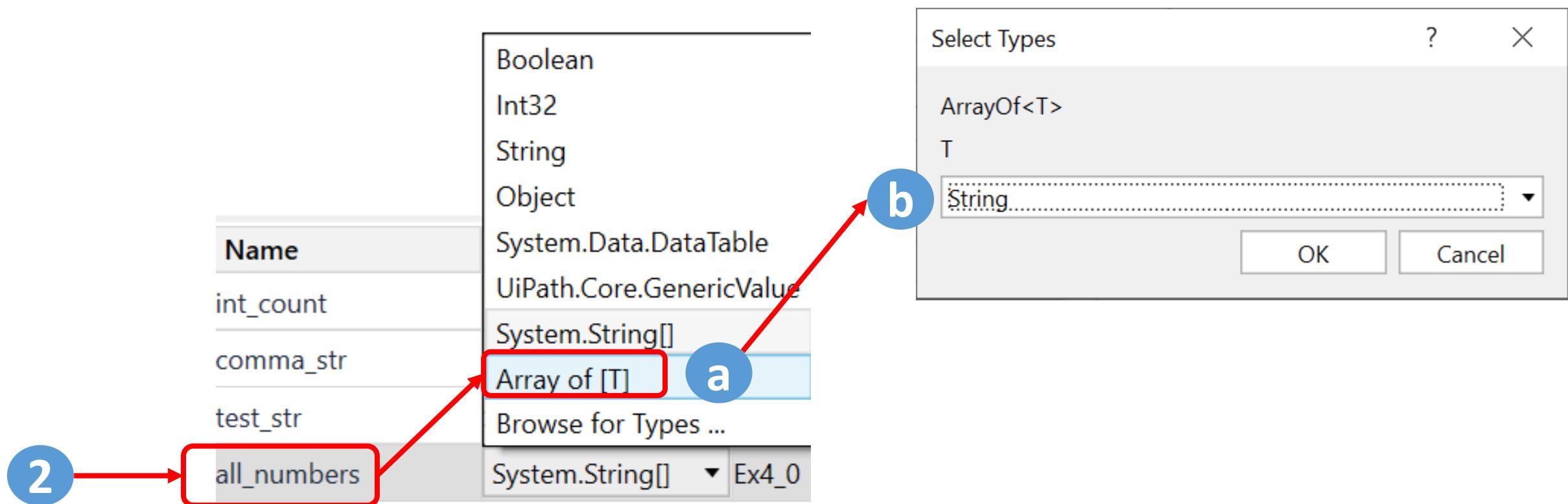


1. Create a new variable `comma_str` and initialize it as `,`

Name	Variable type	Scope	Default
comma_str	String	Ex4_0	<code>","</code>

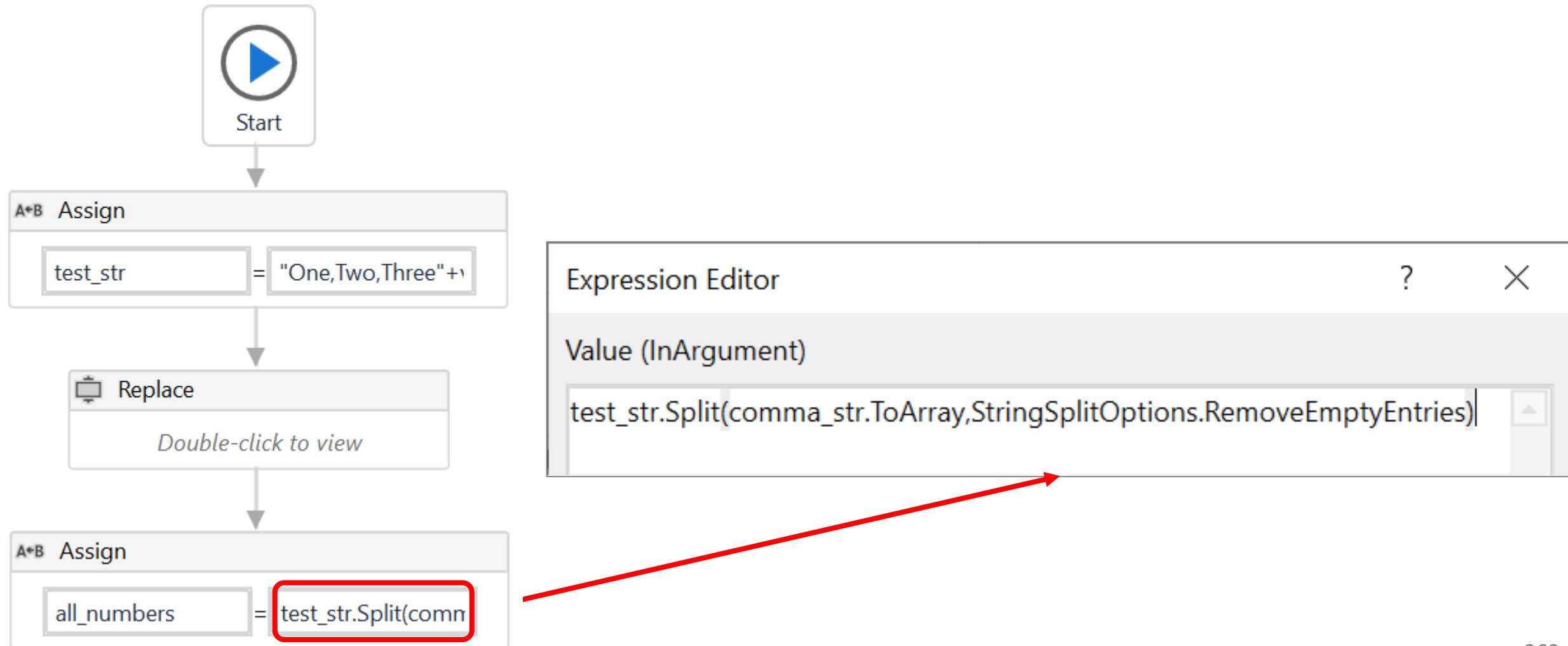
Exercise 10e (Step 2)

2. Create a new variable with name “all_numbers”.
 - a) Click the drop-down and select Array of [T]
 - b) Choose “String” then click ok



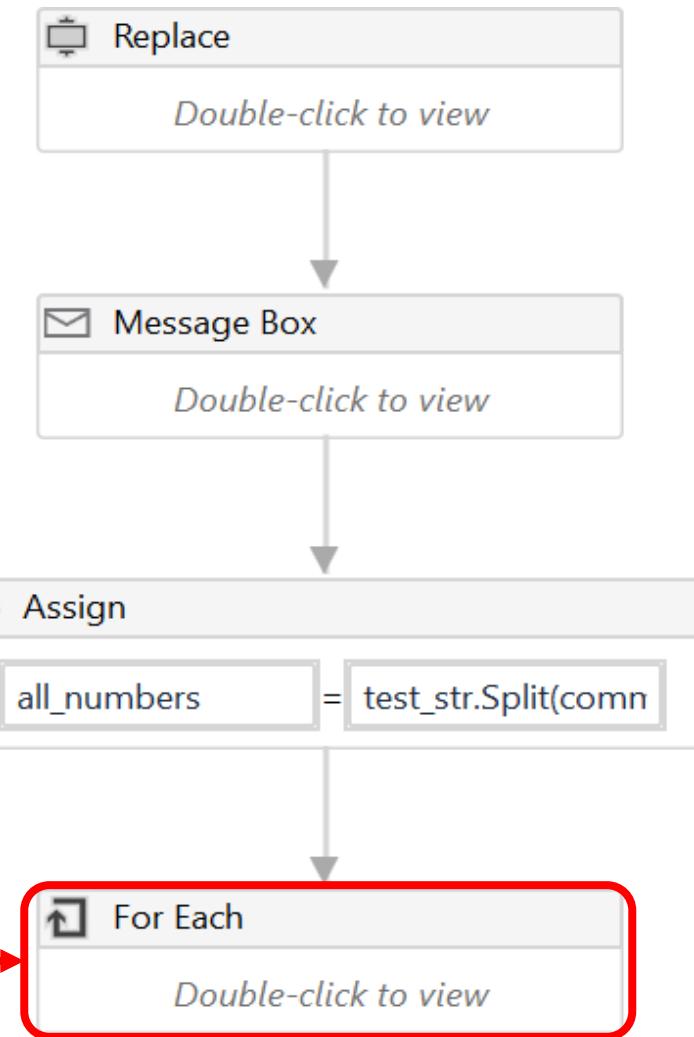
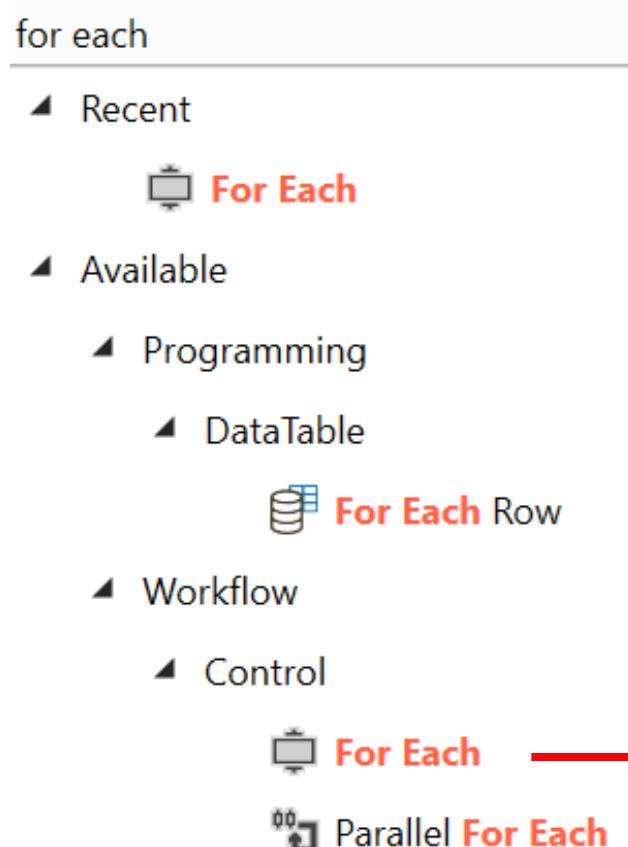
Exercise 10e (Step 3)

Insert an activity “Assign”



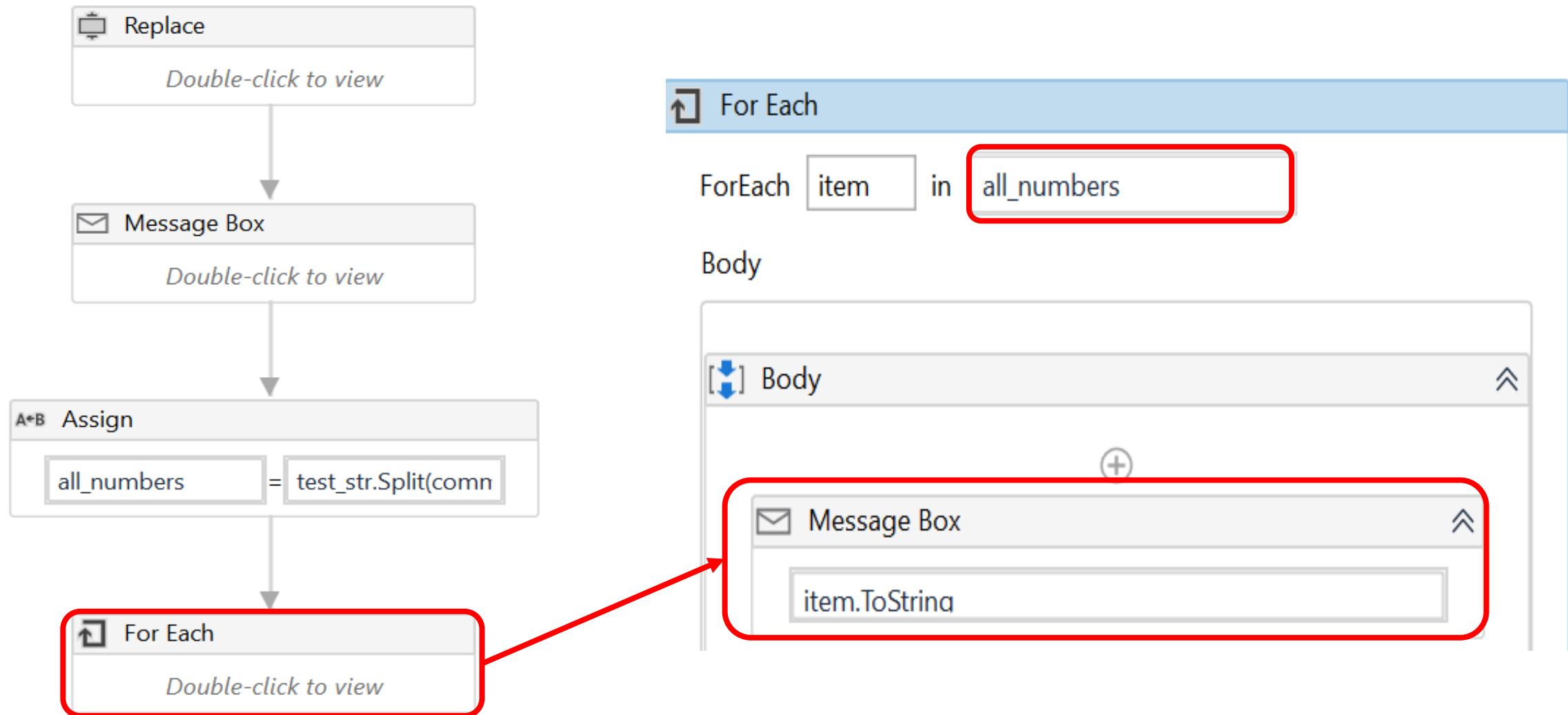
Exercise 10e (Step 4)

Insert an activity “For Each”



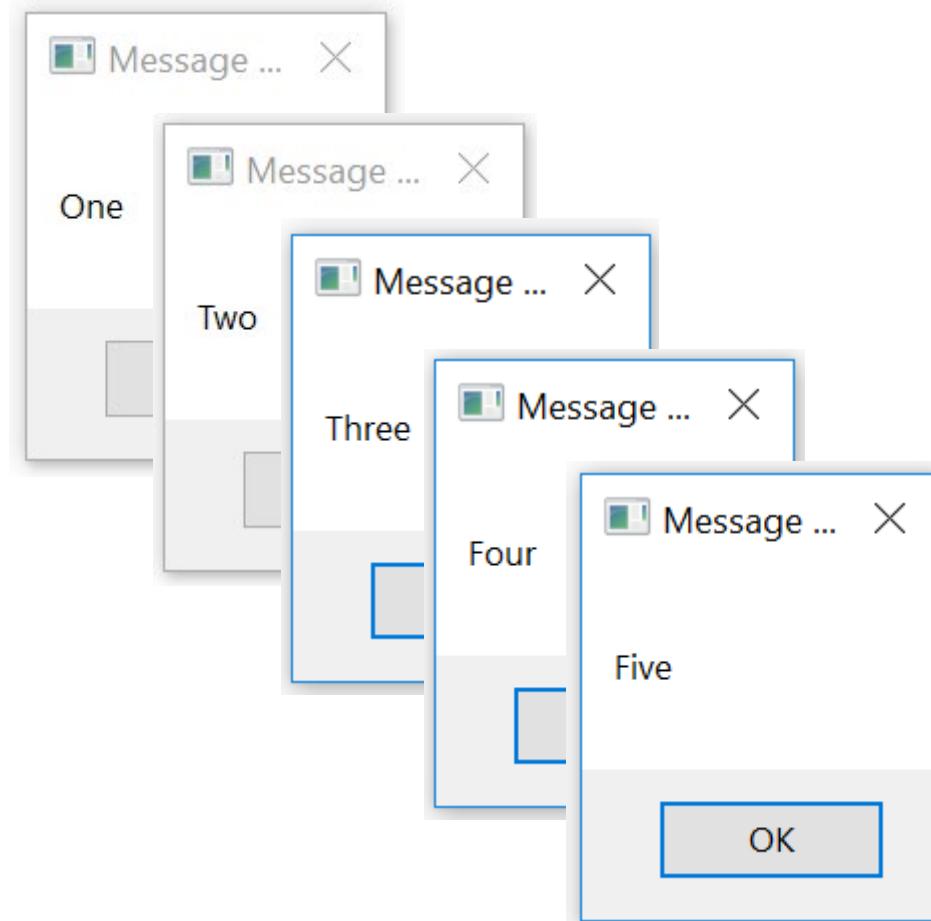
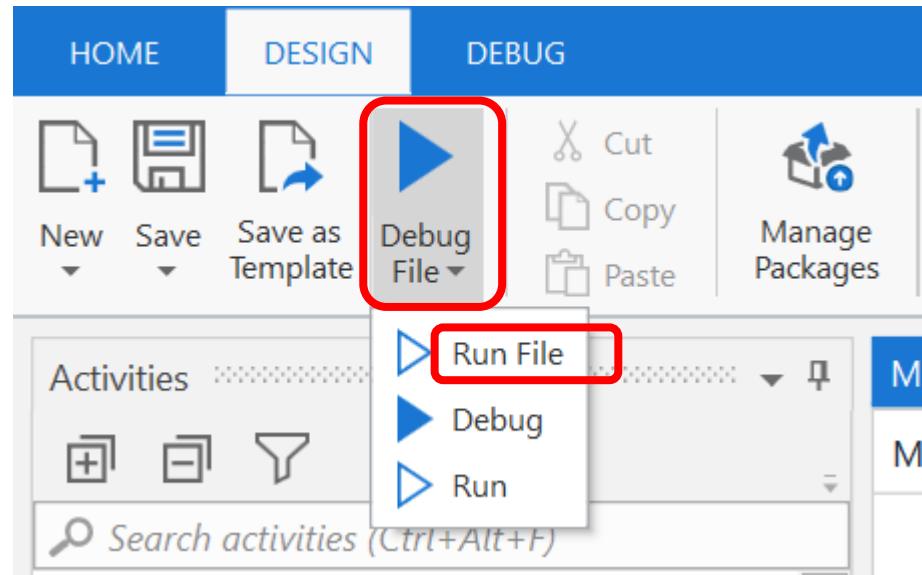
Exercise 10e (Step 5)

Insert a “Message Box” within the “For Each” activity.



Exercise 10e (Step 6)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message boxes.



Exercise 10e

You should see a series of similar message boxes.

“Split” has extracted sub-strings from test_str using comma as the separator.

“Split” then placed the output into all_numbers as an array of strings.

	0	1	2	3	4
all_numbers(0) ->	O	n	e		
all_numbers(1) ->	T	w	o		
all_numbers(2) ->	T	h	r	e	e
all_numbers(3) ->	F	o	u	r	
all_numbers(4) ->	F	i	v	e	



Ex10.1 – Ex 10.5 PDF – extracting info

1. Create a new Process and name it “*Robot6_XXXXXX*” with description as “*RPA Lesson8 Ex10.1 to Ex10.5*
XXXXXXX is your first name, e.g. Jun_Hao or Jeremy etc
2. Script Robot to read PDF files.
3. Refer to Ex10.1 to 10.5 in “RPA UiPath Hands-On Guide” for step-by-step instruction



	A	B	C	D	E	F	G
1	Seller	Invoice_Ref	Invoice_Date	Amt_Due	Due_Date	Bill_To	Invoice_Purpose
2	Acme Inc	A100	7/3/2020	\$200.00	21/3/2020	John Tan	Catered Food
3							

Exercise 10.1

Read PDF files



Exercise 10.1

This exercise shows how RPA can be used to read the content of an Adobe Acrobat file as text.

Acme Invoice.pdf (in the training material) is provided for your use.

You may also wish to create your own invoice, from the template at <https://templates.office.com/en-us/Invoices>. Fill in the Excel file, then print to PDF

Exercise 10.1

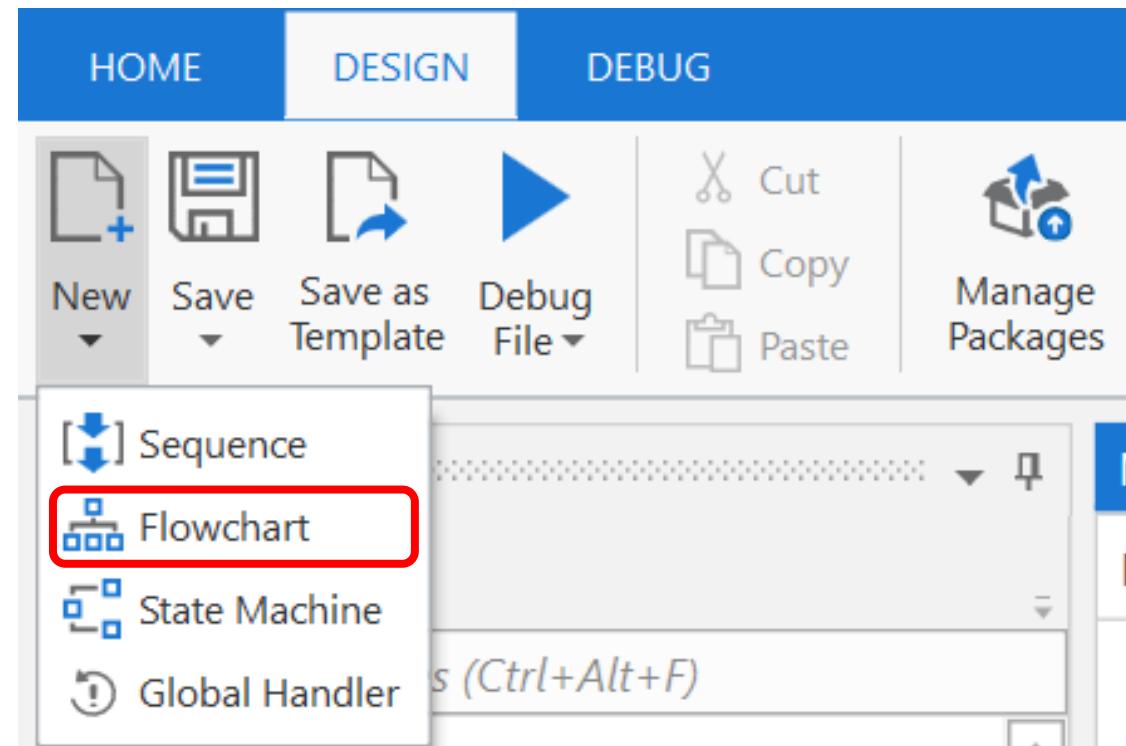
- We will be working with a sample PDF file named “Acme Invoice”.
- We wish to extract the following fields:
 - Seller
 - Invoice Date
 - Invoice Reference Number
 - Invoice Purpose
 - Amount Due
 - Amount Due Date
 - Bill To



INVOICE											
DATE:	7/3/2020										
INVOICE #	A100										
FOR:	Catered Food										
Acme Inc											
12 Acme Road Singapore 123456 Phone +65 6789 1234 Fax +65 68901 2345											
Bill To: John Tan Me Co 18 Dover Street Singapore 234561											
<table border="1"> <thead> <tr> <th>DESCRIPTION</th> <th>AMOUNT</th> </tr> </thead> <tbody> <tr> <td>Curry Puffs x10 boxes</td> <td>\$ 150.00</td> </tr> <tr> <td>Chicken Wings x50</td> <td>\$ 50.00</td> </tr> <tr> <td>TOTAL</td> <td>\$ 200.00</td> </tr> <tr> <td>DUE ON</td> <td>21/3/2020</td> </tr> </tbody> </table>		DESCRIPTION	AMOUNT	Curry Puffs x10 boxes	\$ 150.00	Chicken Wings x50	\$ 50.00	TOTAL	\$ 200.00	DUE ON	21/3/2020
DESCRIPTION	AMOUNT										
Curry Puffs x10 boxes	\$ 150.00										
Chicken Wings x50	\$ 50.00										
TOTAL	\$ 200.00										
DUE ON	21/3/2020										
Make all checks payable to Acme Inc If you have any questions concerning this invoice, contact George Koh at +65 6789 1234											
THANK YOU FOR YOUR BUSINESS!											

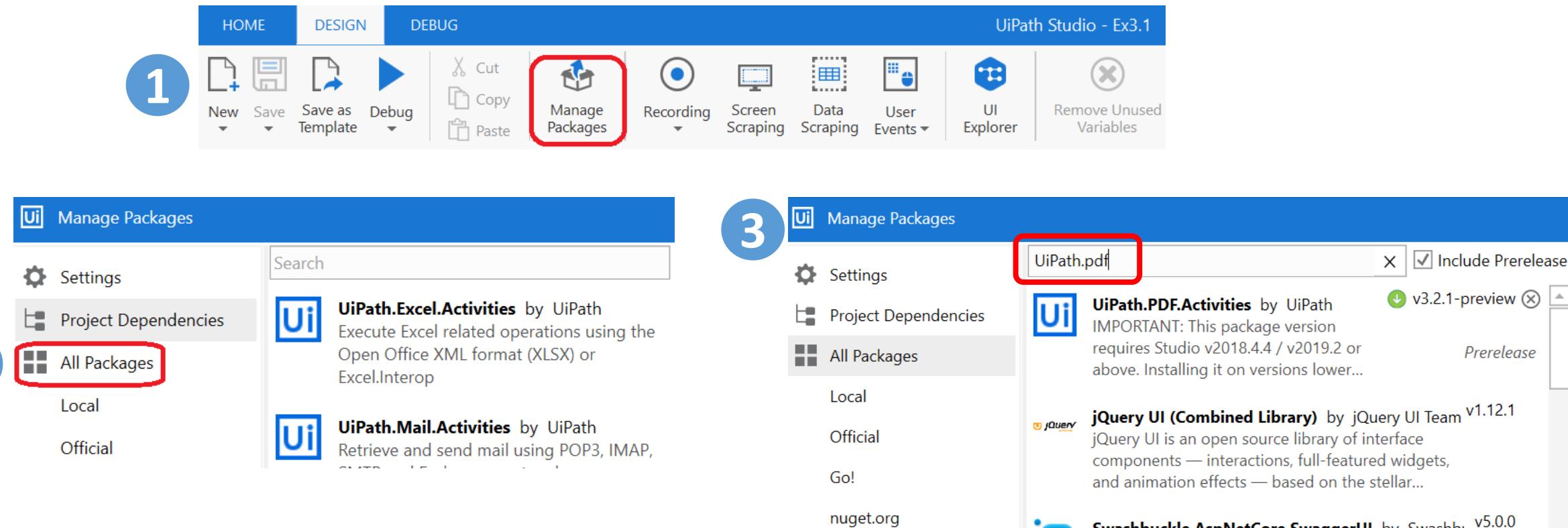
Exercise 10.1 (Step 1)

In UiPath Studio, create a new flowchart, and name it “Ex10.1”.



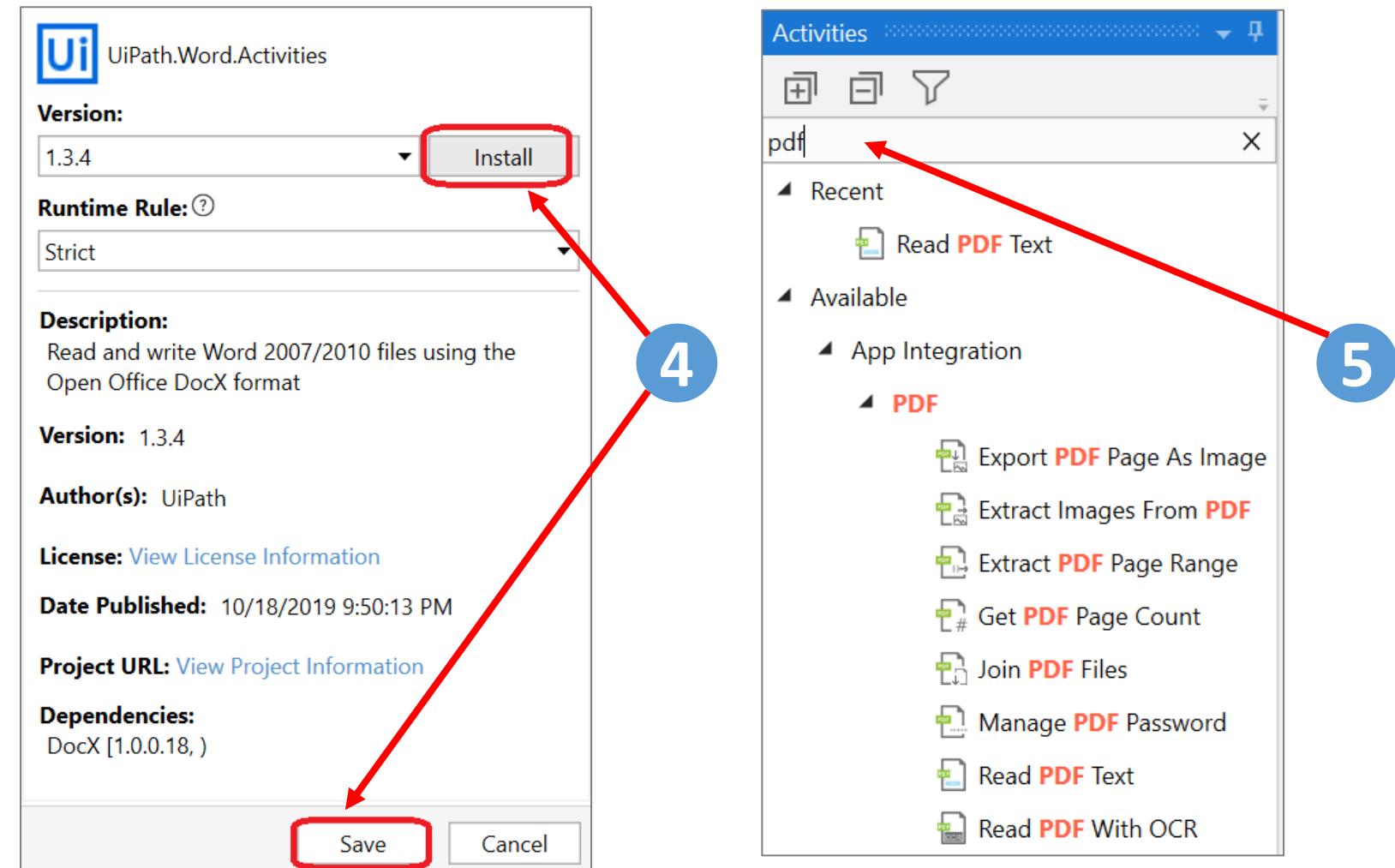
Exercise 10.1 (Step 2)

1. Click on “Manage Packages” in Design Toolbar
2. Next, click “All Packages”
3. In the search bar, type “UiPath.PDF” and select the package



Exercise 10.1 (Step 2)

4. Click on “Install”, then “Save” at bottom right. Accept the license agreement
5. To check that the package is installed, type “pdf” into Activities search bar to view available functions



Exercise 10.1 (Step 3)

1. Insert “Read PDF text”. Make it the start node.
2. Set **FileName** of the invoice and set output as a new string variable string “**All_Text**”



+ - F

read pdf

- ◀ Recent
 - Read PDF Text
- ◀ Available
 - ◀ App Integration
- ◀ PDF
 - 1 Read PDF Text
 - Read PDF With OCR

+ - F

Start

Read PDF Text

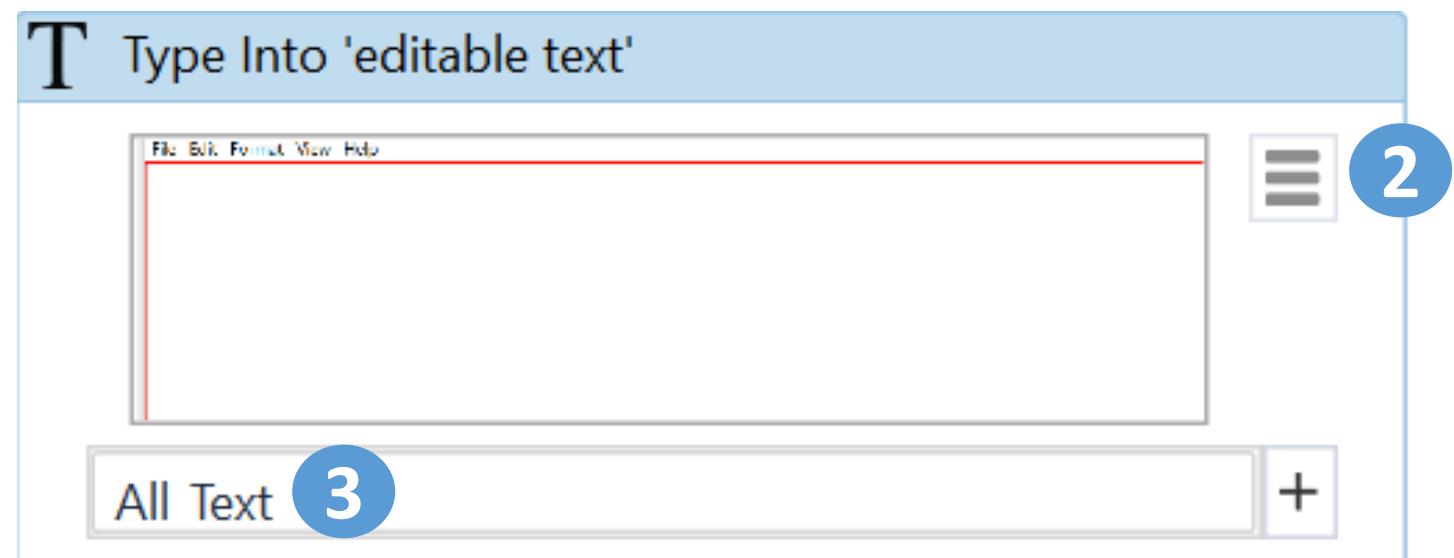
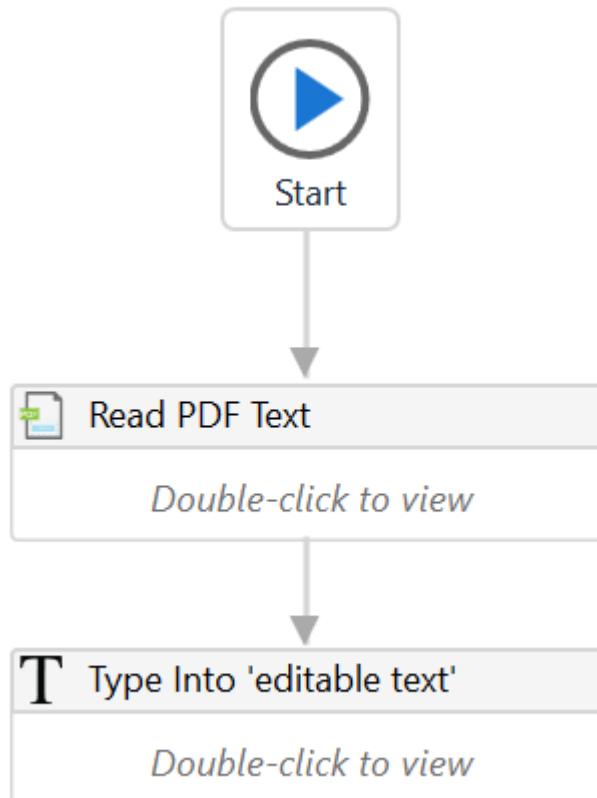
Double-click to view

Properties

UiPath.PDF.Activities.ReadPDFText	
Common	
DisplayName	Read PDF Text
File	
FileName	"D:\RPA\RPA_Intermediate\Tra...
Password	The password of the PDF file, if...
Input	
PreserveFormatti...	A flag used to signal to preserv...
Range	"All"
Misc	
Private	<input type="checkbox"/>
Output	
Text	All_Text

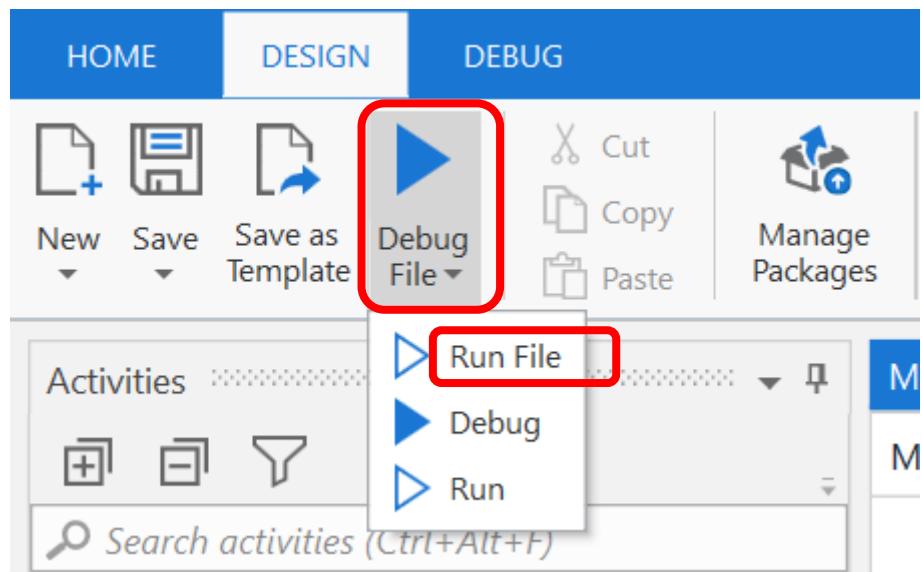
Exercise 10.1 (Step 4)

1. Open Notepad
2. Insert a “Type Into” activity and indicate on screen to select notepad
3. Specify “**All_Text**” as the input



Exercise 10.1 (Step 5)

Click the **Debug File** button and select “Run File” to run your script.
Inspect the content of the notepad.



Untitled - Notepad
File Edit Format View Help
Acme Inc INVOICE

12 Acme Road DATE: 7/3/2020
Singapore 123456 INVOICE #A100
Phone +65 6789 1234 Fax +65 68901 2345 FOR: Catered Food

Bill To:
John Tan
Me Co
18 Dover Street
Singapore 234561

DESCRIPTION	AMOUNT
Curry Puffs x10 boxes \$	150.00
Chicken Wings x50 \$	50.00
TOTAL \$	200.00

DUE ON 21/3/2020

Make all checks payable to Acme Inc
If you have any questions concerning this invoice, contact George Koh at +65 6789 1234

THANK YOU FOR YOUR BUSINESS!

Exercise 10.1 (Review)

Notice how the contents of the invoice is extracted without formatting.
How are we going to read these fields?

- Seller
- Invoice Date
- Invoice Reference Number
- Invoice Purpose
- Amount Due
- Amount Due Date
- Bill To

Exercise 10.2

String manipulation (I)

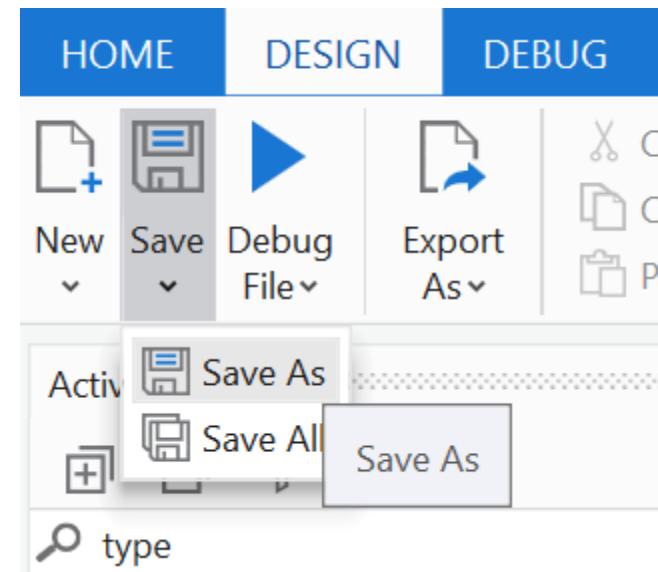
Exercise 10.2

This exercise shows how to use RPA to read the content of the sample PDF file “Acme Invoice” as a long string and *split the string into individual lines.*

INVOICE											
Acme Inc											
12 Acme Road Singapore 123456 Phone +65 6789 1234 Fax +65 68901 2345	DATE: 7/3/2020 INVOICE #: A100 FOR: Catered Food										
Bill To: John Tan Me Co 18 Dover Street Singapore 234561											
<table border="1"> <thead> <tr> <th>DESCRIPTION</th> <th>AMOUNT</th> </tr> </thead> <tbody> <tr> <td>Curry Puffs x10 boxes</td> <td>\$ 150.00</td> </tr> <tr> <td>Chicken Wings x50</td> <td>\$ 50.00</td> </tr> <tr> <td>TOTAL</td> <td>\$ 200.00</td> </tr> <tr> <td>DUE ON</td> <td>21/3/2020</td> </tr> </tbody> </table>		DESCRIPTION	AMOUNT	Curry Puffs x10 boxes	\$ 150.00	Chicken Wings x50	\$ 50.00	TOTAL	\$ 200.00	DUE ON	21/3/2020
DESCRIPTION	AMOUNT										
Curry Puffs x10 boxes	\$ 150.00										
Chicken Wings x50	\$ 50.00										
TOTAL	\$ 200.00										
DUE ON	21/3/2020										
Make all checks payable to Acme Inc If you have any questions concerning this invoice, contact George Koh at +65 6789 1234											
THANK YOU FOR YOUR BUSINESS!											

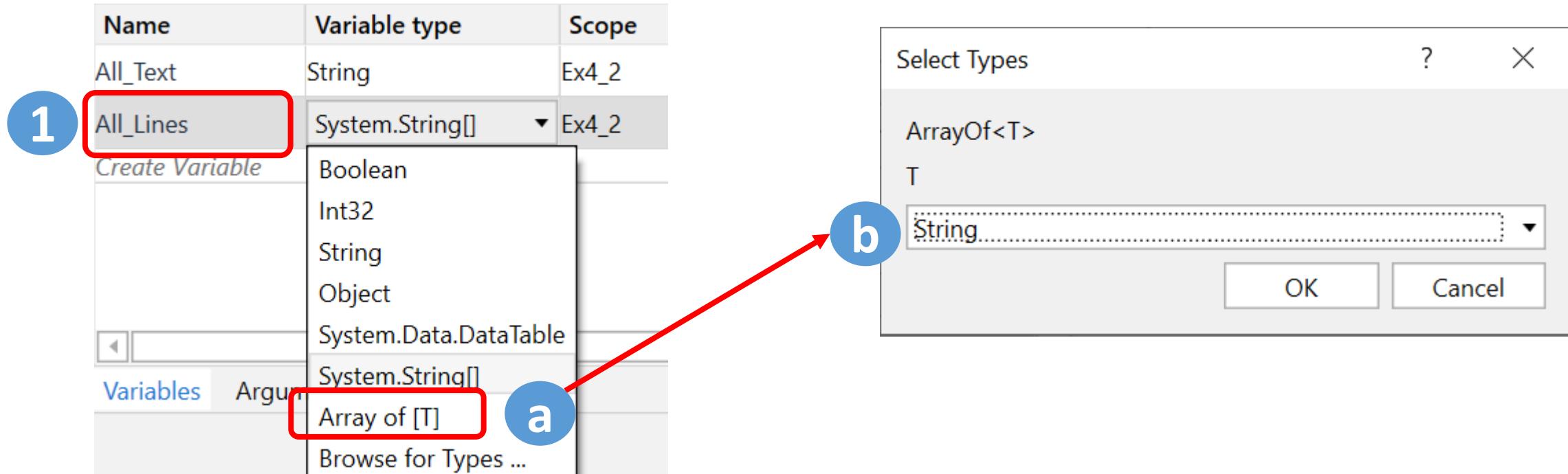
Exercise 10.2 (Step 1)

In UiPath Studio, save “Ex10.1” as “Ex10.2” .



Exercise 10.2 (Step 2)

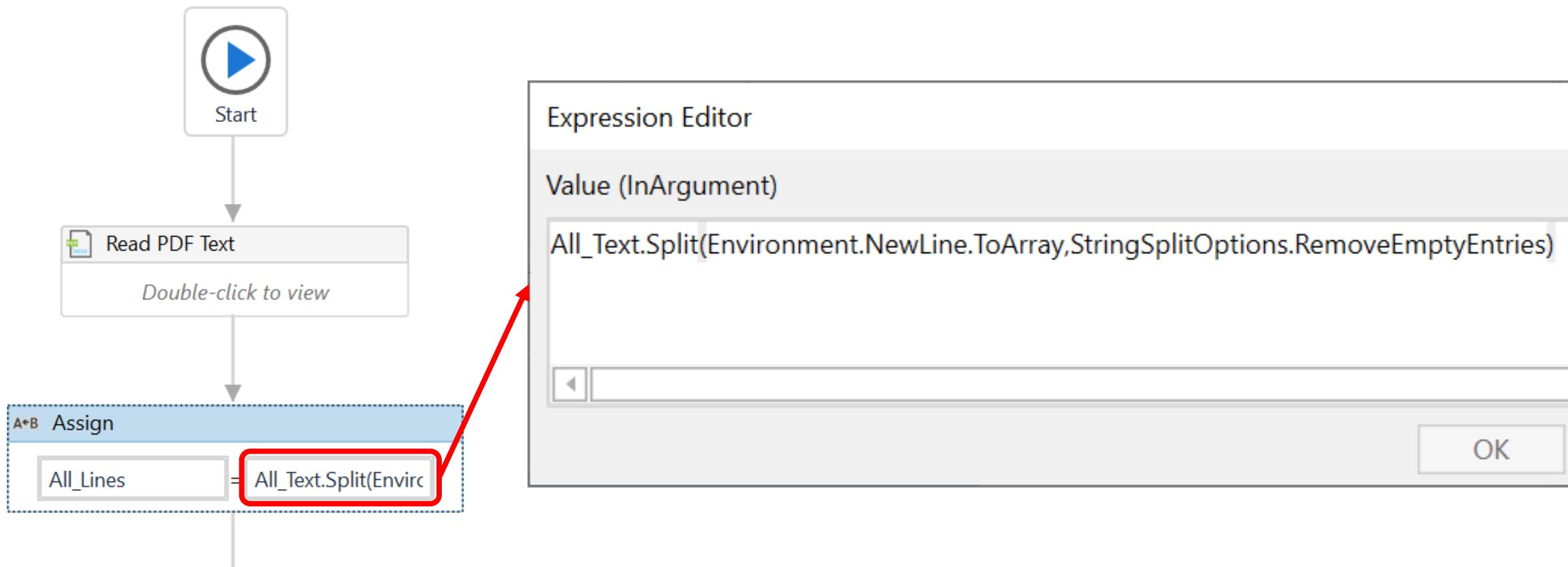
1. Create a new variable with name “**All_Lines**”.
 - a) Click the drop-down and select Array of [T]
 - b) Choose “String” then click ok



Exercise 10.2 (Step 3)

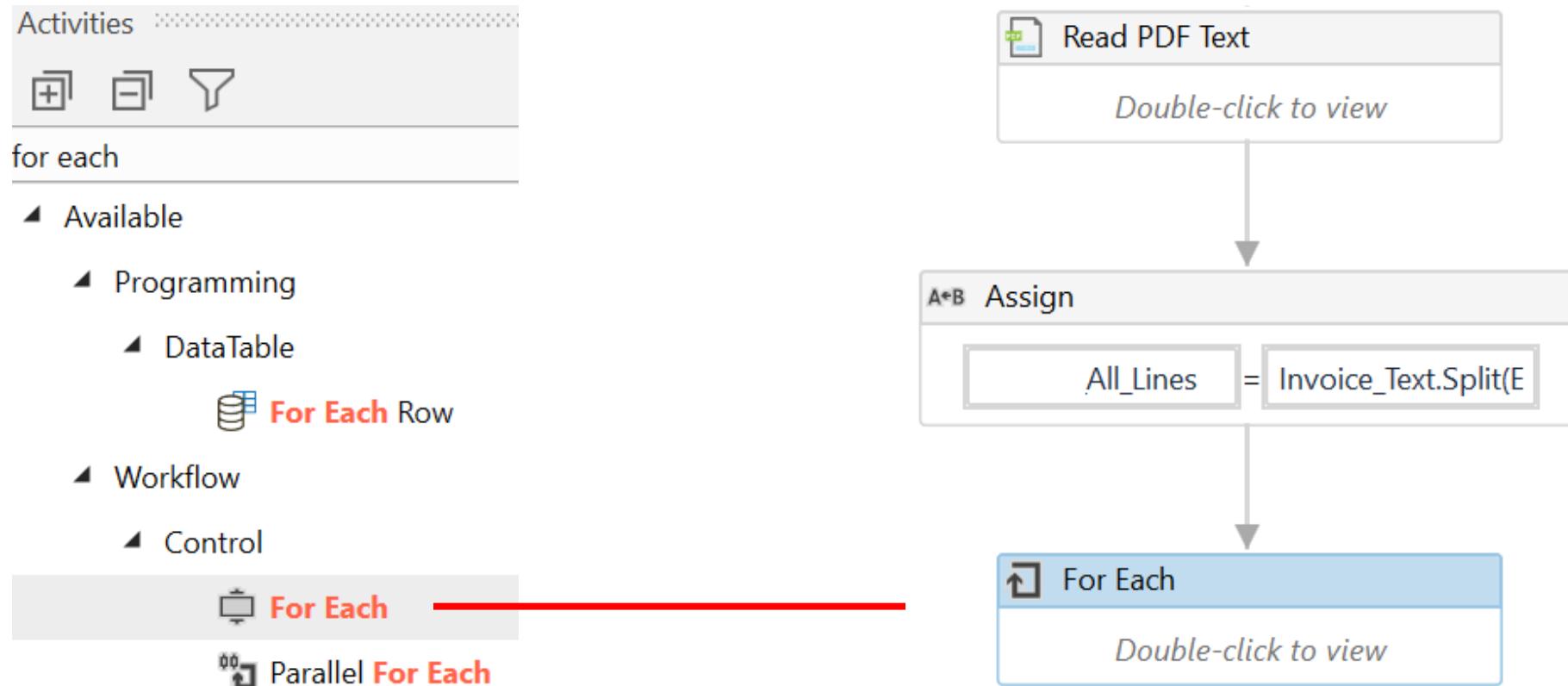
Add an “Assign” activity to split “All_Text” into individual lines.

All_Text.Split(Environment.NewLine.ToArray, StringSplitOptions.RemoveEmptyEntries)



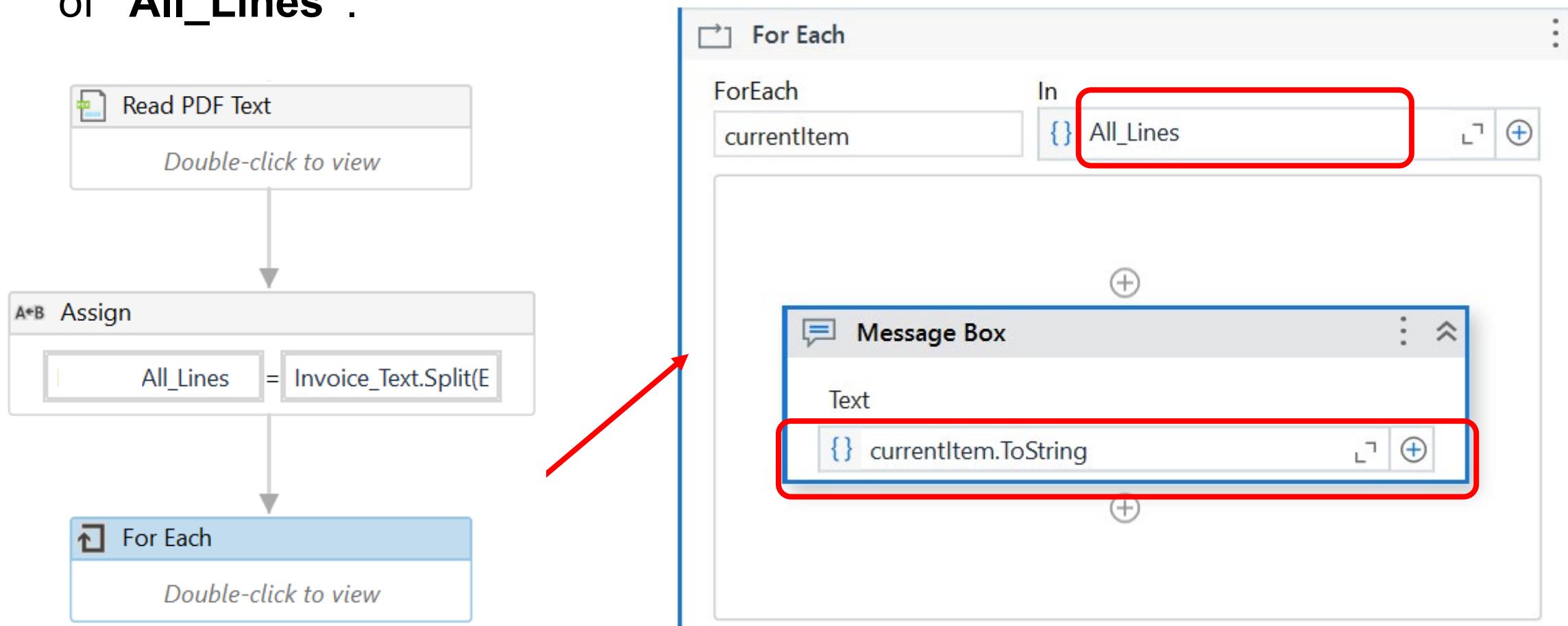
Exercise 10.2 (Step 4)

Add a “For each” activity to cycle through the contents of “All_Lines” (the individual lines created by the “Split” function).



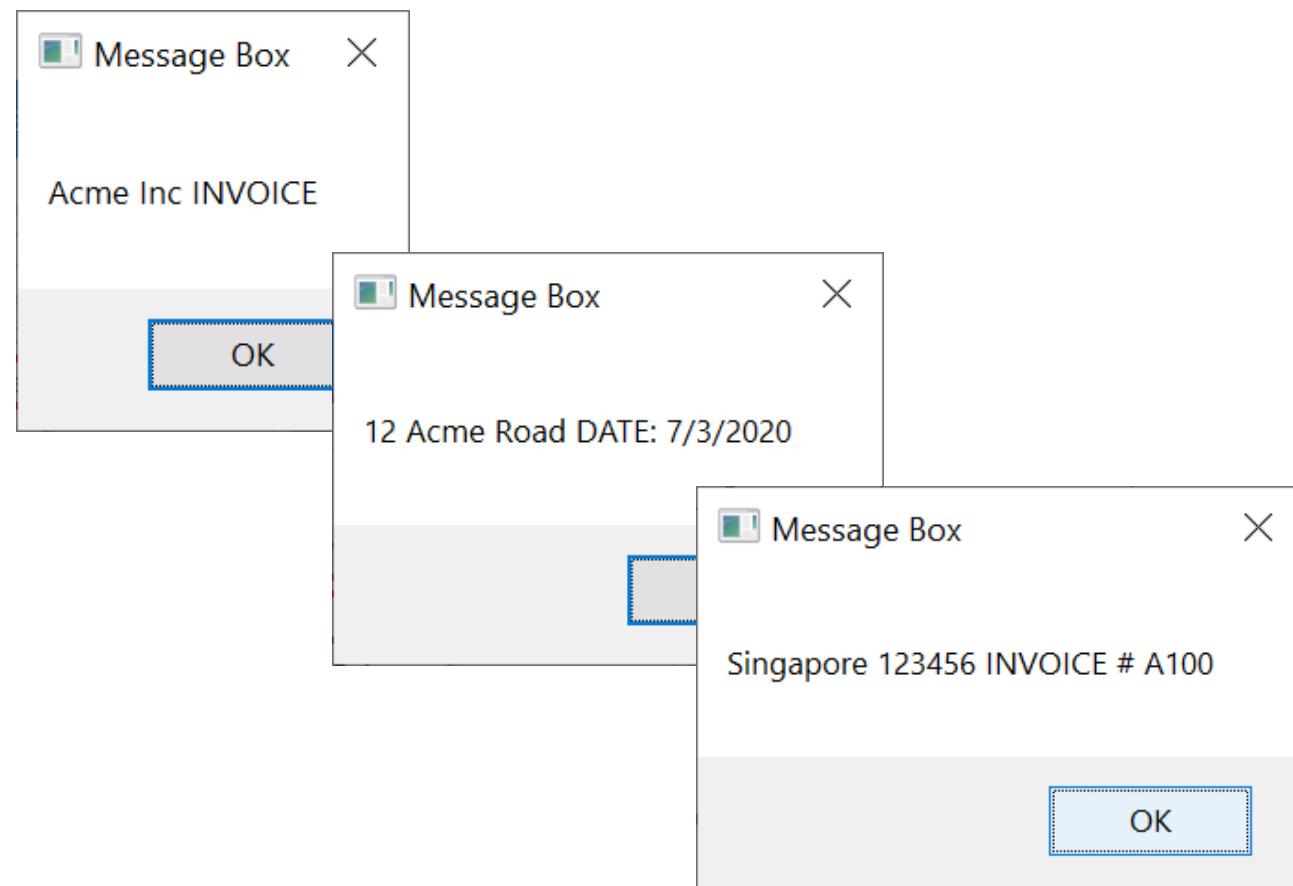
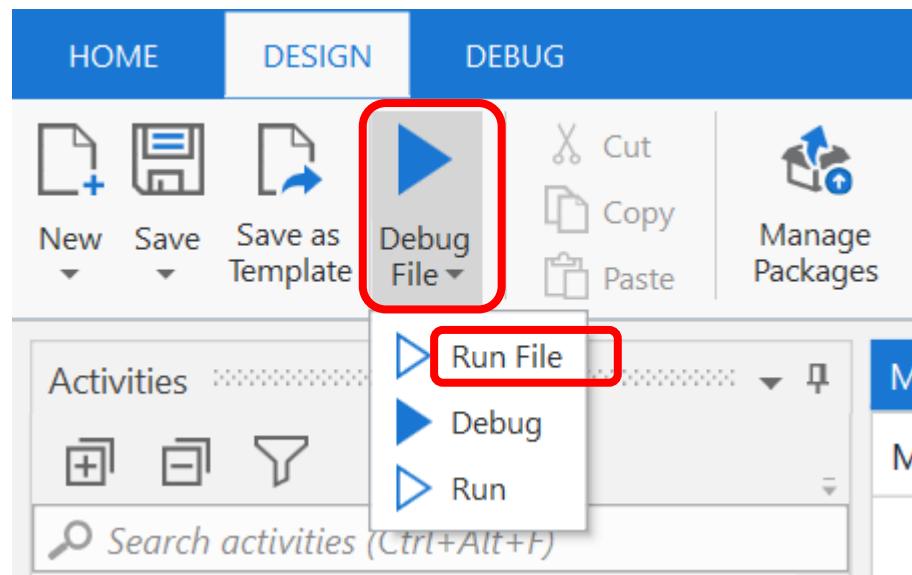
Exercise 10.2 (Step 5)

Within the “For Each” activity, add a message box to output the contents of “All_Lines”.



Exercise 10.2 (Step 6)

Click the **Debug File** button and select “Run File” to run your script.
Inspect the contents of the message boxes.



Exercise 10.2 (Review)

We have just split the invoice into individual lines.

These lines are presented as the variable, “All_Lines”, an array of strings.

All_Lines(0) ->	A c m e I n c I N V O I C E
All_Lines(1) ->	1 2 A c m e R o a d D A T E : 7 / 3 / 2 0 2 0
All_Lines(2) ->	S i n g a p o r e 1 2 3 4 5 6 I N V O I C E # A 1 0 0
:	
:	
:	
All_Lines(16) ->	T H A N K Y O U F O R Y O U R B U S I N E S S !

Exercise 10.2 (Review)

Why did we split the text into lines?

→ So that we can extract different fields by position.

E.g. Seller is found in the first line of the invoice

How do we extract the other fields?

We will look for a keyword for each field:

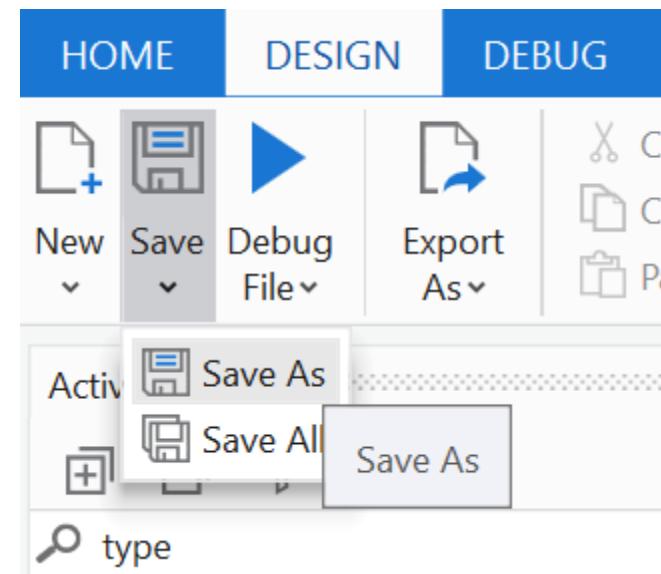
- The person in your company who ordered the product is in the line immediately below the keyword "**Bill To**"
- For Invoice Date, the keyword is "**DATE:**"
- Invoice Reference Number-- "**INVOICE #**"
- Invoice Purpose-- "**FOR:** "
- Amount Due-- "**TOTAL**"
- Amount Due Date-- "**DUE ON**"

Exercise 10.3

String manipulation (II)
Extract Seller

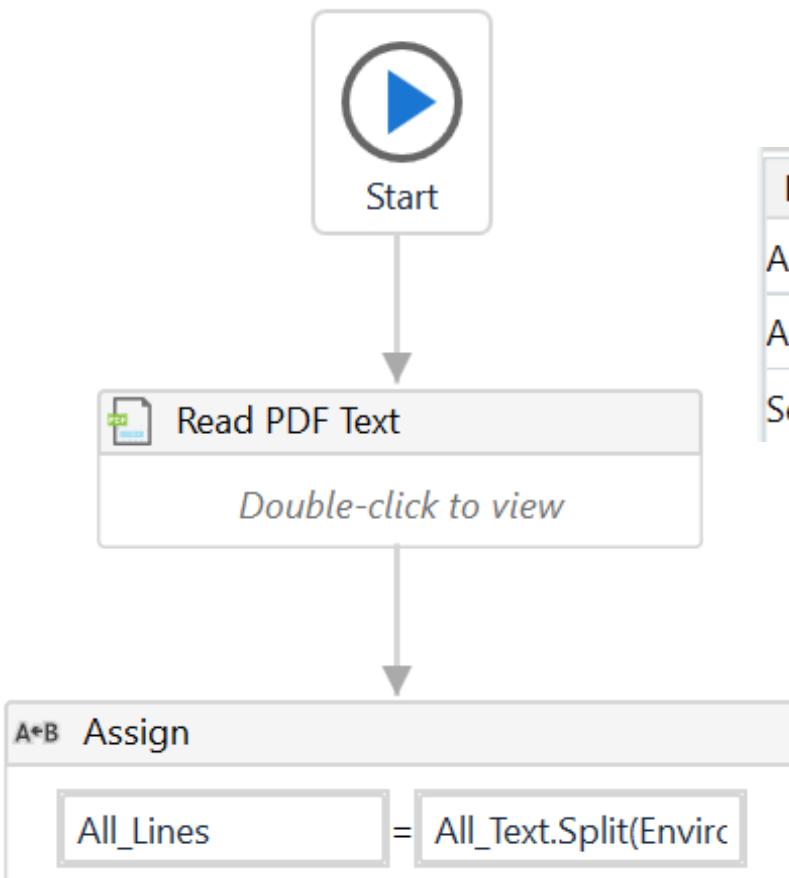
Exercise 10.3 (Step 1)

In UiPath Studio, save “Ex10.2” as “Ex10.3” .



Exercise 10.3 (Step 2)

1. Create a new string variable “Seller”.
2. Delete “For Each”



Name	Variable type	Scope
All_Text	String	Ex10_1
All_Lines	String[]	Ex10_1
Seller	String	Ex10_1

Exercise 10.3

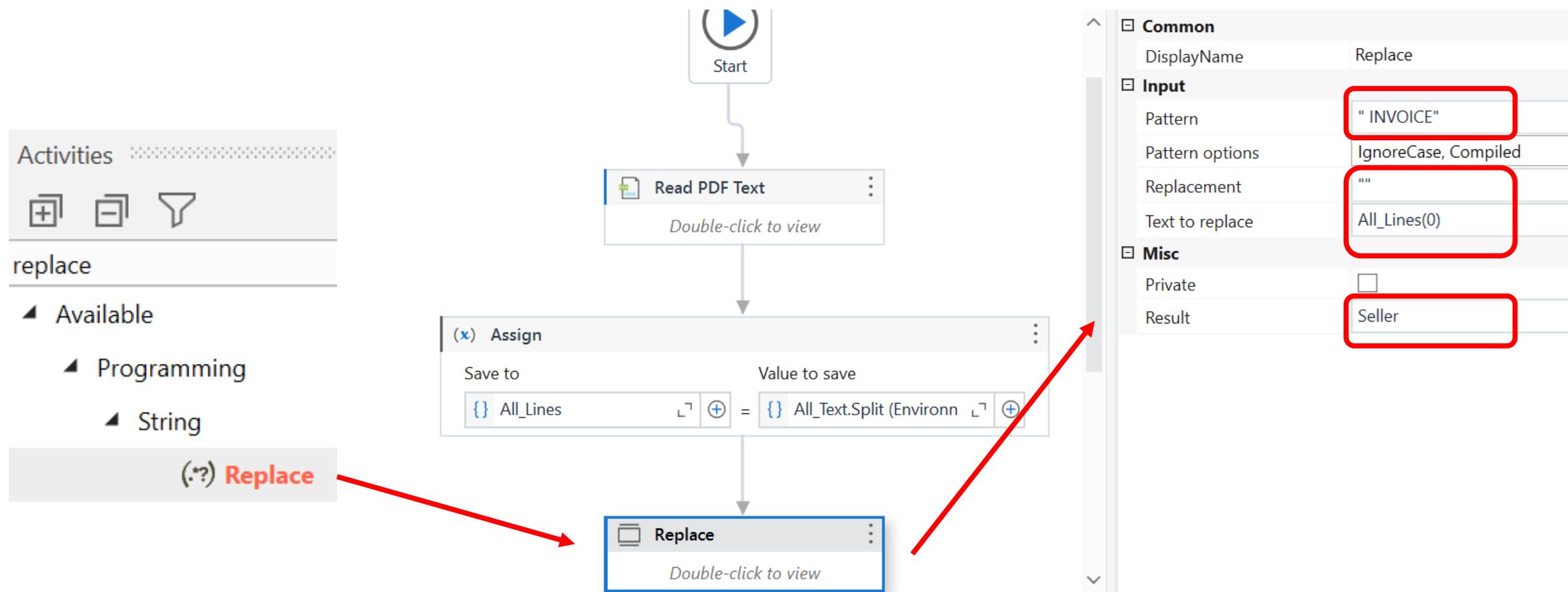
- As seller is found in the first line of the invoice, we will access that first line by using the first array element **All_Lines(0)**
- Likewise, the second line is **All_Lines(1)**
- Without performing a manual count, can you guess how the last line is assessed? It's **All_Lines(All_Lines.Count -1)**

All_Lines(0) ->	A c m e I n c I N V O I C E
All_Lines(1) ->	1 2 A c m e R o a d D A T E : 7 / 3 / 2 0 2 0
All_Lines(2) ->	S i n g a p o r e 1 2 3 4 5 6 I N V O I C E # A 1 0 0
:	
:	
:	
All_Lines(16) ->	T H A N K Y O U F O R Y O U R B U S I N E S S !

All_Lines.count where All.Lines is an array

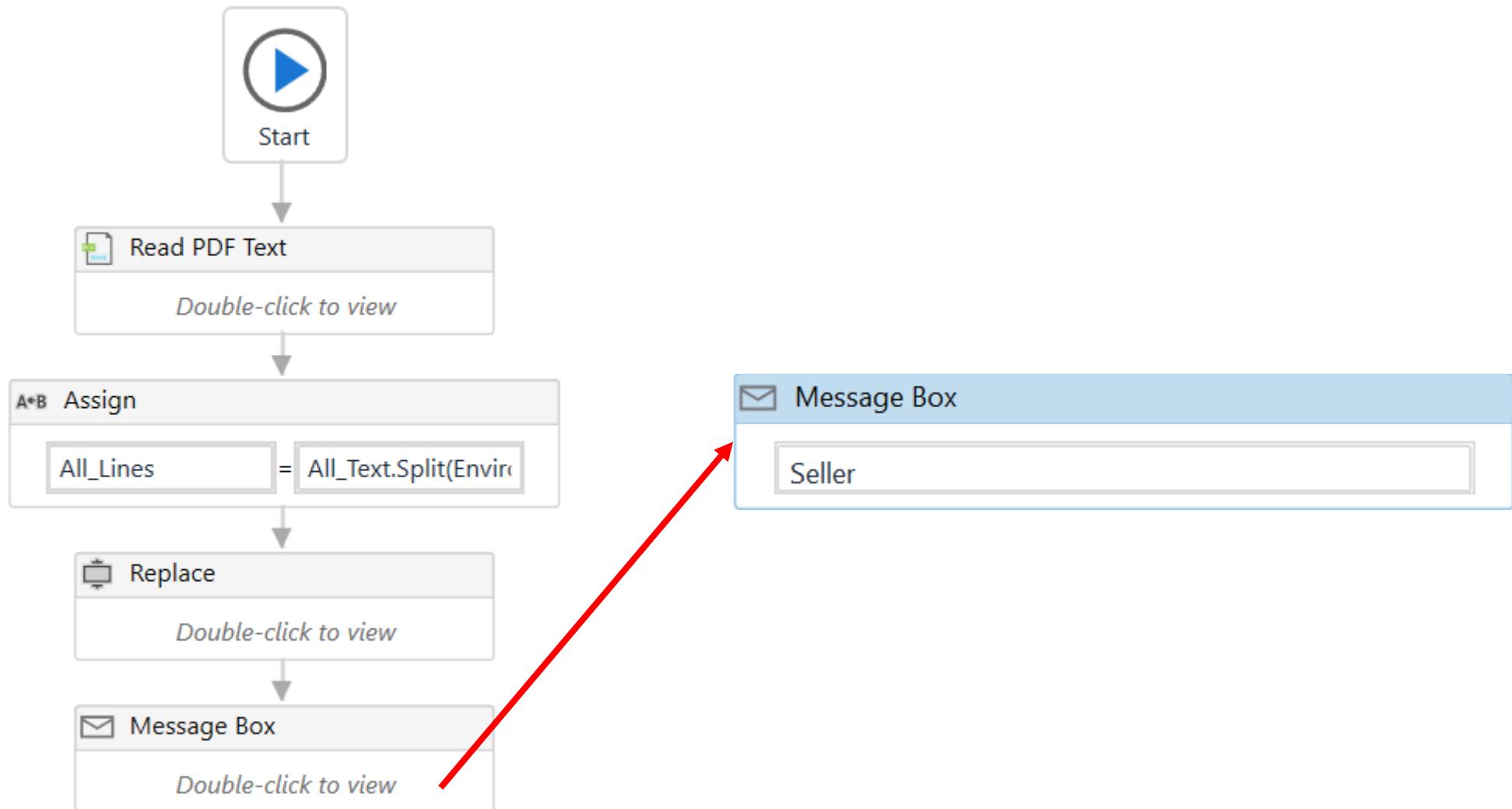
Exercise 10.3 (Step 3)

Use a “Replace” activity to substitute “INVOICE” with a null string, and save the output as “Seller”.



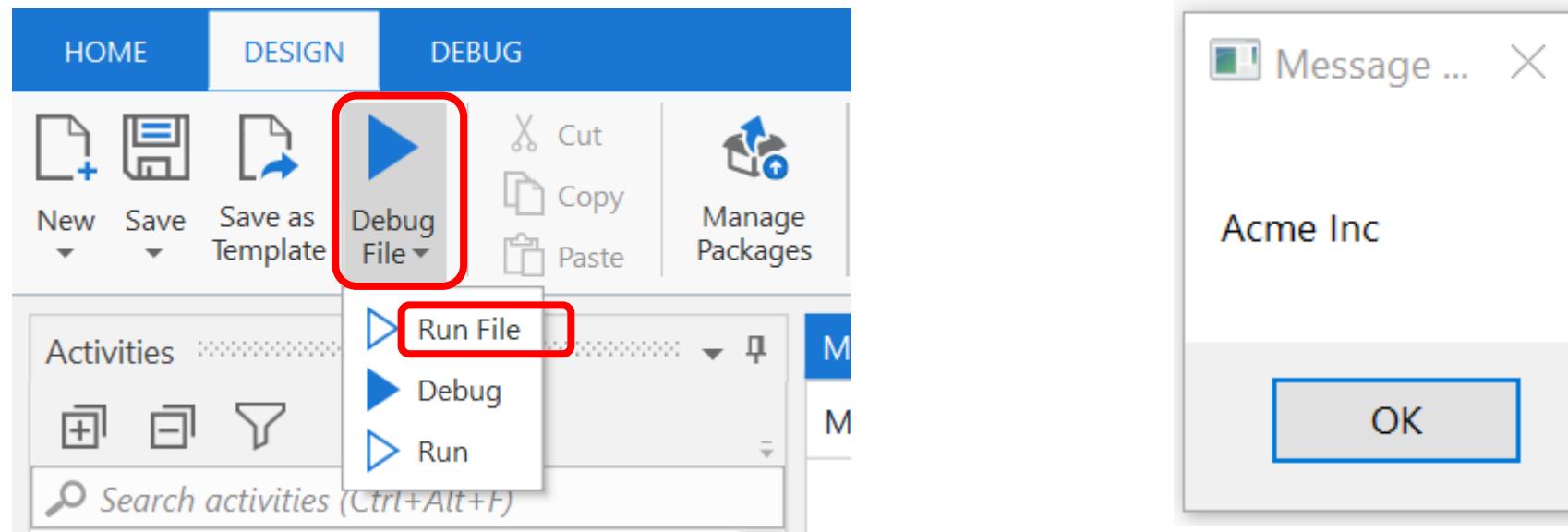
Exercise 10.3 (Step 4)

Add a “Message Box” to display the “Seller” variable.



Exercise 10.3 (Step 5)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message boxes.



Exercise 10.3 (Review)

- We can access different lines in the invoice directly. For example, we just used All_Lines(0) to access it directly
- An element in an array of string is a string, and string methods (e.g. Replace) can be used on it
- Experiment with 'Replace' to see what it can do

All_Lines(0) ->	A c m e I n c I N V O I C E
All_Lines(1) ->	1 2 A c m e R o a d D A T E : 7 / 3 / 2 0 2 0
All_Lines(2) ->	S i n g a p o r e 1 2 3 4 5 6 I N V O I C E # A 1 0 0
:	
:	
:	
All_Lines(16) ->	T H A N K Y O U F O R Y O U R B U S I N E S S !

Exercise 10.3 (Review)

We have extracted Seller

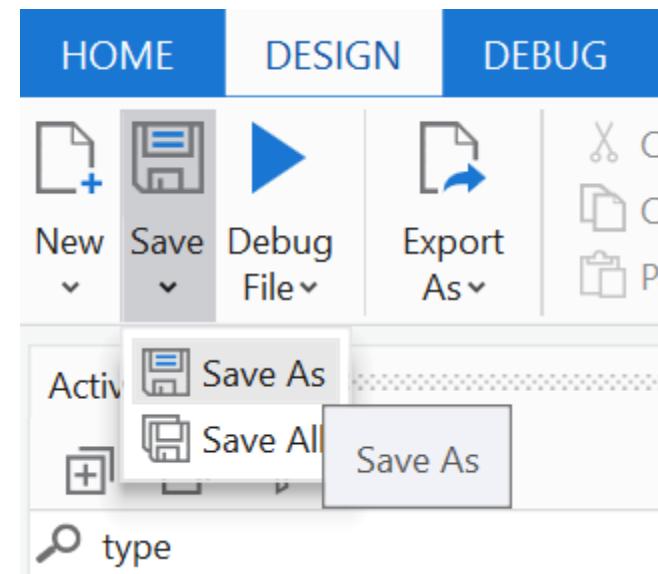
- Fields by position
 - ✓ Seller is found in the first line of the invoice
- For the following fields, we'll look for a keyword.
 - The person in your company who ordered the product is in the line immediately below the keyword "**Bill To**"
 - For Invoice Date, the keyword is "**DATE:**"
 - Invoice Reference Number-- "**INVOICE #**"
 - Invoice Purpose-- "**FOR:**"
 - Amount Due-- "**TOTAL**"
 - Amount Due Date-- "**DUE ON**"

Exercise 10.4

String manipulation (III)
Extract “Bill To”

Exercise 10.4 (Step 1)

In UiPath Studio, save “Ex10.3” as “Ex10.4” .



Exercise 10.4 (Step 2)

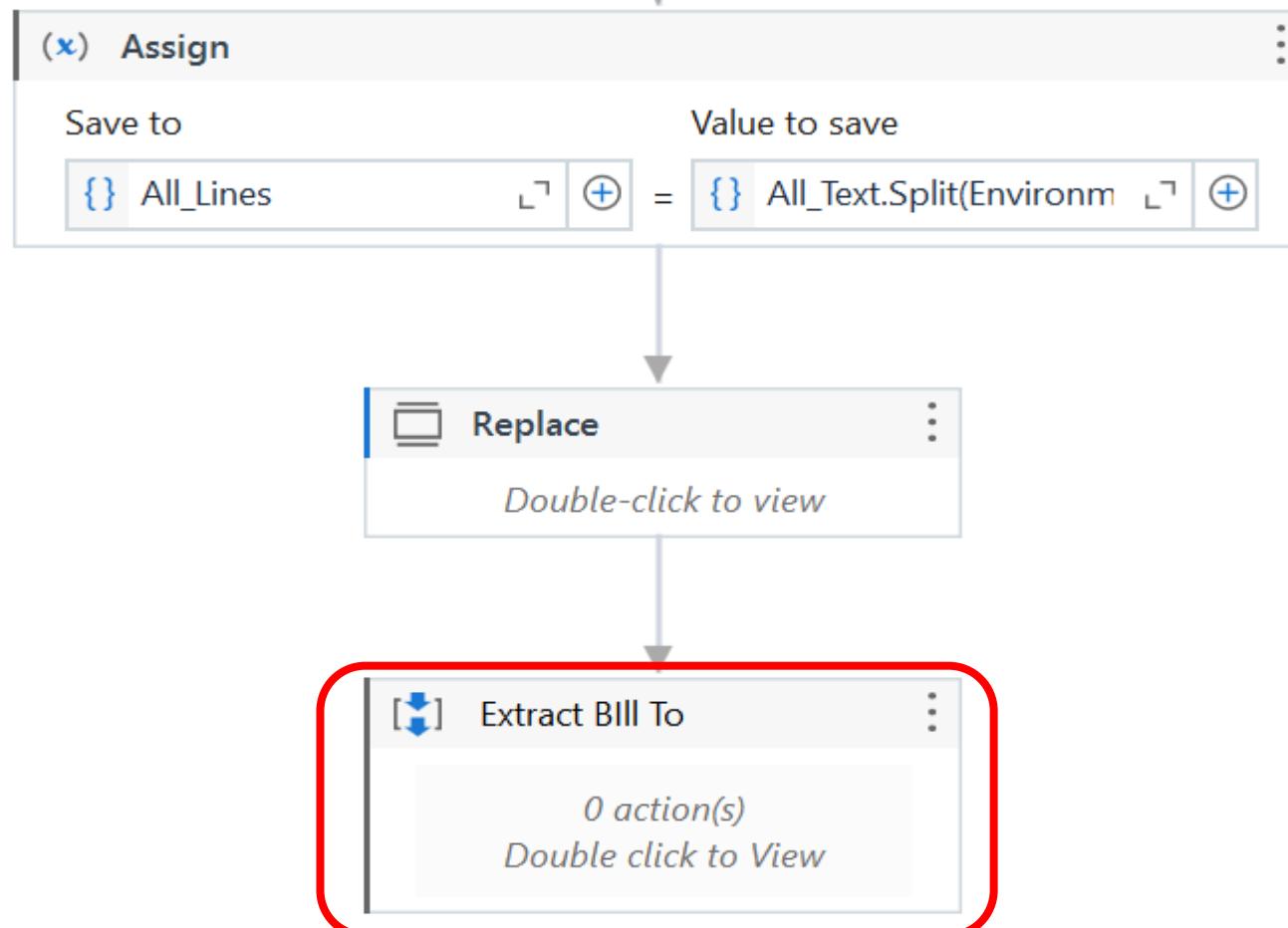
Create these new variables.

Name	Variable type	Scope
All_Text	String	Ex10_1
All_Lines	String[]	Ex10_1
Seller	String	Ex10_1
Bill_To	String	Ex10_1
Keyword_Start_Position	Int32	Ex10_1
All_Lines_Startng_From_Keyword	String[]	Ex10_1
All_Text_Startng_From_Keyword	String	Ex10_1

Exercise 10.4 (Step 3)

Del Messagebox

Insert a **Sequence** and rename it to “Extract Bill To”

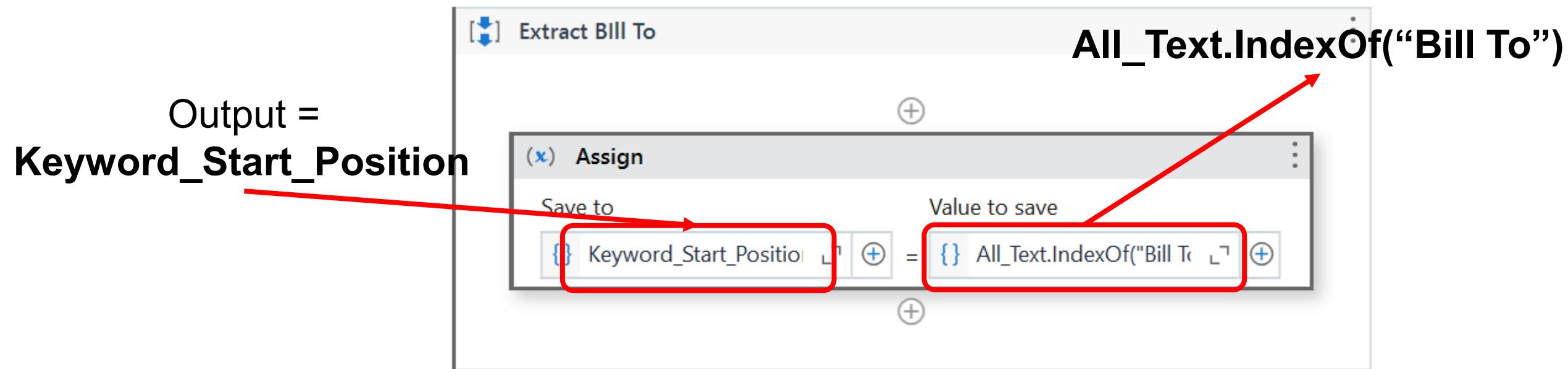


Exercise 10.4 (Step 4)

Double click the “Extract Bill To” sequence.

Insert an “**Assign**” activity.

We will use “**IndexOf**” to search for the starting position of the keyword “**Bill To**”.



Exercise 10.4

“**All_Text**” is a long string containing all of the invoice.

(\n indicates the start of a newline)

The “**Keyword_Start_Position**” points to the start of the keyword “**Bill To**”.

A	c	m	e	I	n	c	I	N	V	O	I	C	E	\n	1	2	A	c	m	R	o	a	d				
D	A	T	E	:	7	/	3	/	2	0	2	0	\n	S	i	n	g	a	p	o	r	e	1	2	3	4	5
6	:																										
B	i	l	l	T	o	:	\n	J	o	h	n	T	a	n	\n	M	e	C	o								
:\n																											
\n	T	H	A	N	K	Y	O	U	F	O	R	Y	O	U	R	B	U	S	I	N	E	S	S	!			

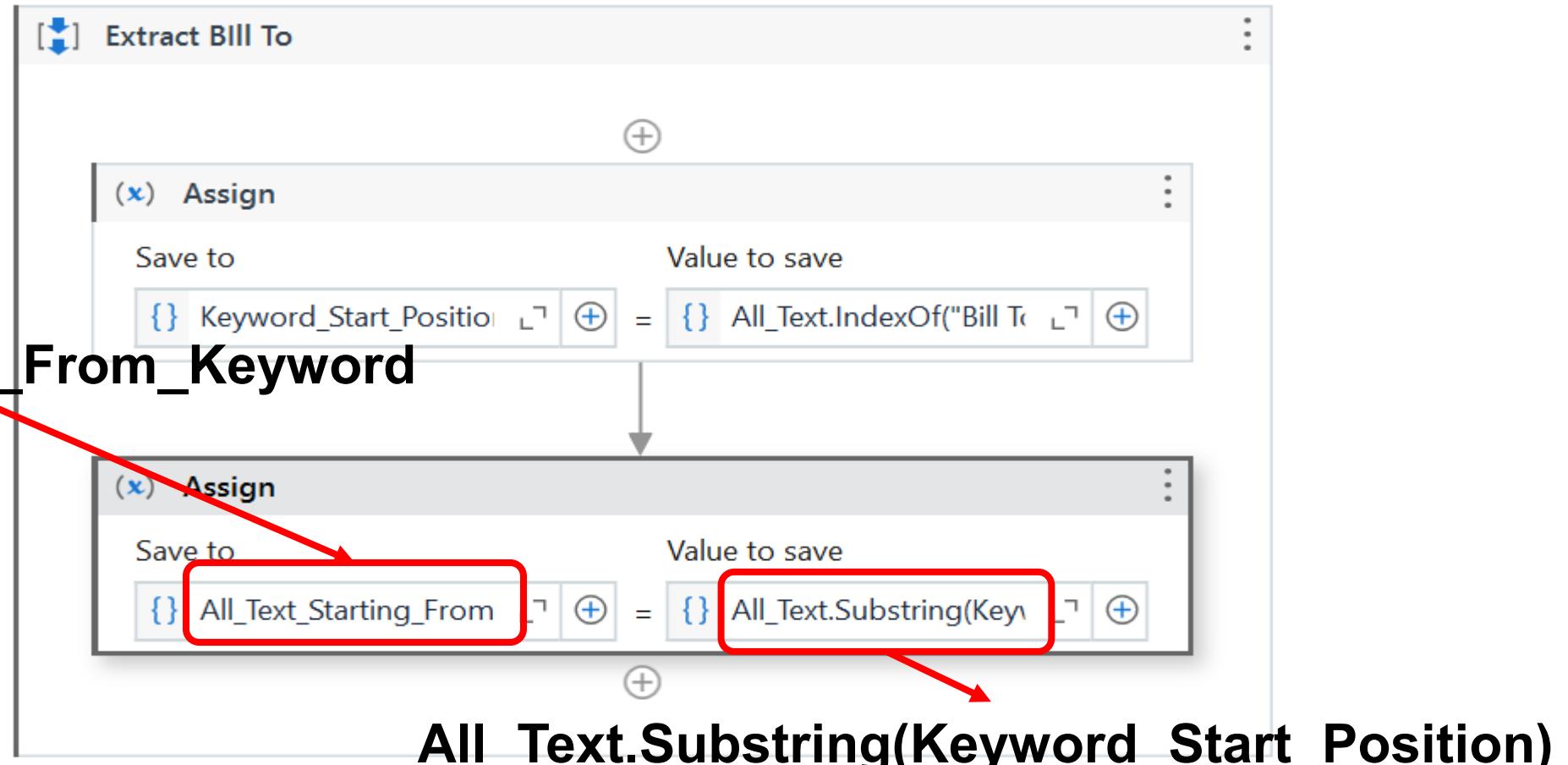
Keyword_Start_Position

Exercise 10.4 (Step 5)

Insert an “Assign” activity.

We will use “Substring” to extract all text starting from the keyword “Bill To”.

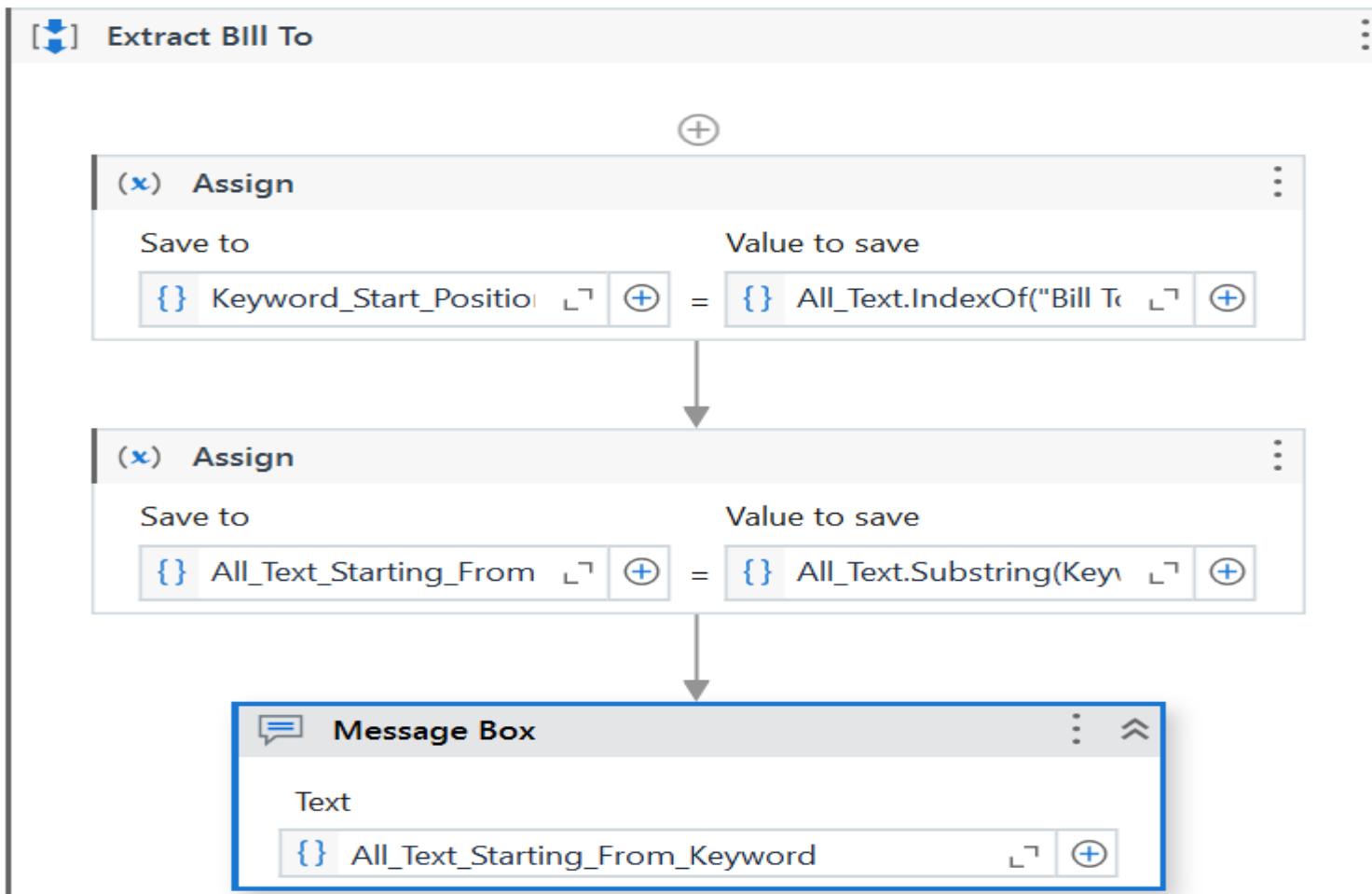
All_Text_Startng_From_Keyword



All_Text.Substring(Keyword_Start_Position)

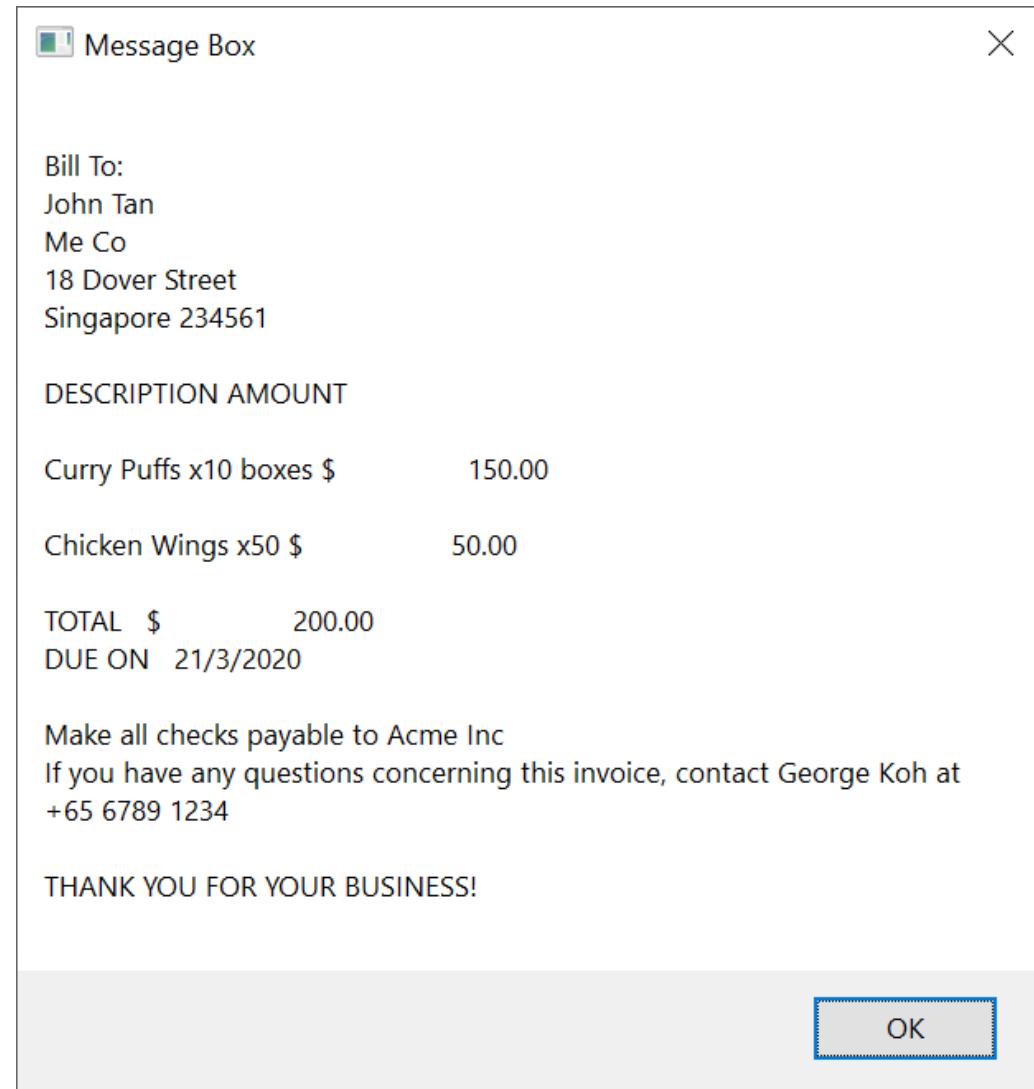
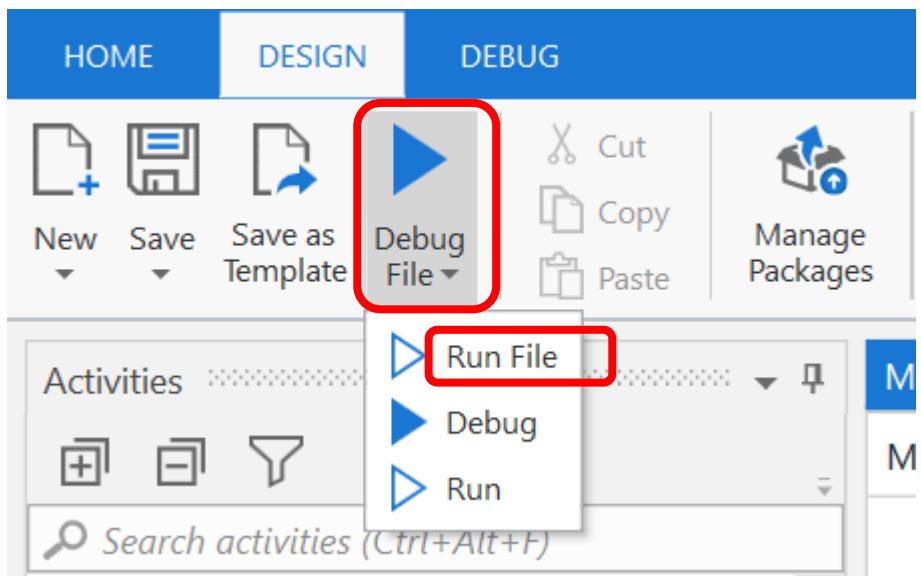
Exercise 10.4 (Step 6)

Use a “Message Box” to inspect the contents of string variable “All_Text_Starts_From_Keyword”.



Exercise 10.4 (Step 7)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message box.



Exercise 10.4

Using the substring command, we have extracted the text starting from keyword “Bill To” up till the end of the string.

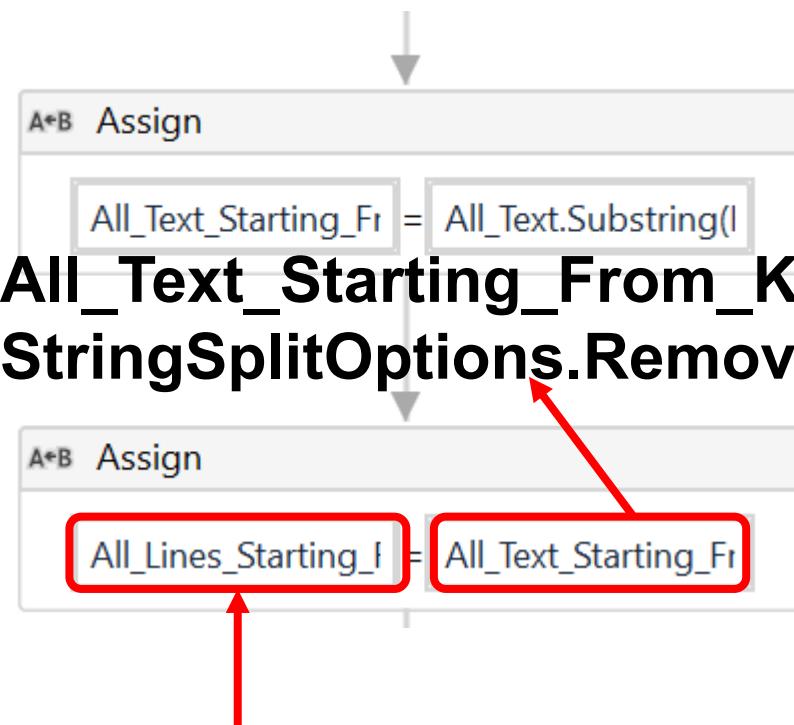
A	c	m	e	I	n	c	I	N	V	O	I	C	E	\n	1	2	A	c	m	e	R	o	a	d																								
D	A	T	E	:	7	/	3	/	2	0	2	0	\n	S	i	n	g	a	p	o	r	e	1	2	3	4	5																					
6	:	:	B	i		T	o	:	\n	J	o	h	n	T	a	n	\n	M	e	C	o	:	:	\n	T	H	A	N	K	Y	O	U	F	O	R	Y	O	U	R	B	U	S	I	N	E	S	S	!

All_Text_Startng_From_Keyword

Exercise 10.4 (Step 8)

Insert an “Assign” activity.

We will use “Split” to split “All_Lines_Startng_From_Keyword” into an array of substrings.

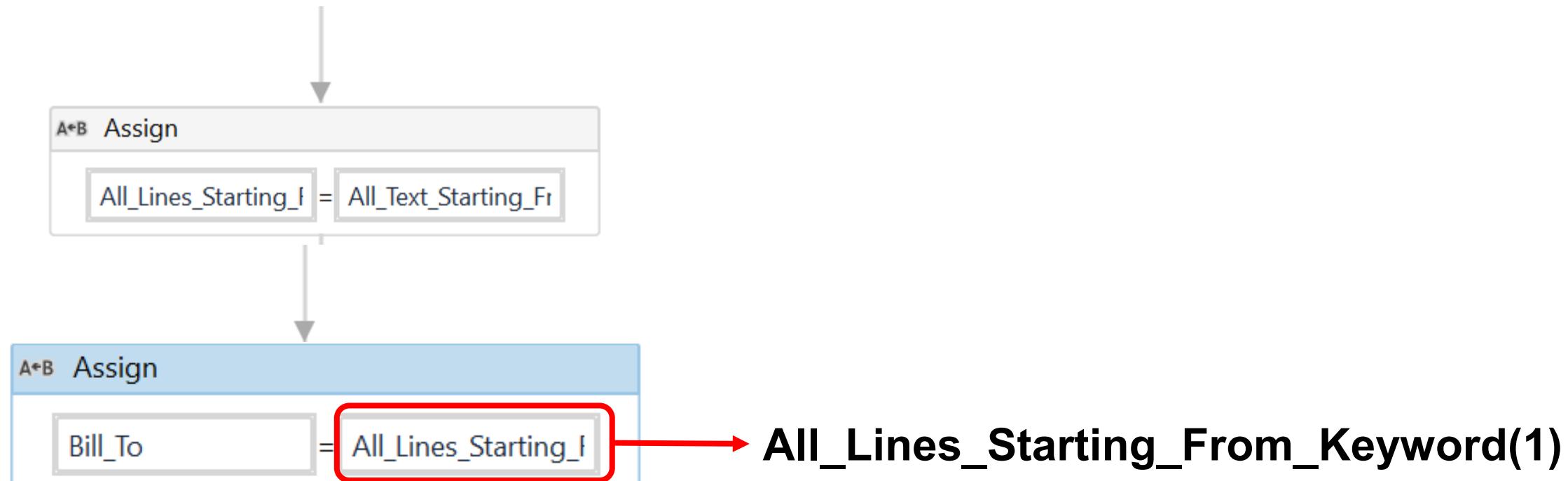


All_Lines_Startng_From_Keyword

Exercise 10.4 (Step 9)

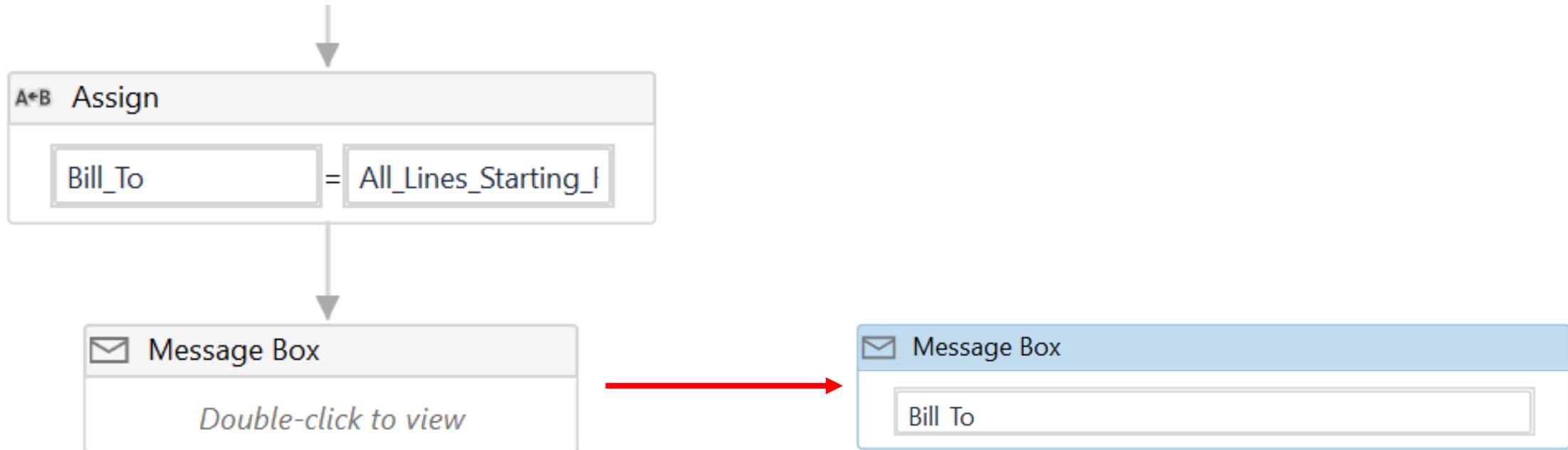
Insert an “Assign” activity.

We will extract “Bill_To” from “All_Lines_Startng_From_Keyword”.



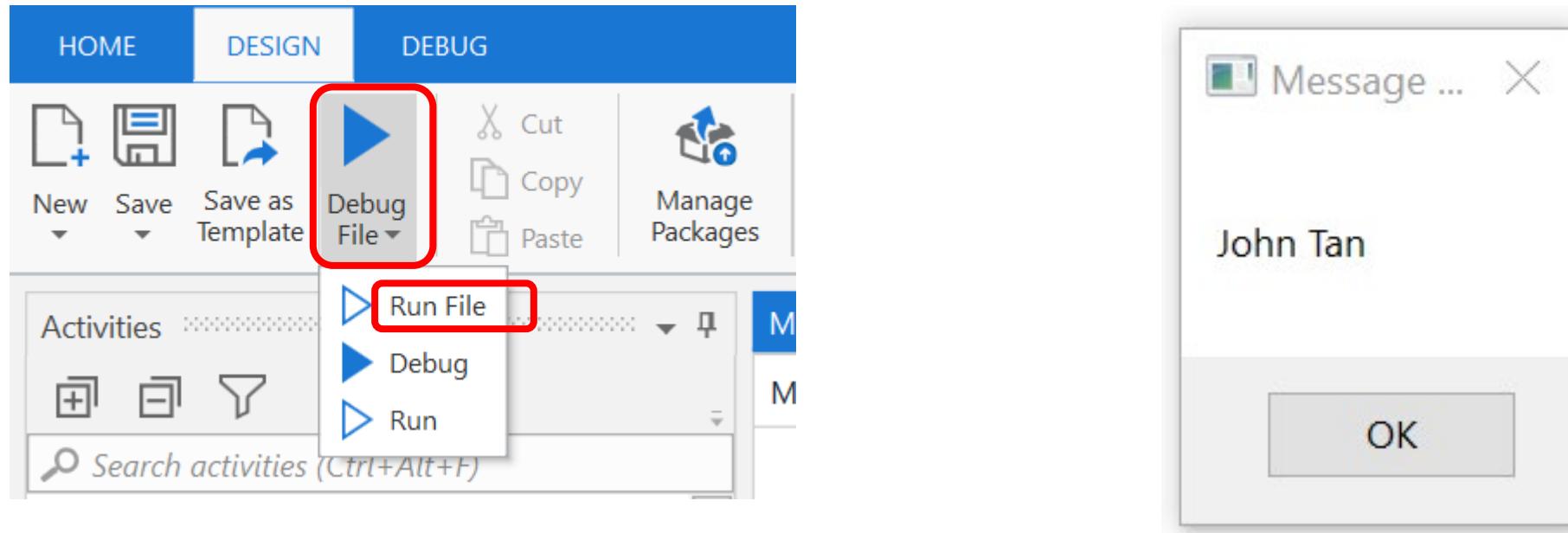
Exercise 10.4 (Step 10)

Edit the “Message Box” to display “Bill_To”.



Exercise 10.4 (Step 11)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message box.



Exercise 10.4

- Using the substring command, we have extracted the text starting from keyword “Bill To” up till the end of the string.
- Using “Split”, we have split “All_Lines_Startng_From_Keyword” into an array of substrings.
- We then extracted the “Bill To” variable from “All_Lines_Startng_From_Keyword(1)”.

All_Lines_Startng_From_Keyword(0)	->	B	i				T	o	:
All_Lines_Startng_From_Keyword(1)	->	J	o	h	n		T	a	n

Exercise 10.4 (Review)

- We've just used string methods to extract the person who requested the product/ services from the seller
- The technique makes assumptions about the format of the invoice.
- For good order, you might need a different script for each seller.

Exercise 10.4 (Review)

We have extracted Seller and Bill To

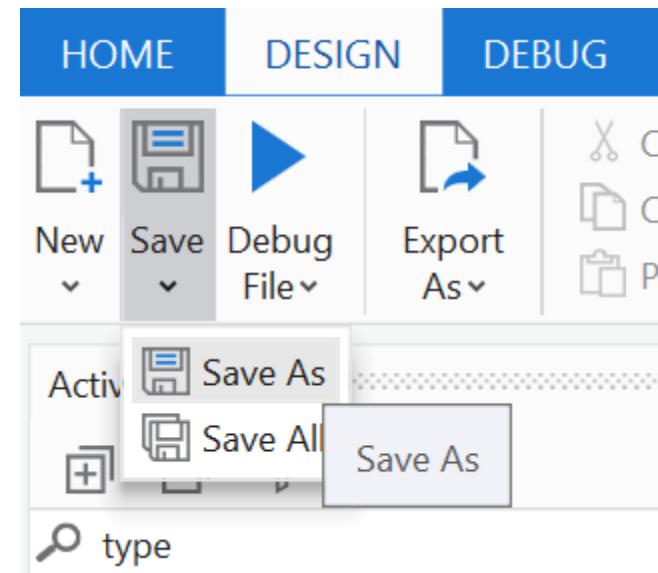
- Fields by position
 - ✓ Seller is found in the first line of the invoice (DONE!)
- For the following fields, we'll look for a keyword.
 - The person in your company who ordered the product is in the line immediately below the keyword "Bill To" (DONE!)
 - For Invoice Date, the keyword is "**DATE:**"
 - Invoice Reference Number-- "**INVOICE #**"
 - Invoice Purpose-- "**FOR:** "
 - Amount Due-- "**TOTAL**"
 - Amount Due Date-- "**DUE ON**"

Exercise 10.5

String manipulation (IV)
Extract Invoice Date

Exercise 10.5 (Step 1)

In UiPath Studio, save “Ex10.4” as “Ex10.5” .



Exercise 10.5 (Step 2)

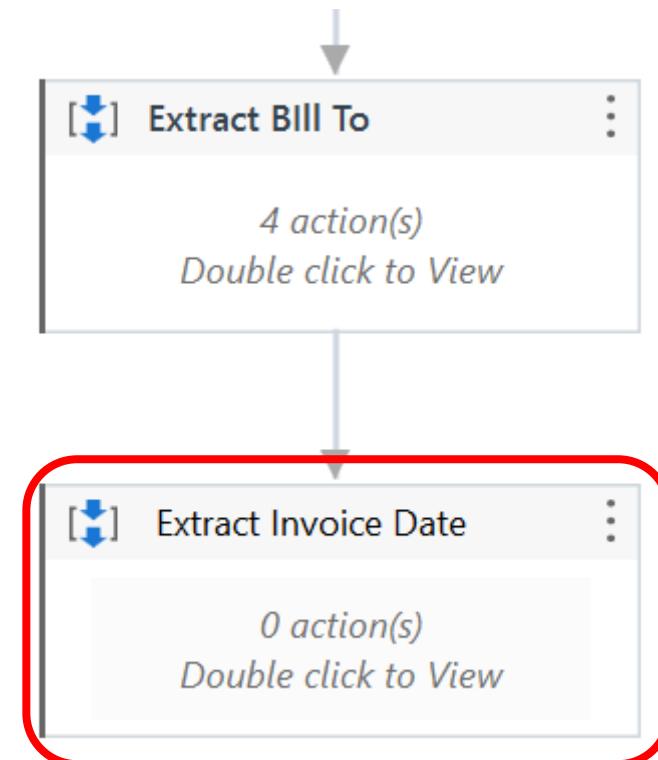
Create the following new variables.

Name	Variable type	Scope
Invoice_Date	String	Ex10_1
Invoice_Ref	String	Ex10_1
Invoice_Purpose	String	Ex10_1
Amt_Due	String	Ex10_1
Due_Date	String	Ex10_1
All_Lines_After_Kw	String[]	Ex10_1
All_Text_After_Kw	String	Ex10_1
Keyword	String	Ex10_1



Exercise 10.5 (Step 3)

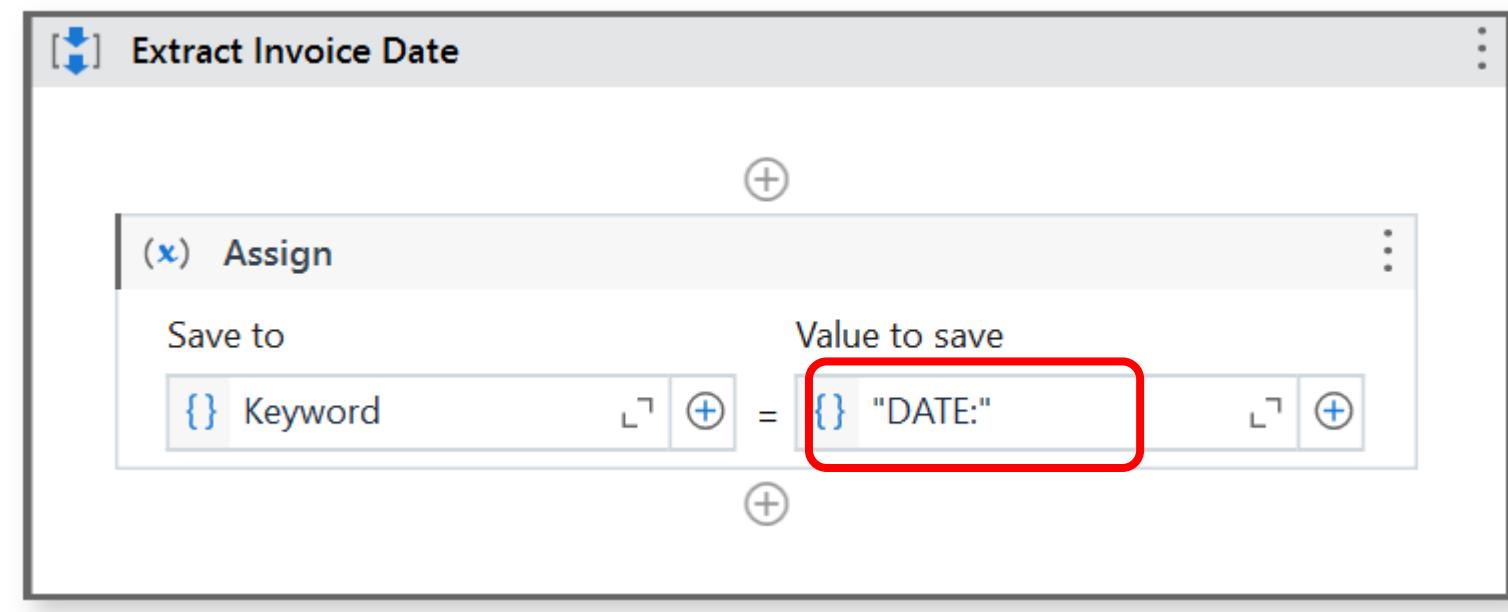
Insert a **Sequence** and rename it to
“Extract Invoice Date”.



Exercise 10.5 (Step 4)

Double click this sequence.

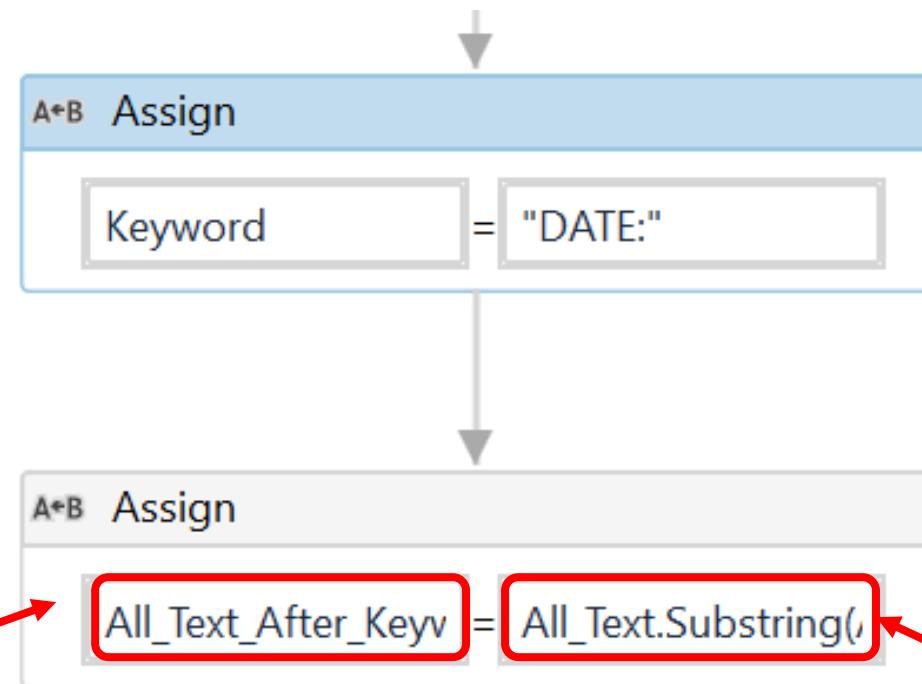
Insert an “**Assign**” activity to specify the keyword to search for as “**DATE:**”.



Exercise 10.5 (Step 5)

Insert an “Assign” activity.

We will use “IndexOf” to locate the position of the text “DATE:”.



All_Text_After_Keyword

All_Text.Substring
(All_Text.IndexOf(Keyword)+Keyword.Length)

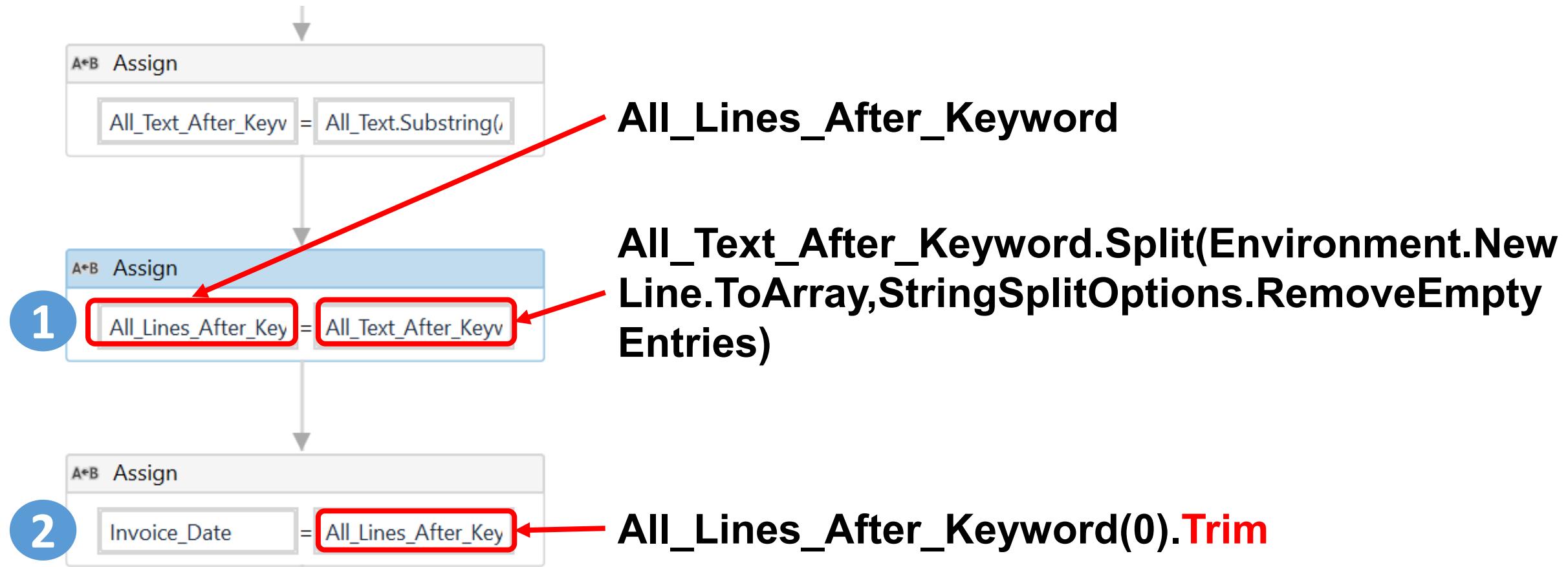
Exercise 10.5 (Step 6)

1. `IndexOf("DATE:")` returns this position
2. Add `Keyword.length` to arrive here. Yes, it's a blank character. We'll use "Trim" to remove preceding and trailing blank spaces
3. Now "**All_Text_After_Keyword**" is the greyed area

```
A c m e   I n c   I N V O I C E \n 1 2   A c m e   R o a d
1 D A T E :   7 / 3 / 2 0 2 0 \n S i n g a p o r e   1 2 3 4 5
2 :
:
B i l l   To : \n J o h n   T a n \n M e   C o
:
:
\n T H A N K   Y O U   F O R   Y O U R   B U S I N E S S !
```

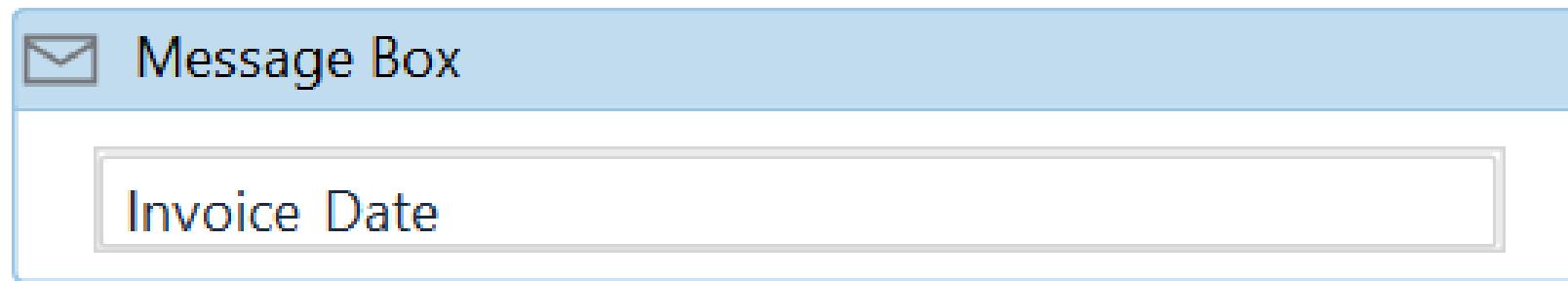
Exercise 10.5 (Step 7)

1. Insert an “Assign” activity. We will use “Split” to isolate the line with the variable “Invoice_Date”.
2. Insert another “Assign” activity to trim the variable “Invoice_Date”.



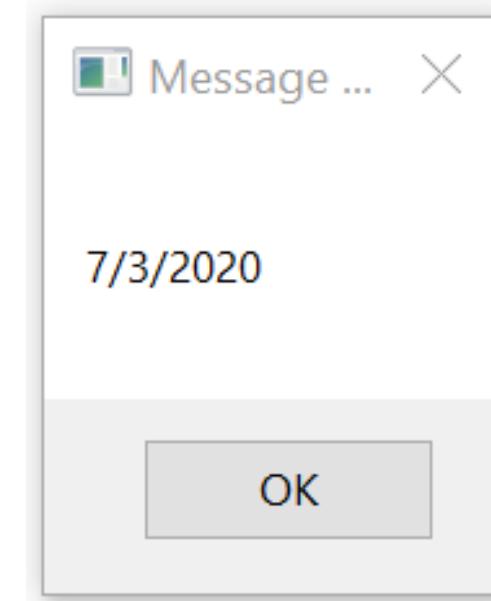
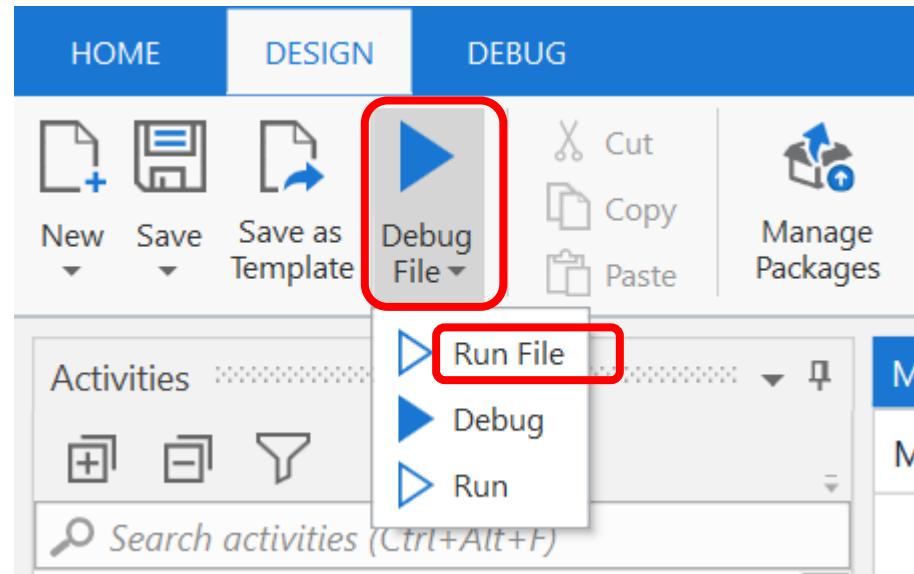
Exercise 10.5 (Step 8)

Change the input text of the “Message Box” to **Invoice Date**.



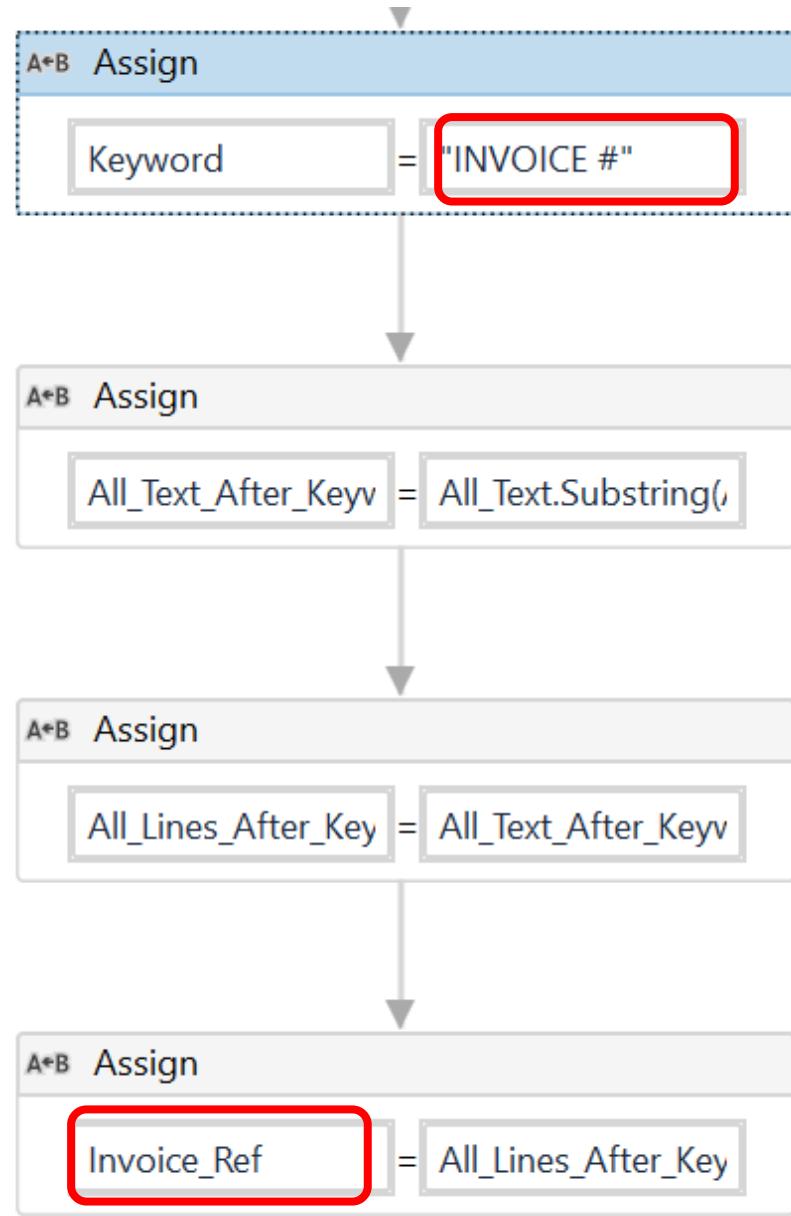
Exercise 10.5 (Step 9)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the message box.



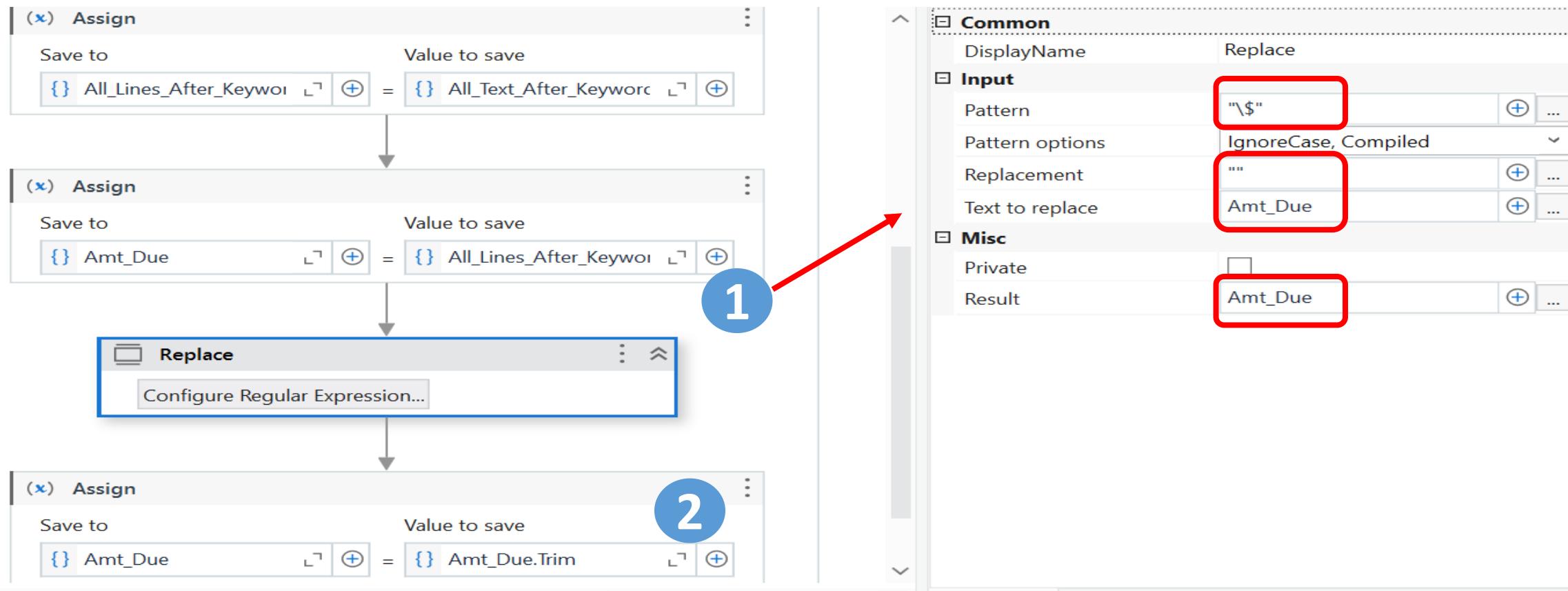
Exercise 10.5 (Step 10)

1. Delete the “Message Box”.
2. Insert a Sequence for each of the below.
3. Repeat with respective keywords in each Sequence created in step 2.
 - Invoice Reference Number-- **"INVOICE #"**
 - Invoice Purpose-- **"FOR: "**
 - Amount Due-- **"TOTAL"**
 - Amount Due Date-- **"DUE ON"**
4. Change the **keyword** and the **destination variable** for each field.



Exercise 10.5 (Step 11)

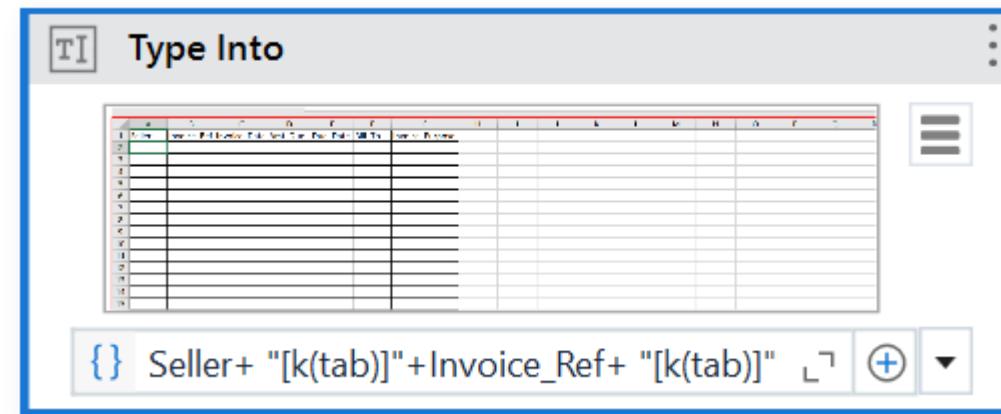
1. Insert an “Replace” activity to do some post-processing for “Amt_Due”. The dollar sign that was captured can be removed.
2. Finally, insert another “Assign” to trim “Amt_Due” again.



Exercise 10.5 (Step 12)

1. Open the spreadsheet “Invoice_file.xlsx”. Click on the leftmost column in the first empty row
2. Ensure scope of all variable is “Ex10.1”
3. Insert only ONE “Type Into” activity boxes to input the extracted variables into the spreadsheet:

Seller
 Invoice_Ref
 Invoice_Date
 Amt_Due
 Due_Date
 Bill_To
 Invoice_Purpose

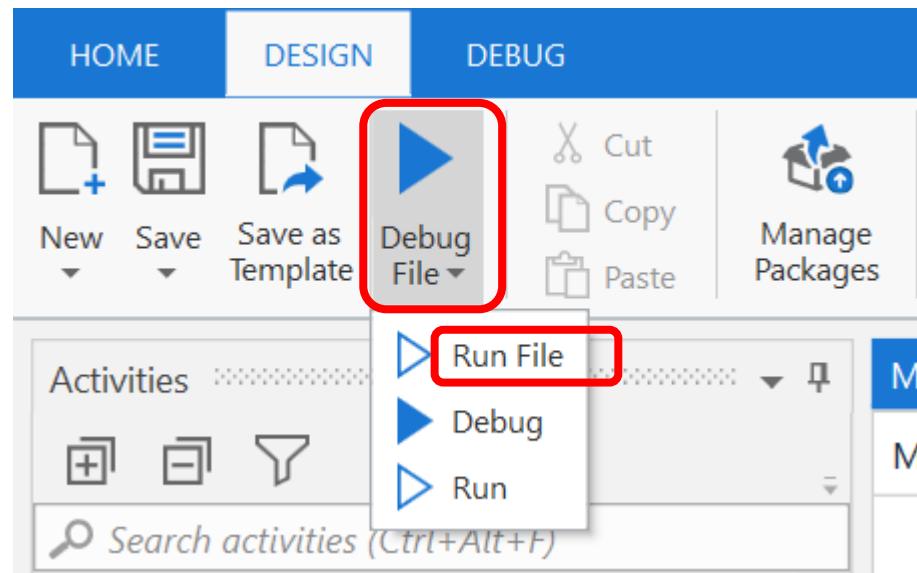


Seller+ "[k(tab)]"+Invoice_Ref+ "[k(tab)]"+Invoice_Date+ "[k(tab)]"+Amt_Due+ "[k(tab)]" + Due_Date + "[k(tab)]"+Bill_To+ "[k(tab)]"+Invoice_Purpose

[k(Tab)] sends a **Tab** keystroke to Excel, and sends the active cell one column to the right.

Exercise 10.5 (Step 13)

Click the **Debug File** button and select “Run File” to run your script. Inspect the contents of the Excel spreadsheet.



A	B	C	D	E	F	G
Seller	Invoice_Ref	Invoice_Date	Amt_Due	Due_Date	Bill_To	Invoice_Purpose
Acme Inc	A100	7/3/2020	200	21/3/2020	John Tan	Catered Food



Exercise 10.5 (Review)

- Optical Character Recognition could be used, but there are trade-offs

Help Resources

- UiPath Studio Documentation

<https://studio.uipath.com/>

- UiPath Video Tutorials

<https://www.uipath.com/tutorials>

- UiPath Academy

<https://www.uipath.com/rpa-academy>

- AUTOMATION BEST PRACTICES – UI Automation

<https://studio.uipath.com/docs/ui-automation>

- UiPath Community Forum

<https://forum.uipath.com/>

- YouTube tutorial e.g.

<https://www.youtube.com/watch?v=lgIChrGzzoU>

List of Actions

Send HotKey

Click

Variable

Copy Text

Type into

Do While

Assign

If Then Else

While

Exercise 5.1

Exercise 5.2

Exercise 5.3

Exercise 5.4

Exercise 5.5

Read Range

For Each Row

Get Row Item

Input Dialog

Open Browser

Build Data Table

Add Data Row

Write Range

Exercise 5.6

Exercise 6.1

Exercise 6.2

List of Actions

For Each

Exercise 7.1

SubString
length, location
replace, split

Exercise 10a-10e

Append Range

Exercise 7.2

Read PDF

Exercise 10.1

Send email

**Exercise 8.1
- 8.2**

More String
manipulations

Exercise 10.2-10.3

VLookUp

Exercise 9.1

Excel Pivot

Exercise 9.2



Your **RPA** journey
has just started!