

Step-by-step example using Sequence and Flowchart, Step-by-step example using Sequence and Control flow

RPA AAT- B7

Manav Sharma
Prince Sherawat
Simran Paliwal
Monika K

(1BY21CS088)
(1BY21CS136)
(1BY21CS228)
(1BY22CS407)

Step-by-Step Example Using Sequence and Flowchart

1

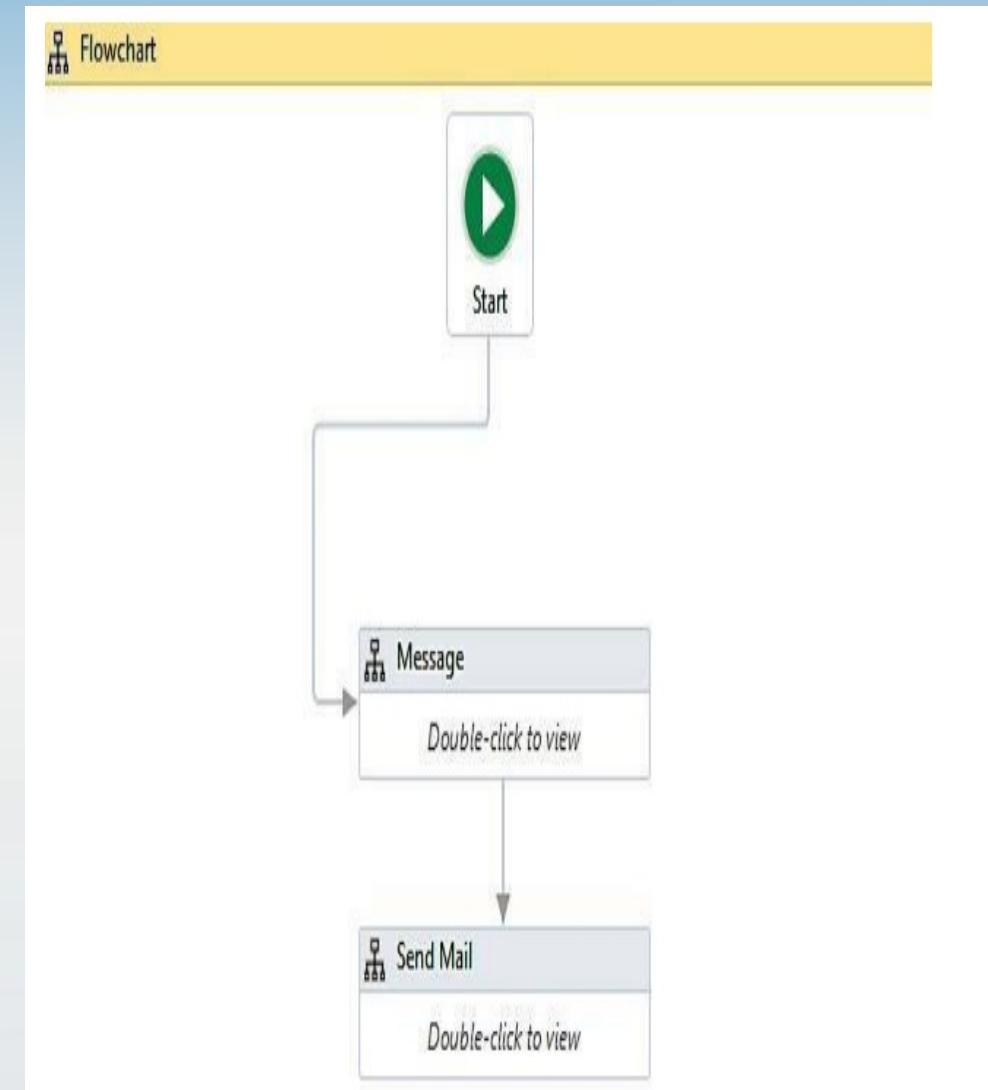
How to use a sequence

There may be different Sequences doing their jobs. We can easily put similar Sequences into a workflow; each workflow represents a task.

2

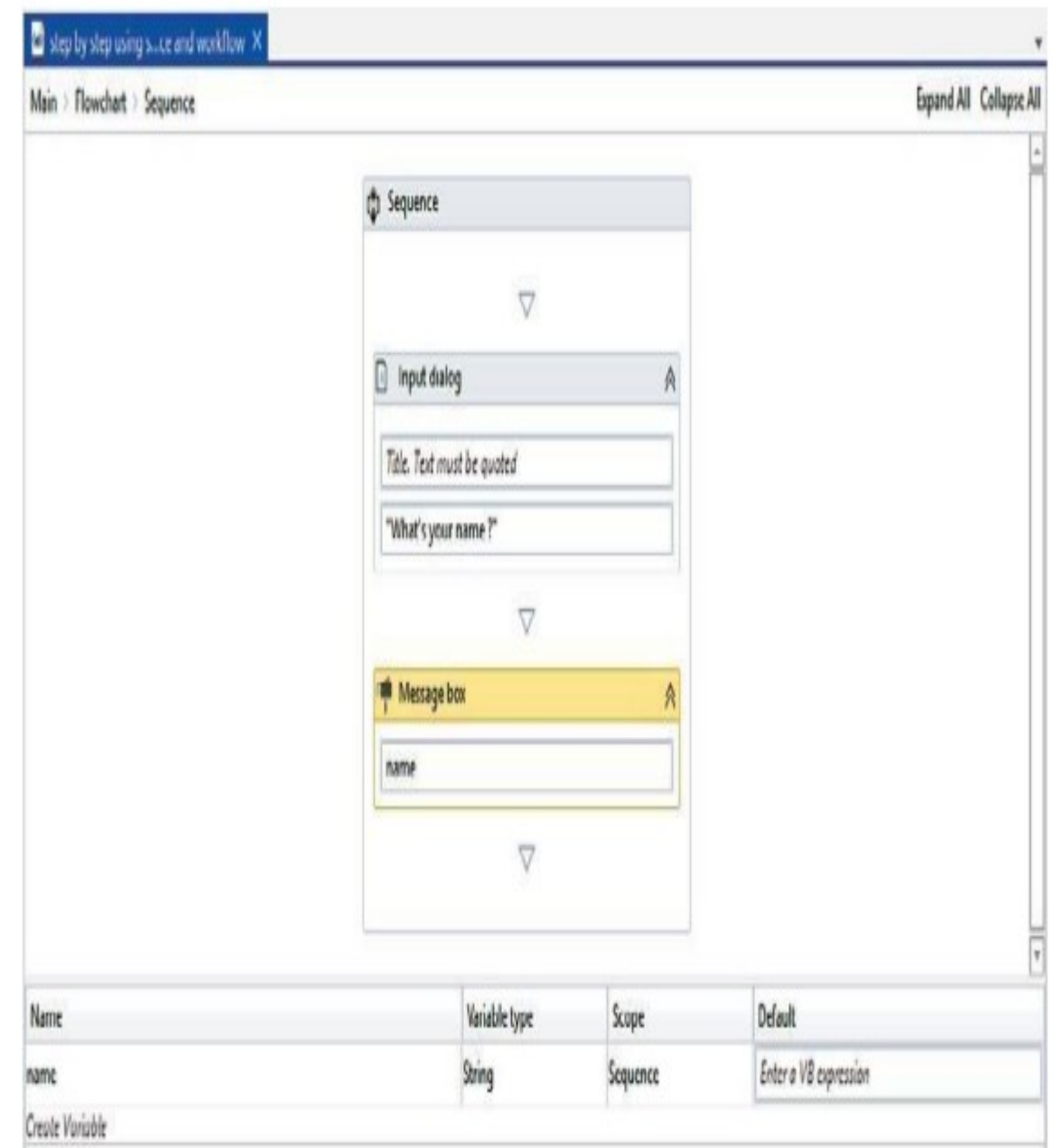
How to use a flowchart

A Flowchart is a container. It can contain activities inside it.



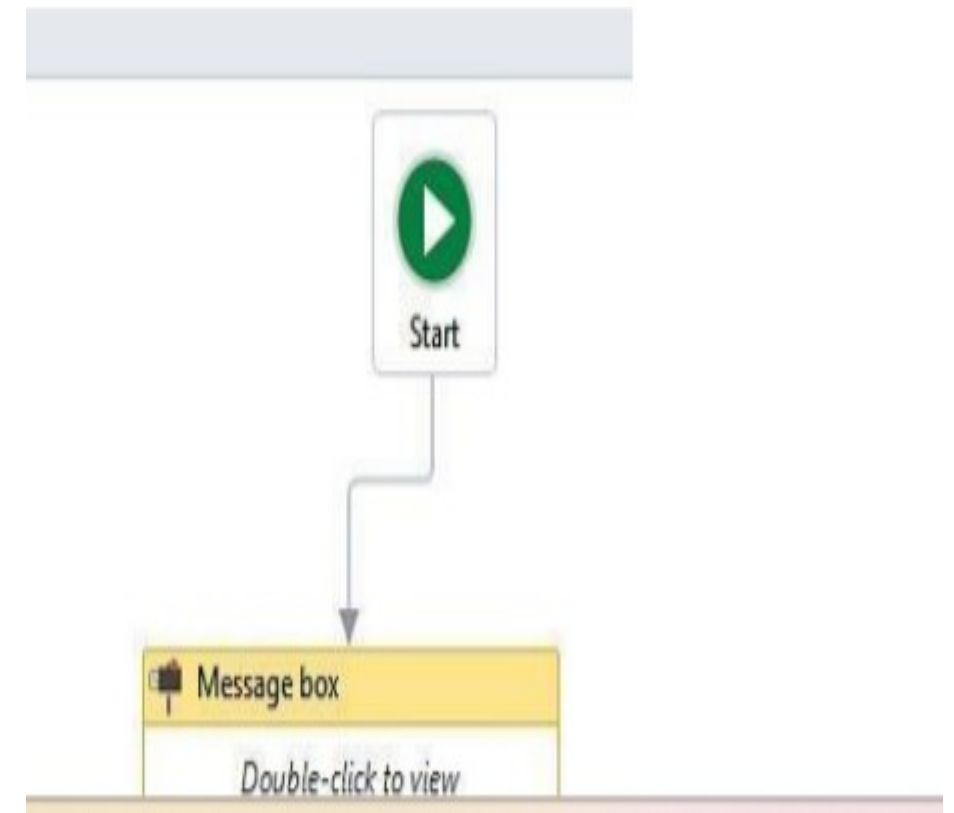
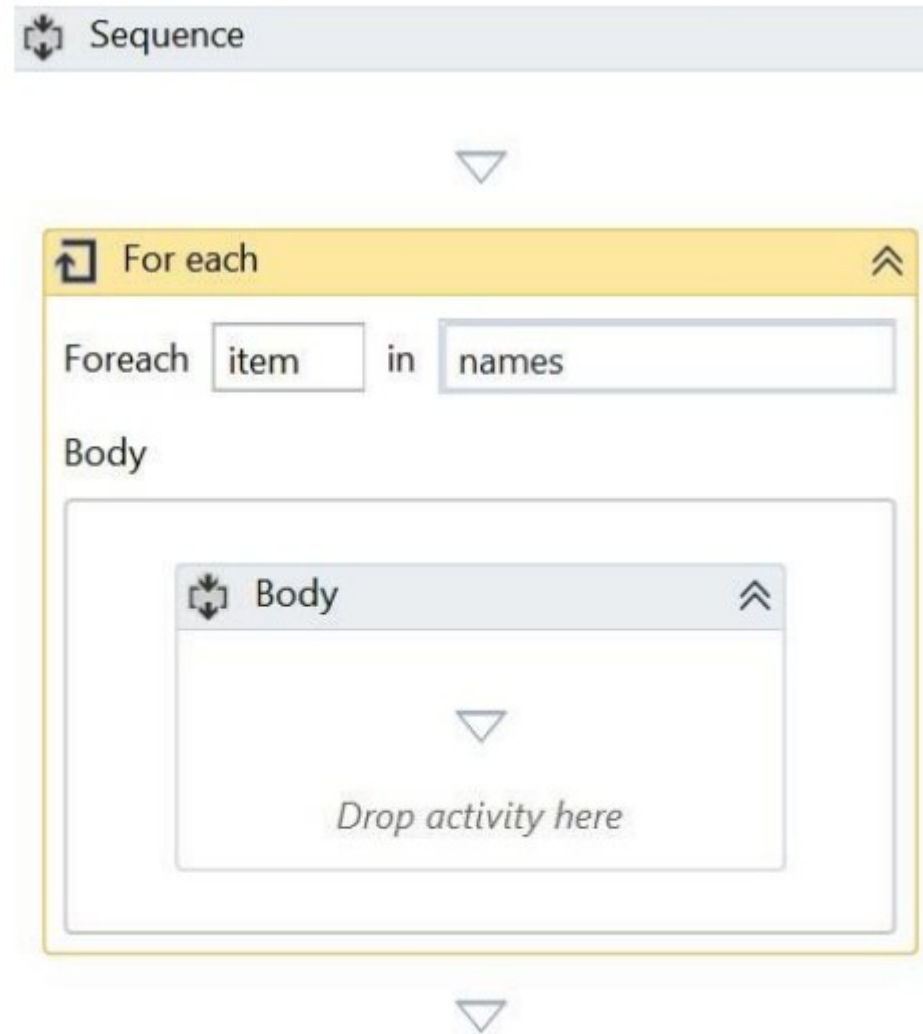
Steps to use a sequence:

- Drag and drop a Flowchart onto the Designer panel. Drag and drop a Sequence activity. Connect the Sequence activity with the Start node.
- Double click on the Sequence activity. Drag and drop an Input dialog activity and a Message box activity. Specify a message in the label property of the Input dialog activity.
- Create a variable of type String. Give it a name. Also, specify this newly created variable's name in the content property of the Message box activity: Hit the Run button or press F5 to see the result.
- We can see clearly that we have used two activities inside the Sequence that are logically related (one for inputting the name and the other for popping it up). Here, the Sequence contains two activities.

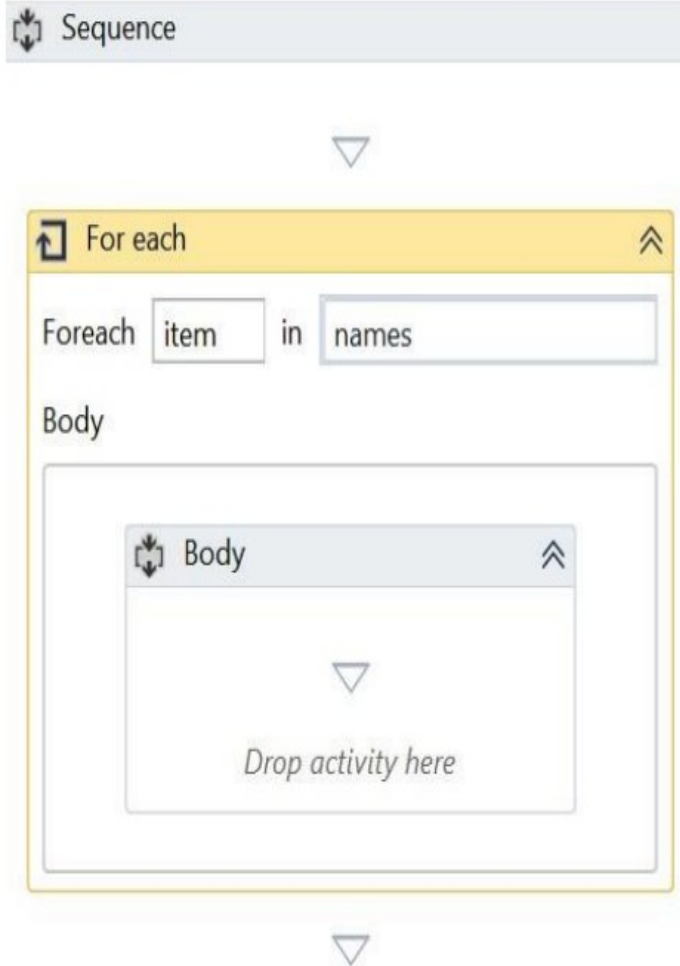


Steps to use a flowchart:

- Let us drag a
- inside the Fl
- Double click
- “Hello World
- to be quoted
- So, when the
- we can use
- Flowchart.
- However, it
- we have a la
- That is why
- related activ
- group the Se



Step-by-Step Example Using Sequence and Control Flow



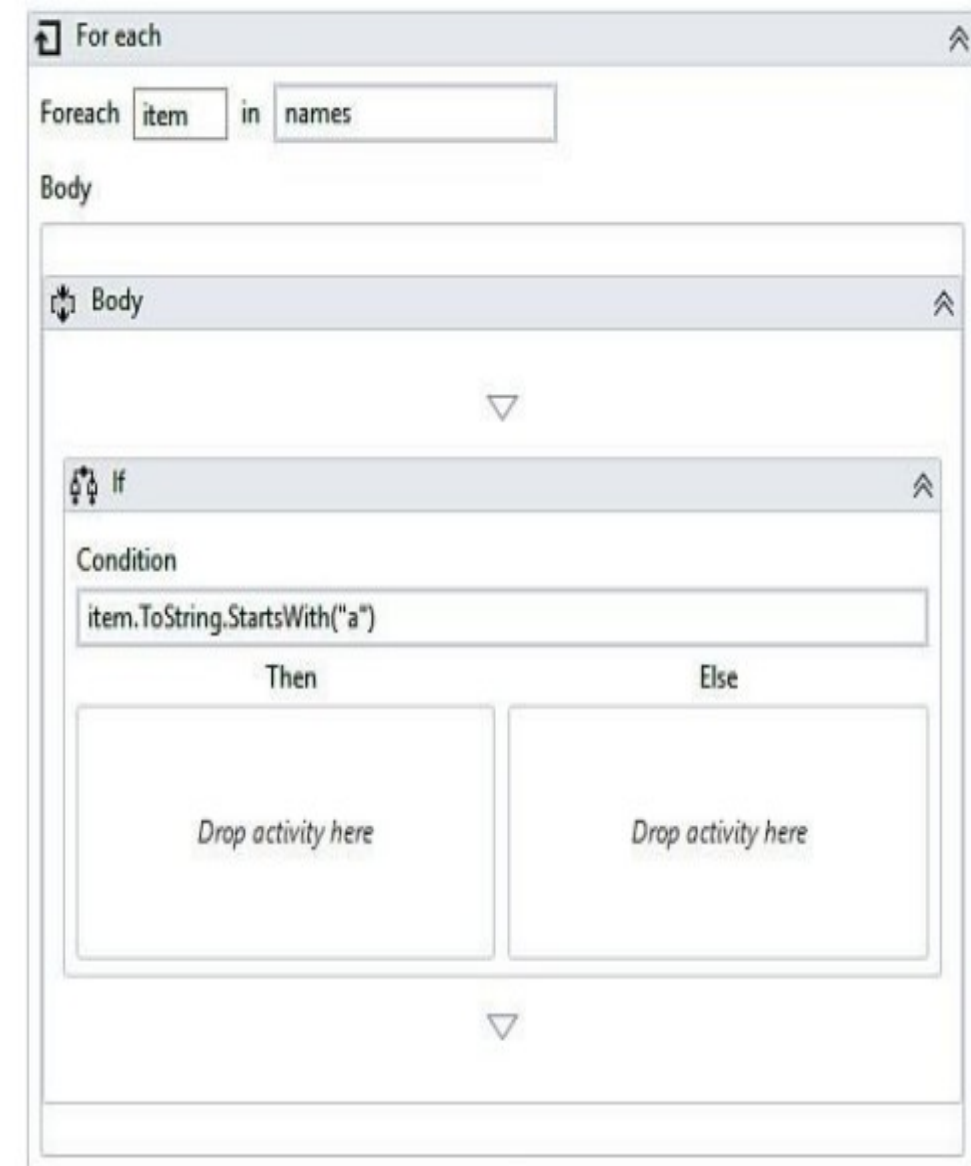
1

How to use sequence and workflow

Consider an array of names. Say we have to find out how many of them start with the letter a. 🕒
We will then create an automation where the number of names starting with a is counted and the result is displayed.

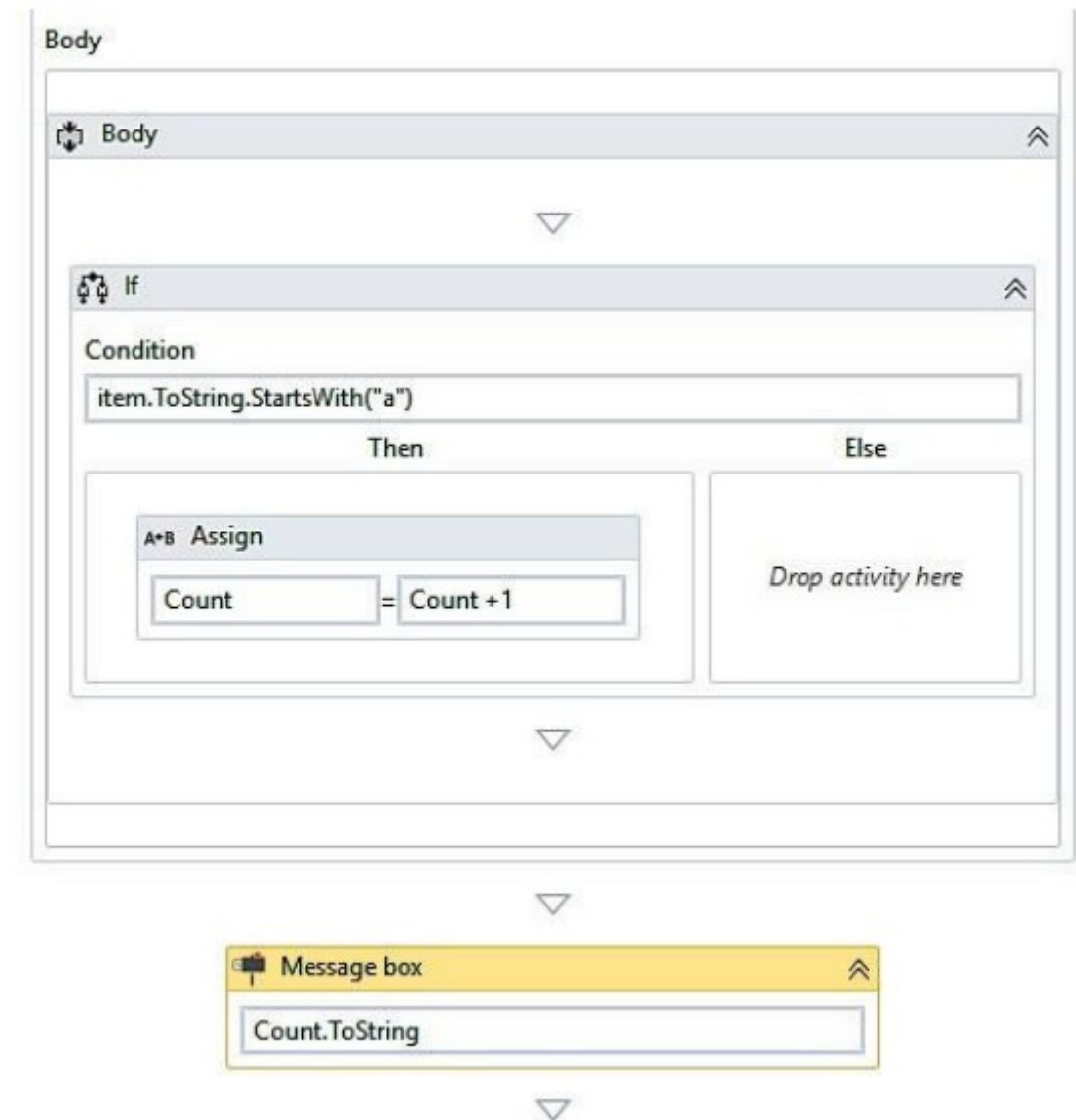
Steps to use sequence and control flow :

- Drag and drop a Flowchart activity from the Activities panel.
- Drag and drop a Sequence activity inside the Flowchart. Connect the Sequence to the Start node by right-clicking on the Sequence activity and selecting the Set as Start node option.
- Double click on the Sequence activity. Create a variable. Give it a name (in our case, we will create an array of type string and name the variable as name). Set the variable type to Array of [T]. When asked for the type of array, select String. Also, initialize the array in the Default section of the variable by giving it a default values. For example, {"john", "sam", "Andrew", "Anitha"}.
- 4. Create a variable of type integer Count for storing the result. Set the variable type to Int32:



Steps to use sequence and control flow :

- Drag and drop a For each activity inside the Sequence. Also, specify the array name in the expression box of the For each activity. The For each activity is used to iterate over the array. It will pick up one name from the array each time until it reaches the end:
- .Drag and drop the If activity from the Activities panel and place it inside the For each activity at the location where Drop activity here is mentioned. Specify the condition in the expression box of the If activity. The If activity is used to check for a particular condition/expression. ⌚ If that expression is satisfied, the Then block will be executed. Otherwise, the Else block will be executed. ⌚ We have specified the expression as `Item.ToString.StartsWith('a')`. This expression specifies the name present in the item variable starts with the letter 'a'. ⌚ The For each activity iterates over the array, picks up one name at a time, ⌚ and stores it as a variable, item:



Best Practices for Effective UiPath Automation

Identify Suitable Processes

Carefully evaluate which tasks and workflows are best suited for automation, considering factors like repeatability, rules-based nature, and time-saving potential.

Design for Scalability

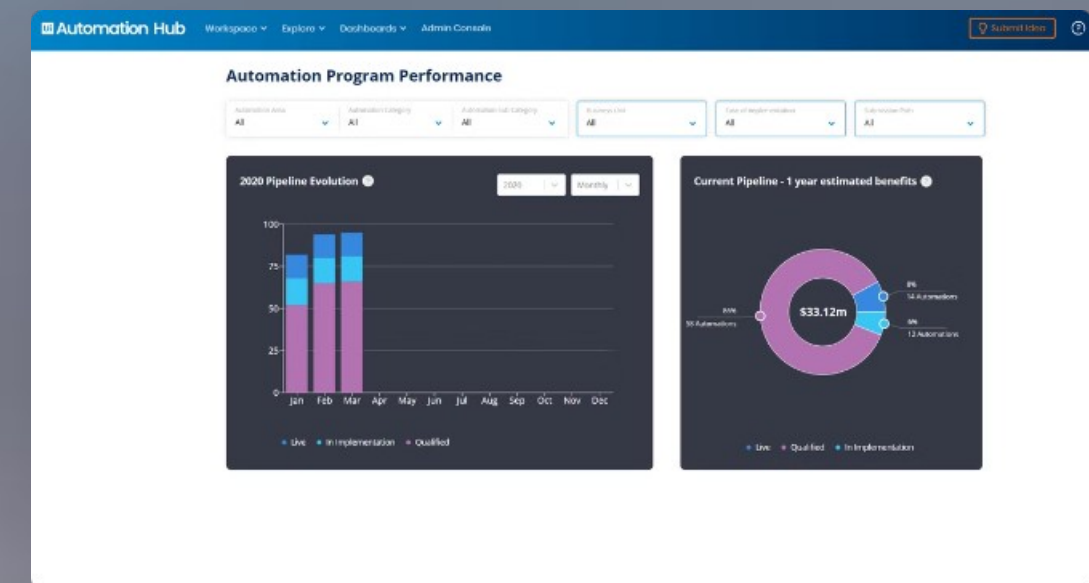
Build your automations with future growth and expansion in mind, using modularity, reusable components, and robust exception handling.

Implement Effective Monitoring

Incorporate logging, error reporting, and performance tracking to ensure your automations are running smoothly and provide valuable insights for optimization.

Foster Continuous Improvement

Regularly review and refine your automations, incorporating user feedback and leveraging new UiPath features to enhance efficiency and effectiveness.



Conclusion and Key Takeaways



Flexibility

UiPath offers multiple approaches to automation, each with its own strengths and trade-offs.



Best Practices

Adopting effective design, implementation, and maintenance strategies is crucial for successful UiPath automations.



Continuous Improvement

Regularly reviewing and refining your automations will help you maximize their efficiency and impact.

