ROBOTIC PROCESS AUTOMATION

Chapter-3: Sequence, Flowchart, and Control Flow

Chapter-3: Sequence, Flowchart, and Control Flow

- We have seen how simple it is to train a UiPath Robot by recording the activities of a task and running it.
- In this chapter, we will learn about methods to lay down activities in an orderly fashion and how to control the flow.
- These are basic to any kind of programming.
- We will learn about putting activities in Sequences, Flowcharts, and loops.
- We will also look at logical control using if-else.

Sequencing the workflow

UiPath provides four types of projects:

1. **Sequences** (used for simple automation)
2. **Flowcharts** (used for simple automation)
3. **User Events** (beneficial for implementing front office robots)
4. **State Machines** (used for dealing with complex business processes)

What is a Sequence?

- A Sequence is a group of logical steps.
- Each step represents an action or a piece of work.
- A Sequence is used for processes that happen in linear succession, that is, one after the other.
- Among the three types of projects in UiPath, Sequences are the smallest.

What is a Sequence? (To be performed practically)

In the following example, we will make a simple project that asks for the name of the user and then displays his or her response:

1. Open UiPath Studio and click on Blank to start a fresh project.
2. Give it a meaningful name.
3. On the Designer panel, drag and drop a Flowchart activity from the Activities panel.
4. Search for Sequence in the Activities panel and then drag and drop it into the Flowchart.
5. Double-click on the Sequence.
6. We now have to add the steps that we want to perform.
7. Consider each step as an action.
8. We can add many steps inside a Sequence. For the sake of simplicity, we will add two steps:
    1. Ask for the username in an Input dialog
    2. Display the username in a Message box

What is a Sequence? (To be performed practically)

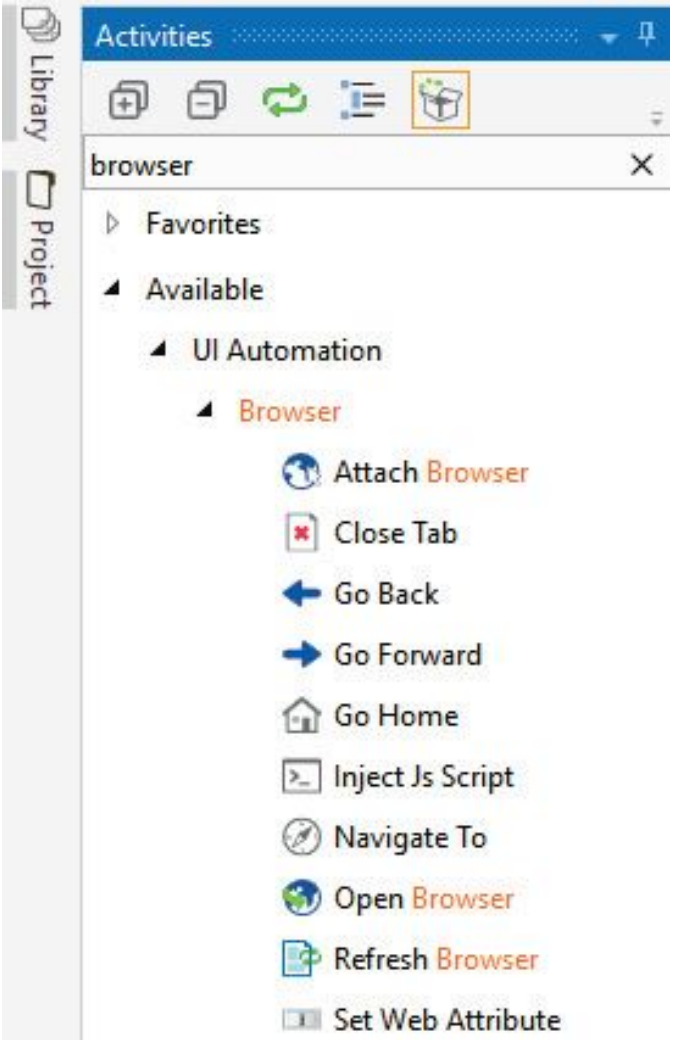
9. Search for Input dialog in the Search panel of the Activities panel.
10. Drag and drop the Input dialog activity inside the Sequence
11. Write the appropriate message on the Label of this Input dialog to ask for the user's name.
12. Drag and drop a Message box activity into the Sequence
13. Next, create a variable and give it the desired name.
14. This variable will receive the text that the user has entered in the Input dialog box in response to our question, that is, the user's name
15. We now have to specify the Result property (in the Properties panel) of the Input dialog box.
16. On specifying the variable name there, it will receive the text that the user entered.
17. Click on the dotted icon that appears on the right side of the Result property.
18. Now, specify the variable name.

What is a Sequence? (To be performed practically)

19. Specify the variable name that we have created in the Text area of the Message box (the Text area of the Message box is used to input text that will be displayed in the Message box).
20. We just need to connect the Sequence to the Start icon.
21. This can be done by right-clicking on the Sequence activity and choosing the Set as Start node option.
22. Hit the Run button and see the result.

Activities

- An activity represents the unit of an action.
- Each activity performs some action.
- When these activities combine together, it becomes a process.
- Every activity resides on the Activities panel of the main Designer panel.
- You can search for a particular activity and use it in your project.
- For example, when we search for browser, all the browser activities will appear in the Activities panel, as shown in the following screenshot



Using activities with workflows

We have seen how we can easily search for a particular activity. Now, let us see how to use them in a workflow:

1. Search for Flowchart in the same way that we have searched for the browser activities in the Activities panel search bar
2. Drag and drop the Flowchart activity inside the Designer panel.
3. The Flowchart appears in the Designer panel and we have a given Start node.
4. The Start node specifies where the execution begins.
5. We are ready to use different activities in our Flowchart.
6. You can use any activity/activities inside the Flowchart
7. For the sake of simplicity, let us just use a Write line activity.
8. Drag and drop the Write line activity inside the Flowchart.
9. Set its text property by providing a string value
10. Connect this Write line activity with the Start node by right-clicking on the Write line activity and selecting Set as Start Node.

Using activities with workflows

11. We have made a smaller module and now it is time to extract it as a workflow.
12. Right-click on the main Designer panel and choose Extract as Workflow
13. A window will pop up asking for the name.
14. Give it a meaningful name and click on Create.
15. This will be the name of your workflow
16. We have just used activities and extracted them in a workflow
17. It automatically generates the Invoke test Workflow activity.
18. Now, when we run the program, it will invoke the workflow that we have extracted (double-click on the Invoke test workflow activity to see which workflow it is going to invoke and where it is generated).

Control flow, various types of loops, and decision making

- Control flow refers to the order or the particular manner in which actions are performed in an automation.
- UiPath provides numerous activities for performing the decision-making process.
- These activities, present in the Activities panel, are put into the workflow either using the double click method or the drag and drop method.

Control flow, various types of loops, and decision making

Different types of control flow activities are as follows:

1. **The Assign activity**
2. **The Delay activity**
3. **The Break activity**
4. **The While activity**
5. **The Do While activity**
6. **The For each activity**
7. **The If activity**
8. **The Switch activity**

Control flow, various types of loops, and decision making

1. The Assign activity:

- The Assign activity is used to designate a value to the variable.
- The Assign activity can be used for different purposes, such as incrementing the value of a variable in a loop, or using the results of a sum, difference, multiplication, or division of variables and assigning it to another variable.

2. The Delay activity:

- The Delay activity, as the name suggests, is used to delay or slow down an automation by pausing it for a defined period of time.
- The workflow continues after the specified period of time.
- It is in the **hh:mm:ss** format

Control flow, various types of loops, and decision making

3. The Break activity:

- The Break activity is used to break/stop the loop at a particular point, and then continue to the next activity according to the requirement.
- It cannot be used for any other activity apart from the For each activity.

4. The While activity:

- The While activity is used in automation to execute a statement or process based on a certain condition.
- If found true, the loop is executed; that is, the process is executed repeatedly
- This activity is useful while iterating through an array of elements

5. The Do while activity:

- The Do while activity is used in automation when it is required to execute a statement based on the fulfilment of a certain condition.
- How it differs from the While activity is that it executes a statement, then checks whether the condition is fulfilled.
- If the condition is not fulfilled, it exits the loop.

Control flow, various types of loops, and decision making

6. The For each activity:

- The For each activity works by iterating each element from the collection of items or list of elements, one at a time.

7. The If activity:

- The If activity consists of a statement with two conditions: true or false.
- If the statement is true, then the first condition is executed; if not, the second condition is executed.

8. The Switch activity:

- The Switch activity can be used to make a choice.
- When we have various options available and want to execute one option, we frequently use the Switch activity.
- By default, the Switch activity takes an integer argument.

Control flow, various types of loops, and decision making

https://docs.uipath.com/studio/docs/the-break-activity

(you can see this link for all the 8 activities & their examples)

Step-by-step example using Sequence and Flowchart

- A Sequence and a Flowchart are similar concepts.
- They are both used to contain logical steps or actions.
- Sequence is generally used to contain multiple steps to perform an action.
- A Flowchart, on the other hand, is suitable for a particular task.
- When we have lots of steps of a similar kind, we contain them in a Sequence.

(Explain this with sequence example i.e. to display your name & flowchart example i.e. using activities with workflows demo & merge it together)

Step-by-step example using Sequence and Control flow

(Explain this with sequence activity & any of the 8 control flow activities example)

----------End Of Chapter 3----------

ROBOTIC PROCESS AUTOMATION

Unit-2

Chapter-4: Data Manipulation

Chapter-4: Data Manipulation

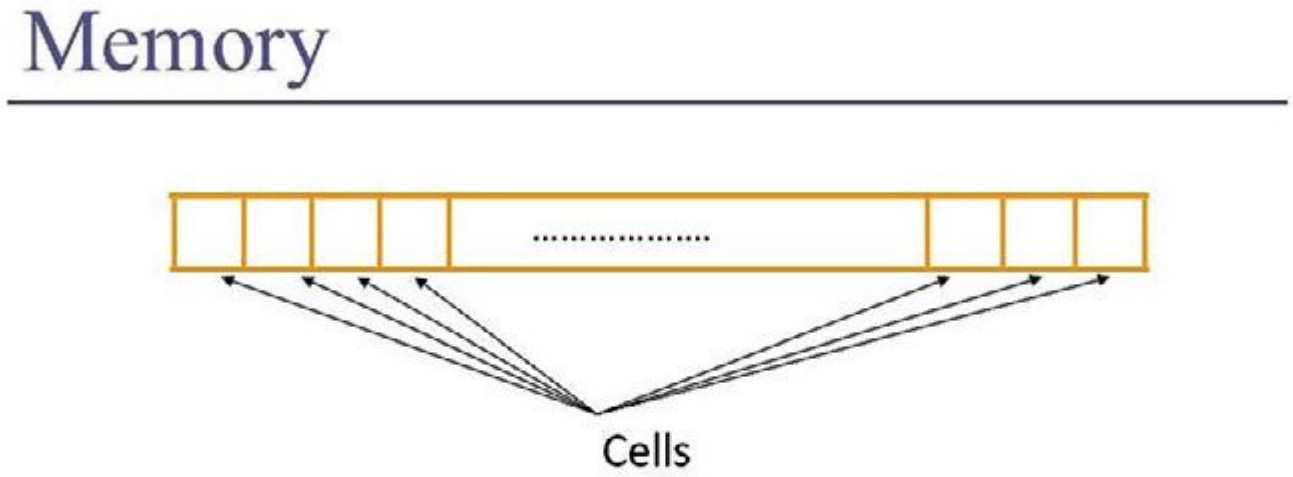
- So far we have learned about the basics of RPA and how to organize steps in a workflow using a Flowchart or Sequence.
- We now know about UiPath components and have a thorough understanding of UiPath Studio.
- We used a few simple examples to make our first robot.
- Before we proceed further, we should learn about variable and data manipulation in UiPath.
- It is not very different from other programming concepts.

Chapter-4: Data Manipulation

- Data manipulation is the process of changing data whether it is adding, removing, or updating it.
- Before learning about data manipulation, we shall see what variables, collections, and arguments are, what kind of data they store, and what their scope is.

Variables and scope

• Before discussing variables, let us take a look at Memory and its structure:



Each cell can store 1 bit of information having the value 0 or 1

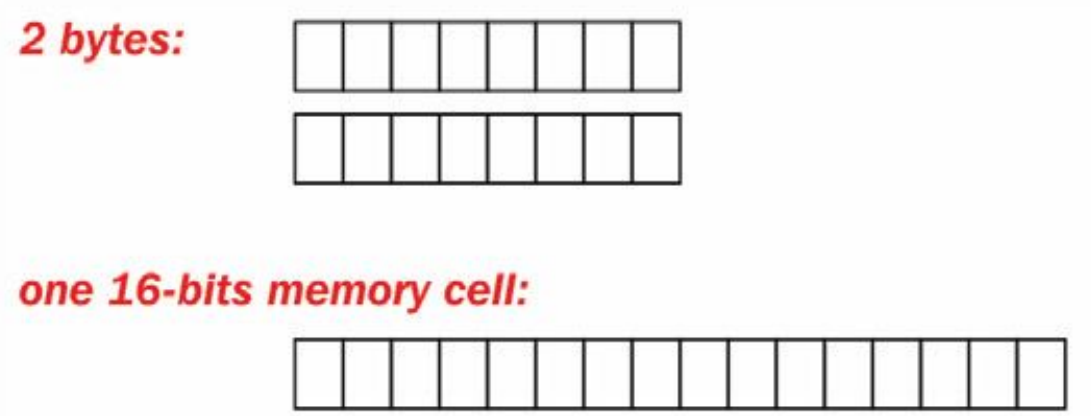
• Memory consists of millions of memory cells and each memory cell stores data in the form of 0s and 1s (binary digits).

• Each cell has a unique address, and by using this address, the cell can be accessed:

Variables and scope

- When data is stored in memory, its content gets split into further smaller forms (binary digits).

- As shown in the preceding diagram, 2 bytes of data consists of several memory cells.



- A variable is the name that is given to a particular chunk of memory cells or simply a block of memory and is used to hold data.

- You can declare any desired name and create a variable to store the data.

Variables and scope

- It is recommended, however, that we use meaningful variable names.
- For example, if we wish to create a variable to store the name of a person, then we should declare

  Name: Andy

- It is a good practice to create meaningful variable names.
- This becomes very useful in debugging the program.
- A variable is used to store data.
- Data is present around us in different types it can be an mp3 file, text file, string, numbers, and so on.
- That is why variables are associated with their respective data types.
- A particular type of variable can hold only that type of data.
- If there is a mismatch between the data and the variable type, then an error occurs

Variables and scope

• The following table shows the type a of variable available with UiPath:

Type	Content
Integer	Whole numbers
String	Text of any kind: "The Quick Fox @4598"
Boolean	True or false
Generic	Anything

• In UiPath, we can declare a variable in the Variables section.

• Just give it a meaningful name and select the appropriate type from the drop-down list.

Variables and scope

- We can also specify the scope of a variable.
- The Scope is the region under which the data has its effect or availability.
- You can choose the Scope of the variable according to your requirements; try to limit it as far as possible.
- For security reasons, it is not a good practice to set the Scope of your variable to fullest as it may accidentally be accessed by another region or could be modified.

(NOTE: Include example given in your book for variable & scope)

Collections

There are different types of variables. Variables can be classified into three categories:

1. Scalar: These are variables that can only hold a single data point of a particular data type, for example; Character, Integer, Double, and so on.
2. Collections: These are variables that can hold one or more data point of a particular data type. For example; array, list, dictionary, and so on.
3. Tables: These are a tabular form of the data structure which consists of rows and columns.

- We are going to see how collections work and how we can store values in the collection variables.
- In a collection, we can store one or more data points, but all the data must be the same.
- Consider an example.
- An array is a collection in which we can store different values of a particular data type.
- It is a fixed data type, meaning if we store five values inside the array, we cannot add or remove any value/values in that array.

(NOTE: Include example given in your book for collections)

Arguments-Purpose and use

- An Argument is simply a variable that can store a value.
- You can create an argument in the Argument section of the main Designer panel.
- But remember, they are not limited to variables.
- An argument has a larger scope than a variable and is used to pass values between different workflows.
- These Arguments are used for interacting with different workflows by exchanging data between them.
- That is why the direction property is associated with Arguments.
- We can choose the direction on the basis of our requirement - either giving the value to some workflow or receiving the value from another workflow.

Arguments-Purpose and use

- We can easily create arguments in the Arguments panel. We can also specify the direction:

1. In: When we have to receive the value from another workflow.
2. Out: This is the current value if we have to send the value to a workflow.
3. In/Out: This specifies both; it can take or receive the value.
4. Property: This specifies that it is not being used currently

(NOTE: Include example given in your book for Arguments)

Data table usage with examples

- A data table is a tabular form of data structure.
- It contains rows and each row has columns, for example

Student name	Roll number	Class
Andrew Jose	1	3
Jorge Martinez	2	3
Stephen Cripps	3	2

- The preceding illustration is an example of a data table that has three rows and three columns.
- You can build a data table in UiPath also.
- We shall build two projects in which we will use a data table:

1. Building a data table
2. Building a data table using data scraping (dynamically)

(NOTE: Explain the above two examples given in your book for data table usage)

Clipboard management

- Clipboard management involves managing the activities of the clipboard, for example, getting text from the clipboard, copying selected text from the clipboard, and so on.

(NOTE: Explain the example given in your book for clipboard management)

File operation with step-by-step example

- In this module, we are going to operate on Excel file.
- The following are the methods that are frequently used with an Excel file:

1. Read cell: This is used to read the value of a cell from an Excel file
2. Write cell: This activity is used to write a value in a cell of an Excel file
3. Read range: This is used to read the value up to the specified range. If the range parameter is not specified, it will read the entire Excel file
4. Write range: This is used to write a collection of rows into the Excel sheet. It writes to the Excel file in the form of a data table. Hence, we have to supply a data table
5. Append range: This is used to add more data into an existing Excel file. The data will be appended to the end.

- Once you get familiar with these methods, it will become very easy for you to use other methods too.

(NOTE: Explain the example of each method given in your book for file operation)

CSV/Excel to data table and vice versa (with a step-by-step example)

- In this section, we will see how to extract data from an Excel file into a data table and vice versa.
- We will achieve this by:

1. Reading an Excel file and creating a data table using data from the Excel file
2. Creating a data table and then writing all its data to an Excel file

(NOTE: Explain both the steps as mentioned in your book for CSV/Excel to data table & vice versa)

----------End Of Chapter 4----------