|  |  |
| --- | --- |
|  |  |

# Control Flow

## 3.1 If Statement

3.1.1 Objective

Build a workflow using an If statement, which asks a user, whether the user will get the second Marshmallow or not.

* Ask the user, “Do you want to eat your first Marshmallow now or after 5 minutes?”
* If the user answers “Now”, respond with “Oops! You will not get the second

Marshmallow.”

* If the user answers “After 5 minutes”, respond with “Congrats! You will also get the second Marshmallow.”
* If the answer is other than “Now” or “After 5 minutes”, respond with “Invalid Input”.

3.1.2 Process Overview

* START
* Use an Input Dialog activity to ask the user “Do you want to eat your first

Marshmallow now or after 5 minutes?”  Store user response in a string variable.

* Use an If activity to check the user response o If the answer is “Now”, use a Message Box activity to display “Oops! You will not get the second Marshmallow.”

o If the answer is “After 5 minutes”, use a Message Box activity to display

“Congrats! You will also get the second Marshmallow.” o If the answer is other than “Now” or “After 5 minutes”, use a Message Box activity to display “Invalid Input”.

* STOP

5.1.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as “If Statement”.

|  |  |
| --- | --- |
| Step 3: | Drag a Sequence activity from the Activities panel and drop it in the Designer panel. |
| Step 4: | Name the Sequence activity as “Sequence – ‘Marshmallow Game’”. |
| Step 5: | Right-click on the Sequence activity container and select Annotations from the context menu. |
| Step 6: | Enter the annotation “This code is to ask the user whether he wants a second  Marshmallow.” |
| Step 7: | Insert an Input Dialog activity within the Sequence activity and name it as “Input Dialog – ‘Question’”. Enter the annotation “Question to User”. |
| Step 8: | In the Input Dialog activity, enter values as shown below: |

|  |  |
| --- | --- |
| Title | Label |
| “Question” | “Do you want to eat your first Marshmallow? Choose among the following options: ” + Environment.NewLine + “1. Now” +  Environment.NewLine + “2. After 5 minutes” |

Step 9: In the Variables panel, create a variable for the above Input Dialog activity as

shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Variable type | Scope | Default |
| UserInput | String | Sequence – ‘Marshmallow Game’ |  |

|  |  |
| --- | --- |
| Step 10: | Go to the Properties panel of the Input Dialog activity and insert UserInput in its Output property. |
| Step 11: | Insert an If activity below the Input Dialog activity and name it as “If - To check if the user input is ‘Now’”. Enter annotation: “This activity judges the User Input whether it is "Now", "After 5 minutes" or "Invalid"”. |
| Step 12: | In the condition input area of If activity, enter the expression: UserInput = “Now”. |
| Step 13: | Insert a Message Box activity in the Then section of the If activity and name it as “Message Box - Failed”. Enter annotation: “Prints Fail message”. |
| Step 14: | In the Message Box activity, enter the text “Oops! You will not get the second |

Marshmallow.”

|  |  |
| --- | --- |
| Step 15: | Insert a second If activity, name it as “If - To check User input is 'After 5 minutes'”, add an annotation “Check whether the input contains ‘After 5 minutes’ or Invalid input” within the Else section of the first If activity. |
| Step 16: | In the condition input area of second If activity, enter the expression:  UserInput = “After 5 minutes”. |
| Step 17: | Insert a Message Box activity in the Then section of the second If activity and name it as “Message Box - Success”. Add an annotation: “Prints Success message”. |
| Step 18: | In the Message Box activity, enter the text “Congrats! You will get the second  Marshmallow.” |
| Step 19: | Insert another Message Box activity in the Else section of the second If activity and name it as “Message Box – Invalid Input”. Add an annotation:  “Prints Invalid Input message”. |
| Step 20: | In the Message Box activity, enter the text “Invalid Input” |
| Step 21: | Save and run the workflow. |
|  |  |

## 3.2 Switch Activity

3.2.1 Objective

Build a workflow using Switch activity that asks users’ their eye color and display their personality in a message box.

* Ask the user for their eye color.
* If the user enters “Blue”, respond with “You must be very Brave!”
* If the user enters “Green”, respond with “You must be Generous!”
* If the user enters “Gray”, respond with “You must be very Wise!”
* If the user enters “Black”, respond with “You must be very Bold!”

3.2.2 Process Overview

* START
* Use an Input Dialog activity to get the eye color input of the user.
* Use a Switch activity to compare the input with four different cases – Blue, Green, Gray, and Black.
* Use Message Box activities to display the result of each case o For “Blue”, display “You must be Brave!” o For “Green”, display “You must be Generous!” o For “Gray”, display “You must be very Wise!” o For “Black”, display “You must be very Bold!”
* STOP

3.2.3 Step by Step Process

|  |  |
| --- | --- |
| Step 1: | Open UiPath Studio. |
| Step 2: | Create a new process and name it as “Switch Activity” |
| Step 3: | Drag a Sequence activity from the Activities panel and drop it in the Designer panel. |
| Step 4: | Name the Sequence activity as “Sequence – ‘Create a Robot that asks user their eye color’” |
| Step 5: | Right-click on the Sequence activity container and select Annotations from the |

context menu.

Step 6: Enter the annotation “This block of code is executed using Switch Activity.”

Step 7: Insert an Input Dialog activity, name it as “Input Dialog – ‘Question’”, Add an

annotation “Question to User” and enter the values as shown below:

|  |  |
| --- | --- |
| Title | Label |
| “Question” | “Enter the color of your eye:” |

Step 8: In the Variables panel, create a variable for the above Input Dialog activity as

shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Variable type | Scope | Default |
| EyeColor | String | Sequence – ‘Create a Robot that asks user their eye color’ |  |

|  |  |
| --- | --- |
| Step 9: | Go to the Properties panel of the Input Dialog activity and insert EyeColor in its Output property. |
| Step 10: | Drag and drop a Switch activity below the Input Dialog activity, name it as “Switch - Eye Color” and Add an annotation “Switch activity compares input with four different cases – Blue, Green, Gray, and Black.” |
| Step 11: | In the Properties panel of the Switch activity ensure that the TypeArgument property is String. |
| Step 12: | In the Expression text area of the Switch activity, enter the variable EyeColor. |
| Step 13: | In the Default section of the Switch activity, insert a Message Box activity and name it as “Message Box - Default”. Add an annotation: “Prints default message” |
| Step 14: | In the text area of the Message Box activity, enter the text “Can’t recognize that color!” |
| Step 15: | Add a new case in the Switch activity by clicking ‘Add new case’ option available. Enter “Blue” in the text area of ‘Case value’. |
| Step 16: | In the ‘Case Blue’ section, add a Message Box activity, and name it as “Message Box - Case Blue”. Add an annotation “Prints Case Blue message” and in the text area enter “You must be very Brave!” |
| Step 17: | Add a new case and enter “Green” in the text area of the ‘Case value’. |

|  |  |
| --- | --- |
| Step 18: | In the ‘Case Green’ section, add a Message Box activity, name it as “Message Box - Case Green”. Add an annotation “Prints Case Green message” and in the text area enter “You must be very Generous!” |
| Step 19: | Add a new case and enter “Gray” in the text area of the ‘Case value’. |
| Step 20: | In the ‘Case Gray’ section, add a Message Box activity, name it as “Message Box - Case Gray”. Add an annotation “Prints Case Gray message” and in the text area enter “You must be very Wise!” |
| Step 21: | Add a new case and enter “Black” in the text area of the ‘Case value’. |
| Step 22: | In the ‘Case Black’ section, add a Message Box activity, and name it as  “Message Box - Case Black”. Add an annotation “Prints Case Black message”. |
| Step 23: | In the text area, enter “You must be very Bold!” |
| Step 24: | Save and Run the project. |
|  |  |

**4.1 Do While Loop**

4.1.1 Objective

Build a workflow for a 'Guessing Game' with the following conditions:

* Generate a random number and prompt the user to input a number.
* In case of a wrong input, a message is displayed to the user stating, 'Please enter a lesser/greater number'.
* The loop keeps on running until the input number equals to the generated number.

4.1.2 Process Overview

* START
* Use an Input Dialog activity within a Do While activity to get the guessed number from the user.
* For Do While activity, set the condition to check guessed number is not equal to the actual number.
* Use a Message Box activity to display “You Guessed it correct” for the correct match.
* Use an If activity within the Do While loop to check if the guessed number is equal to the actual number.
  + If correct, use a Message Box activity to display “You Guessed it correct” for the correct match.
  + Use another If activity within the Else section to check if the guessed number is greater than the actual number.
    - * If correct, use a Message Box activity to display “Please try a smaller number”.
      * If incorrect, use a Message Box activity to display “Please try a greater number”.
* STOP

4.1.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as “Do While Loop”.

|  |  |
| --- | --- |
| Step 3: | Drag a Sequence activity from the Activities panel and drop it in the Designer panel. |
| Step 4: | Name the Sequence activity as “Sequence – ‘Guessing Game”. |
| Step 5: | Right-click on the Sequence activity container and select Annotations from the context menu. |
| Step 6: | Enter the annotation :  “This block of code demonstrates a workflow using a Do While statement for creating a 'Guessing Game' with the following conditions:   1. Generate a random number and prompt the user to input a number. 2. In case of a wrong input, a message is displayed to the user stating, 'Please enter a lesser/greater number’. 3. The loop keeps on running until the input number equals the generated number.” |
| Step 7: | Create variables using Variables panel as shown below: |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Variable type | Scope | Default |
| intRandomNo | Int32 | Sequence – Guessing Game | 25 |
| intGuessedNo | Int32 | Sequence – Guessing Game |  |

|  |  |
| --- | --- |
| Step 8: | Insert a Do While activity within the Sequence activity, name it as “Do While - Guessed Number <> Random Number”, add an annotation “The loop iterates until it reaches the given condition”. |
| Step 9: | Set its condition to int intGuessedNo<> intRandomNo |
| Step 10: | Insert an Input Dialog activity within the Do While activity, name it as “Input |

Dialog - Guessed Number”, add an annotation: “Take Guessed Number as User input” and enter values as shown below:

|  |  |
| --- | --- |
| Title | Label |
| “Number” | “Guess a number” |

Step 11: In the Properties panel of the Input Dialog activity, enter intGuessedNo in the Output property.

|  |  |
| --- | --- |
| Step 12: | Insert an If activity below the Input Dialog activity, name it as “If - User input equals Random Number”, add an annotation: “This activity checks whether the User input is equal to the Random Number or not” and enter the condition  intGuessedNo= intRandomNo |
| Step 13: | In the Then section, insert a Message Box activity and name it as “Message Box - Correct Guess”. Add an annotation: “Prints Correct Guess message”. |
| Step 14: | Enter the text “You Guessed it correct”. |
| Step 15: | Insert another If activity, in the Else section of the first If activity, and enter condition intGuessedNo> intRandomNo. Name it as “If- Guessed number is greater or smaller than Random Number”, add an annotation: “This activity checks whether the user input is greater or smaller than the Random number.” |
| Step 16: | In the Then section, insert a Message Box activity, name it as “Message Box - Try Smaller Number”, add an annotation: “Prints Smaller Number message” and enter the text “Please try a smaller number”. |
| Step 17: | In the Else section, insert a Message Box activity, name it as “Message Box - Try Greater Number”, add an annotation: “Prints Greater Number message” and enter the text “Please try a greater number”. |
| Step 18: | Save and run the workflow. |
|  |  |

## 4.2 While Loop

4.2.1 Objective

Build a workflow using a While loop that tells the user if the input is a prime number or not.

* Ask the user to input a number.
* Check if it is a prime number.
* If the input number is prime, then display "It is a prime number" in a message box.
* If the input number is not prime, then display "It is not a prime number" in a message box.

4.2.2 Process Overview

* START
* Use an Input Dialog activity and ask for any number from the user and store in a variable called intNumber.
* Create two more variables intRandom and c with Variable Type as Int32 and Default value as 2 and 0 respectively in the variables panel.
* Use a While activity and set the condition to intRandom<Number.
* Use an If activity within the While activity and set the condition to intNumber mod intRandom=0.
* Use an Assign activity within the Then section and increment value of intCount by

1.

* Use an Assign activity after/below the If activity, and increment value of intRandom by 1.
* Use another If activity after/below the While activity and enter condition intCount>0.
* Use a Message Box activity within the Then section to display “It is not a prime number”.
* Use a Message Box activity within the Else section to display “It is a prime number”.
* STOP

4.2.3 Step by Step Process

Step 1: Open UiPath Studio.

|  |  |
| --- | --- |
| Step 2: | Create a new process and name it as “While Activity”. |
| Step 3: | Drag a Sequence activity from the Activities panel and drop it in the Designer panel. |
| Step 4: | Name the Sequence activity as “Sequence – ‘This is the code to test whether the input is a prime number or not.’” |
| Step 5: | Right-click on the Sequence activity container and select Annotations from the context menu. |
| Step 6: | Enter the annotation : “This block of code demonstrates a workflow using  While loop that tells the user if the input is a prime number or not.” |
| Step 7: | Insert an Input Dialog activity within the Sequence activity, name it as “Input Dialog – ‘To take the input from the user’” and add an annotation “Take User input as a Number”. |
| Step 8: | In the Input Dialog activity, enter values as shown below: |

|  |  |
| --- | --- |
| Title | Label |
| “Number” | “Enter a number” |

Step 9: In the Variables panel, create three variables as shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Variable type | Scope | Default |
| intNumber | Int32 | Sequence – ‘This is the code to test whether the input is a prime number or not.’ |  |
| intRandom | Int32 | Sequence – ‘This is the code to test whether the input is a prime number or not.’ | 2 |
| intCount | Int32 | Sequence – ‘This is the code to test whether the input is a prime number or not.’ | 0 |

Step 10: Go to the Properties panel of the Input Dialog activity and insert intNumber

in its Output property.

Step 11: Insert a While activity below the Input Dialog activity and name it as “While – ‘To check if the number is a prime number or not’”.

|  |  |
| --- | --- |
| Step 12: | Right-click on the While activity container, and select Annotations from the context menu. |
| Step 13: | Add an annotation “This block of code will check whether the number is prime. If it is, it will increment the value of ' intCount’.” |
| Step 14: | Inside the While activity, enter the condition as intRandom < intNumber |
| Step 15: | In the Body section of the While activity drag and drop a Sequence activity. |
| Step 16: | Rename the Sequence activity to “Sequence – ‘Check the number using ‘If’”. |
| Step 17: | Right-click on the Sequence activity container and select Annotations from the context menu. |
| Step 18: | Add an annotation “In this sequence using 'If' activity, the 'Number' is divided by ' intRandom ' until intRandom = intNumber.” |
| Step 19: | Insert an If activity inside the Sequence activity. |
| Step 20: | Inside the If activity, enter the condition as intNumber Mod intRandom = 0. |
| Step 21: | Inside the Then section of the If activity, insert an Assign activity, and enter |

values as shown below:

|  |  |
| --- | --- |
| To | Value |
| intCount | intCount + 1 |

|  |  |
| --- | --- |
| Step 22: | Change the Assign activity name to “Assign – ‘Increment the value of intCount’”. |
| Step 23: | Right-click on the Assign activity container and select Annotations from the context menu. |
| Step 24: | Add an annotation “Incrementing the value of ‘intCount’ when ‘intNumber is found to be a prime number.” |
| Step 25: | Below the If activity, insert another Assign activity and rename it to “Assign- Incrementing the value of ‘intRandom’. |
| Step 26: | In the Assign activity, enter the values as shown below: |

|  |  |
| --- | --- |
| To | Value |
| intRandom | intRandom +1 |

|  |  |
| --- | --- |
| Step 27: | Right-click on the Assign activity container, and select Annotations from the context menu. |
| Step 28: | Add an annotations "Incrementing the value of ' intRandom ' whenever the loop iterates". |
| Step 29: | Below the While activity, insert an If activity and name it as “If – Print the message”. |
| Step 30: | Right-click on the If activity container, and select Annotations from the context menu. |
| Step 31: | Add an annotation “This block of code will print the message in a message box whether the input is Prime or not.” |
| Step 32: | Inside the If activity, enter the condition intCount >0. |
| Step 33: | In the Then section, insert a Message Box activity and name it as “Message Box - Not a prime number”. Add an annotation “Displays that the number is not a prime.” |
| Step 34: | Enter the text “It is not a prime number.” |
| Step 35: | In the Else section, insert another Message Box activity and name it as “Message Box - Is a prime number”. Add an annotation “Displays that the number is not a prime.” |
| Step 36: | Enter the text “It is a prime number.” |
| Step 37: | Save and run the workflow. |
|  |  |

## 4.3 For Each Loop

4.3.1 Objective

Build a workflow to display file names from a folder in the Output panel and also store names in an MS Word file.

* Locate and select a folder containing multiple files.
* List the directory path of all the files in the Output panel.
* Also, store the updated names in an MS Word file and save and close it.

4.3.2 Process Overview

* START
* Use a Select Folder activity to select a folder containing a few files.
* Use an Assign activity to store file names in an array.
* Use an Attach Window activity below the Assign activity and select MS Word window.
* Use a For Each activity to iterate through each file name in the array.
* Use a Write Line activity within the For Each activity to display file names in the Output panel.
* Use a Type Into activity below the Write Line activity to store file names in an MS Word file.
* Use Click and Send Hotkey activities to save and close the file.
* STOP

4.3.3 Step by Step Process

|  |  |
| --- | --- |
| Step 1: | Open a new MS Word file. |
| Step 2: | Open UiPath Studio. |
| Step 3: | Create a new process and name it as “For Each Activity” |
| Step 4: | Drag a Sequence activity from the Activities panel and drop it in the Designer panel. |
| Step 5: | Name the Sequence activity as “Sequence – ‘'This code is to write all the file |

names in MS Word present in a particular folder.’”

|  |  |
| --- | --- |
| Step 6: | Right-click on the Sequence activity container and select Annotations from the context menu. |
| Step 7: | Enter the annotation shown below:  “1. Locate and select a folder containing multiple files.   1. List the directory path of all the files in the Output panel. 2. Store the updated names in MS Word file and save and close it.” |
| Step 8: | Insert a Select Folder activity in the Designer panel and add an annotation:  “User selects a folder”. |
| Step 9: | In the Variables panel, define a new variable as shown below: |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Variable type | Scope | Default |
| FolderName | String | Sequence - ‘This code is to write all the file names in MS Word present in a particular  folder’ |  |

|  |  |
| --- | --- |
| Step 10: | In the Properties panel of the Select Folder activity, enter FolderName in the Output property. |
| Step 11: | Insert an Assign activity below the Select Folder activity and name it as “Assign - 'File List'”. Add an annotation: “Get all the file names that the user selects and stores it in a list”. |
| Step 12: | In the Variables panel, define a new variable as shown below: |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Variable type | Scope | Default |
| FileList | System.String[] | Sequence - ‘This code is to write all the file names in MS Word present in a particular  folder’ |  |

Step 13: In the Properties panel of the Assign activity, enter FileList in the Output

property.

Step 14: In the Assign activity, enter the value as shown below:

|  |  |
| --- | --- |
| To | Value |
| FileList | Directory.GetFiles.(FolderName) |

|  |  |
| --- | --- |
| Step 15: | Drag and Drop an Attach window activity and name it as “Attach Window – MS Word”. Add an annotation: “Attach MS Word window”. |
| Step 16: | Click on the “Indicate Element on Screen” link and select the MS Word window opened in the background. |
| Step 17: | Click the hamburger button and select Edit Selector. In the bottom panel of the Selector Editor, rename the title of the MS Word to ‘\*’. Click OK to save the changes. |
| Step 18: | Drag and drop a For Each activity in the Do section of the Attach Window activity, insert item in the first text box, and the variable FileList in the second text box. Add an annotation: “Loop iterates for each item in the list”. |
| Step 19: | Drag and Drop a Write Line activity in the Body section of the For Each activity. Enter the expression: item.ToString. Add an annotation: “The result will appear in the Output Panel”. |
| Step 20: | Drag and Drop a Type Into activity in the Body section of the For Each activity and name it as “Type Into - 'File List into MS Word'”. Add an annotation: “Each item in the list results to be printed in the MS Word file.” |
| Step 21: | Click on the “Indicate element on screen” link and select the editor area of MS Word. |
| Step 22: | In the text area of Type Into activity, enter the expression: item.ToString +  “[k(enter)]” |
| Step 23: | Drag a Send Hotkey activity and drop it below the For Each activity. |
| Step 24: | Name the “Send Hotkey” activity as “Send Hotkey - ' Open the Save As dialog box.’”. Add an annotation: “Pressing F12 key opens the 'Save As' Dialog box in the MS Word”. |
| Step 25: | Click on the “Indicate element on screen” link and select the editor area of MS Word. |
| Step 26: | In the Send Hotkey activity, select F12 from the dropdown for Key option. |
| Step 27: | Insert a Click activity. Click “Indicate element inside window” link and select |

the Save button of the ‘Save As’ dialog box.

|  |  |
| --- | --- |
| Step 28: | Name the Click activity as “Click ‘Save’ button”. Add an annotation: “Press  Save button from Save As pop Dialog box.” |
| Step 29: | Insert second Send Hotkey activity and name it “Send Hotkey” activity as Send Hotkey - 'To close the application’” Add an annotation: “Press 'Alt+F4' key to close the MS Word window.” |
| Step 30: | Click on the “Indicate element on screen” link and select the editor area of MS Word. |
| Step 31: | In the Send Hotkey activity, check the box below Alt option, and choose F4 from the dropdown of Key option. |
| Step 32: | Save and run the workflow. |
|  |  |