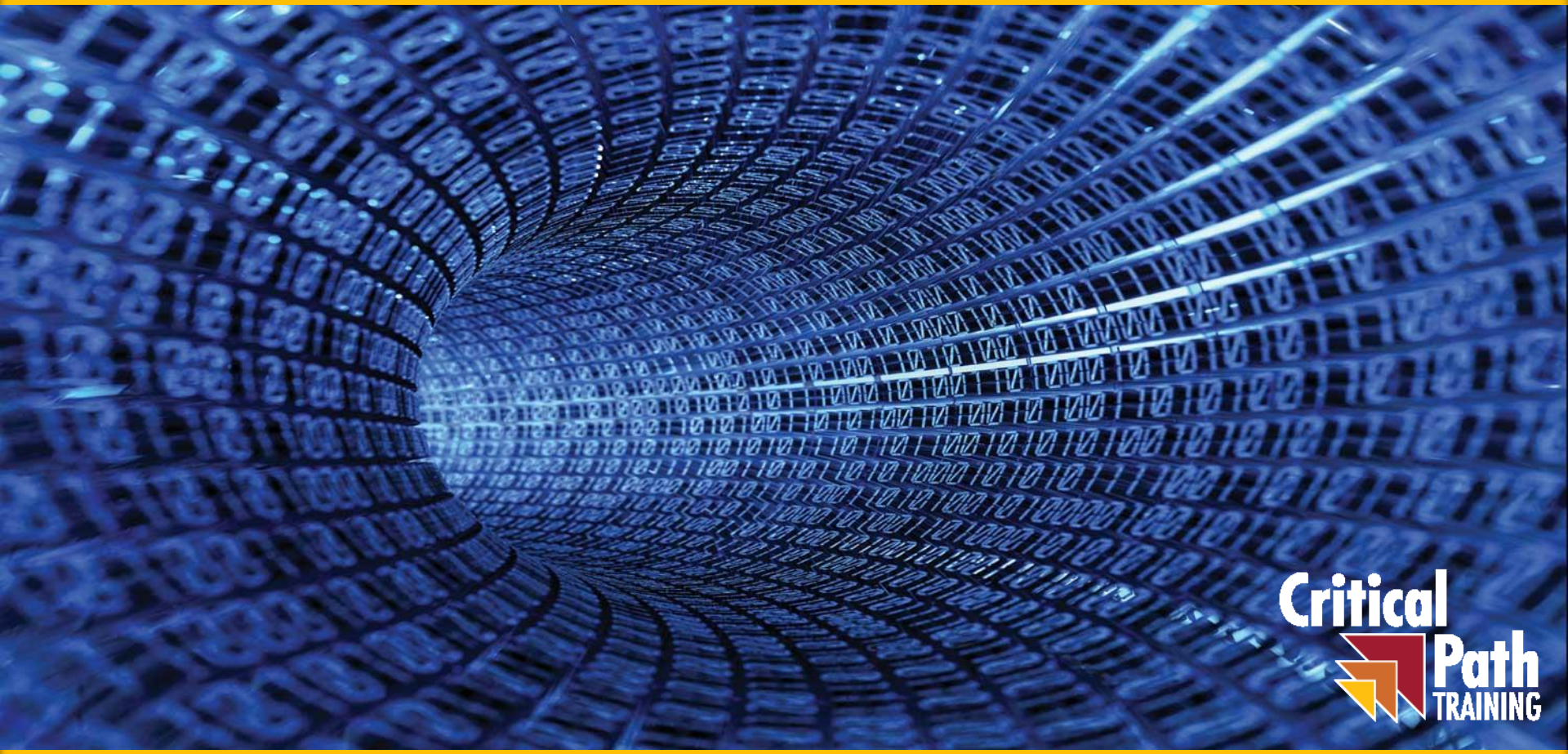
# Getting Started with Power Automate



# Agenda

- Power Automate Fundamentals
- Creating and Testing Flows
- Using Control-of-Flow Actions
- Writing Flow Expressions
- Automating Document Generation

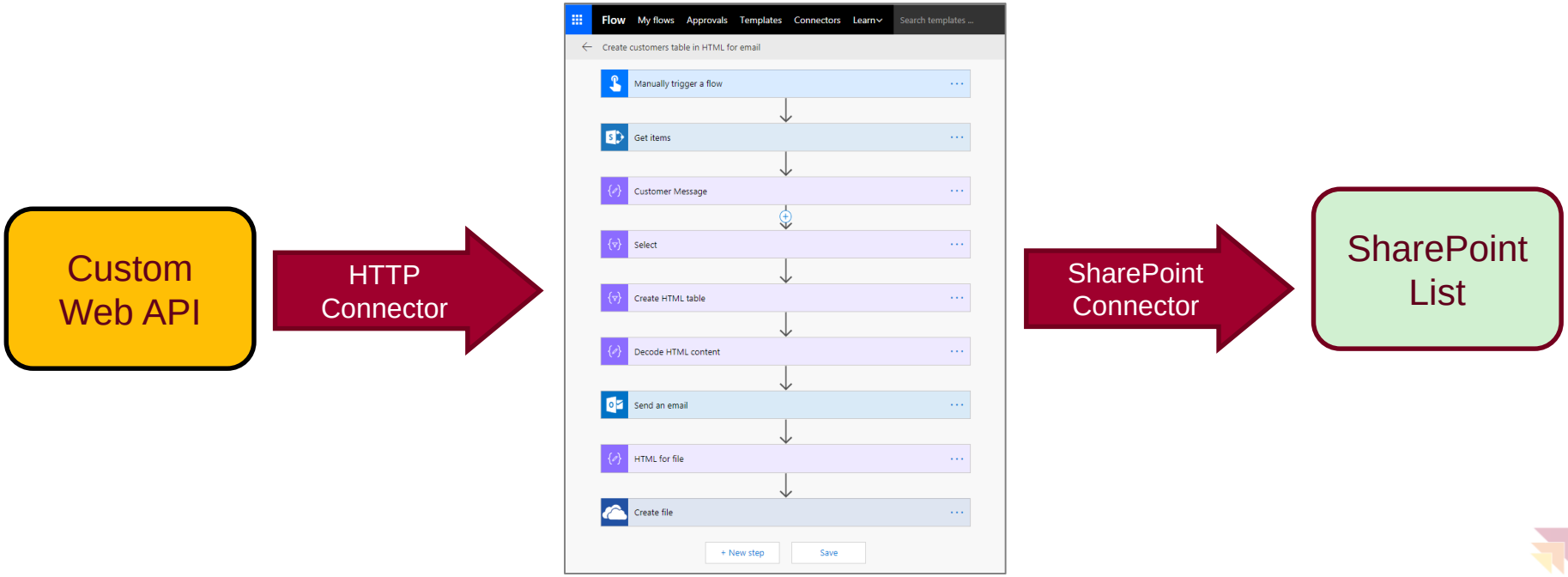
# What is Power Automate?

- Service for automating business and workflow processes
  - You use Power Automate by creating and running "flows"
  - Power Automate provides browser-based flow editing experience
  - Power Automate originally went by the name of "Microsoft Flow"

- What can you automate by creating a flow?
  - Sending notifications
  - Importing and exporting data
  - Generating Microsoft Word documents
  - Automating approvals

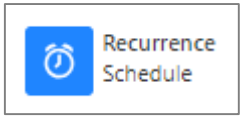
# Building Blocks of Power Automate

- Flows are created using the following building blocks
  - **Triggers** - events that start a flow
  - **Actions** - tasks and operation executed by flow
  - **Services** - sources and destinations for data
  - **Connectors** - wrappers to communicate with service APIs

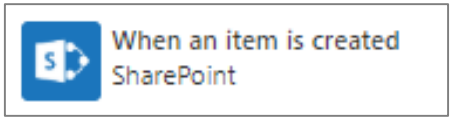


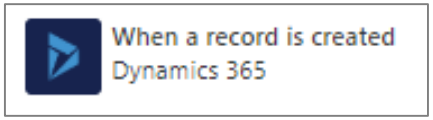
# Flow Trigger Types

- Scheduled Flow Triggers
  - Runs periodically based on an interval



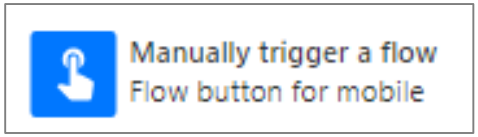

- Automated Flow Triggers
  - Runs when something happens

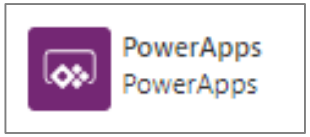




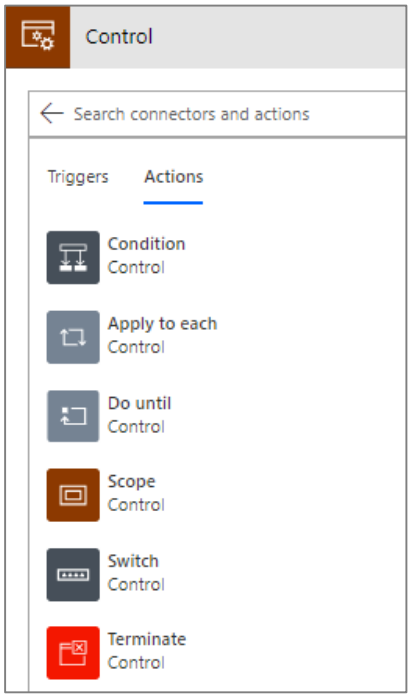
- On-demand Flow Triggers
  - Runs when a user clicks a button

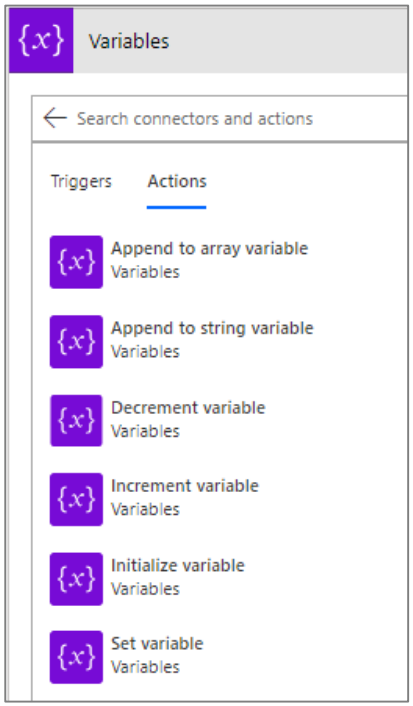


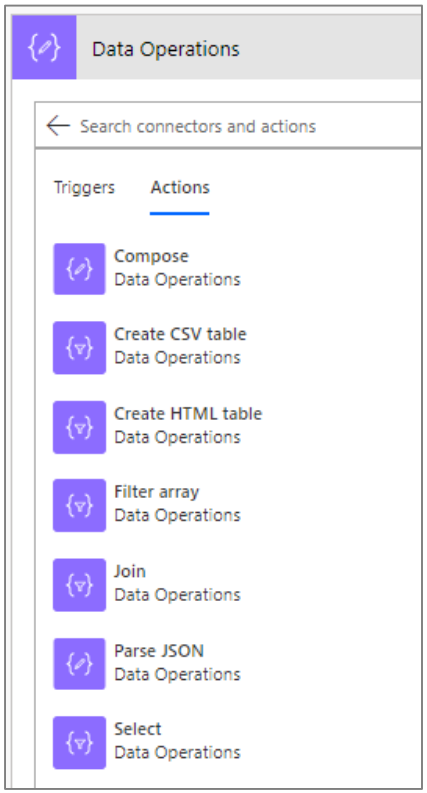



# Core Action Categories

- **Control**: actions to provide control-of-flow
- **Variables**: actions to manage state within flow lifetime
- **Data operations**: action to process data & prepare content

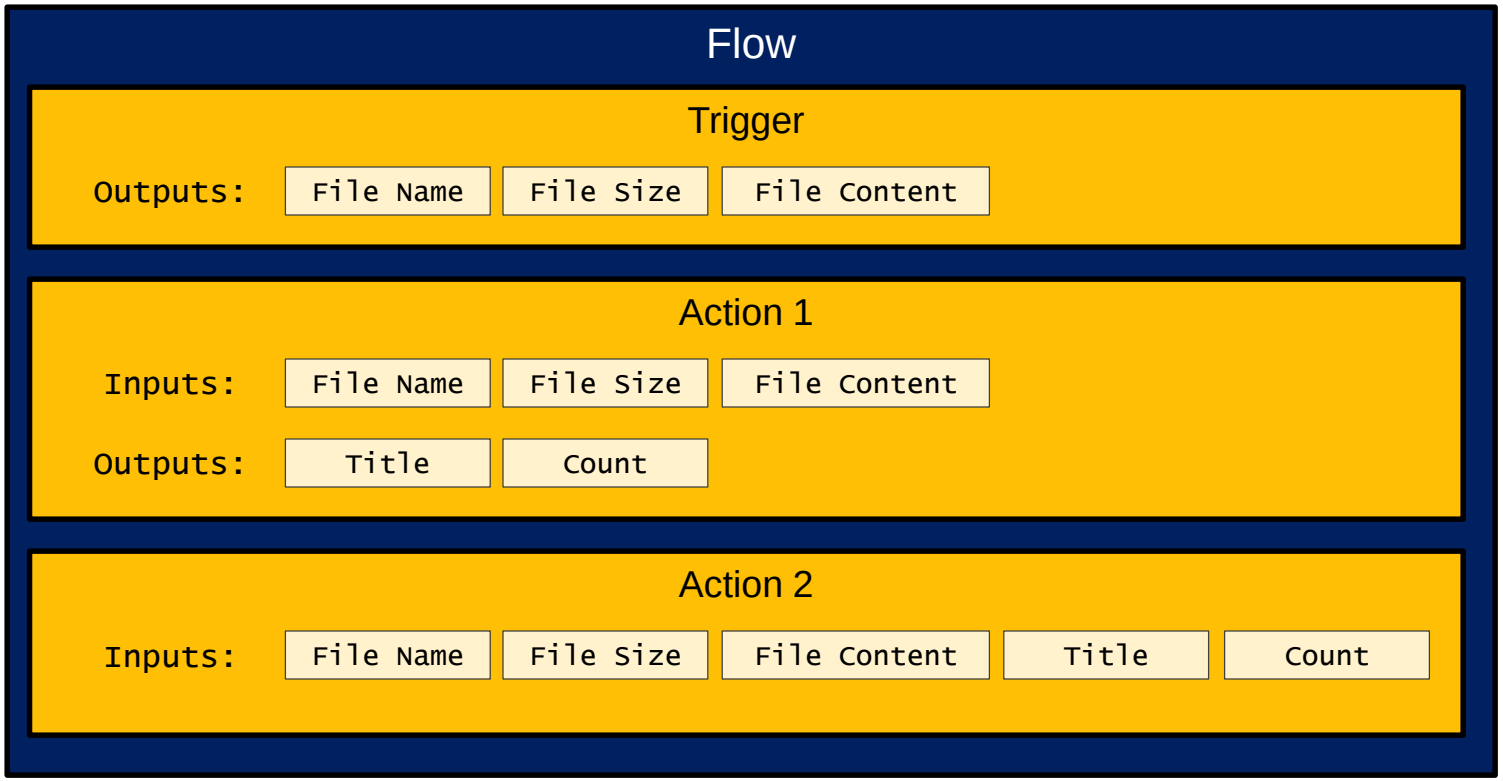


# Data Automatically Flows from Step to Step

- Data in flows added by step outputs
  - Data added in step output is available in later steps
  - It's easy to configure step input data using output data in previous steps
  - Certain outputs displayed/hidden based on types of input and output

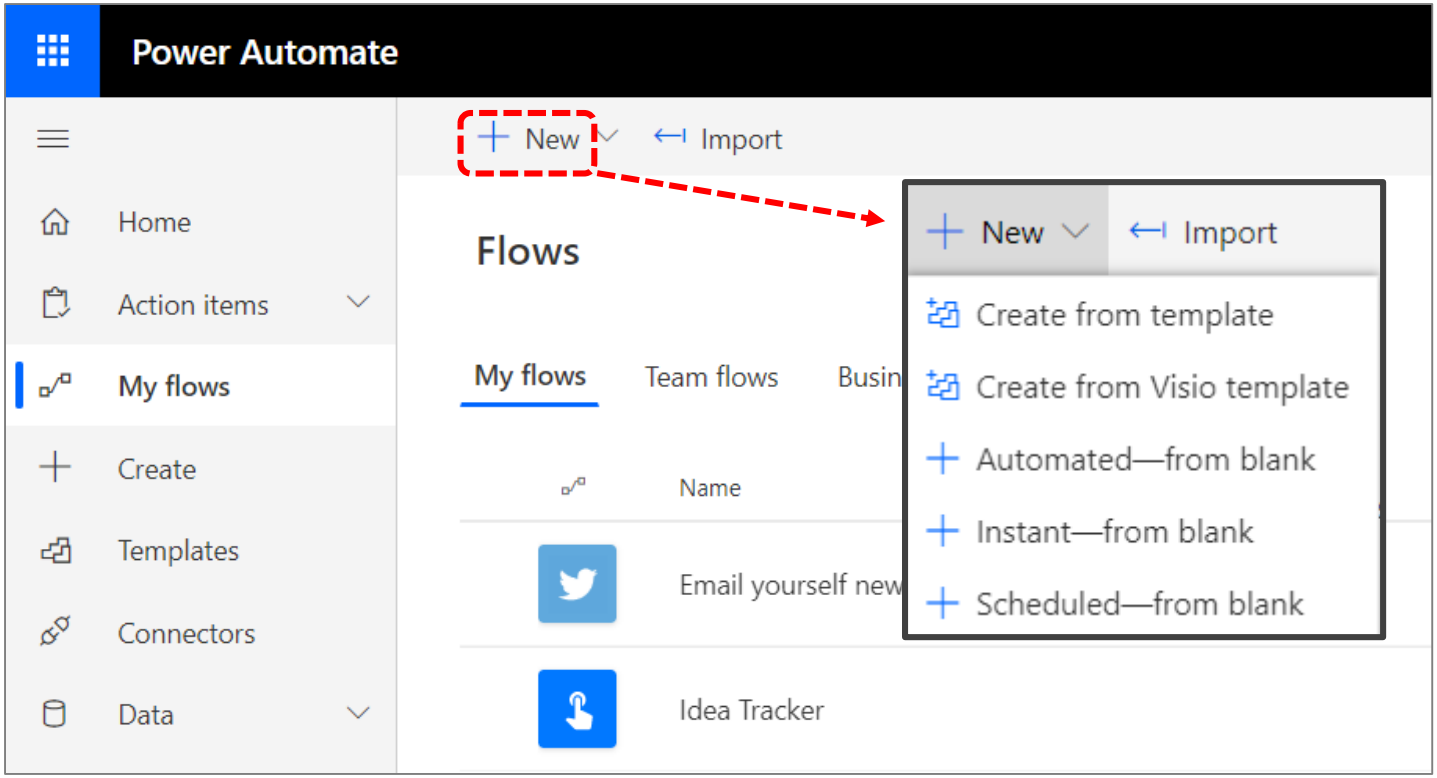


# Agenda

- ✓ Power Automate Fundamentals
- ➢ Creating and Testing Flows
- Using Control-of-Flow Actions
- Writing Flow Expressions
- Automating Document Generation

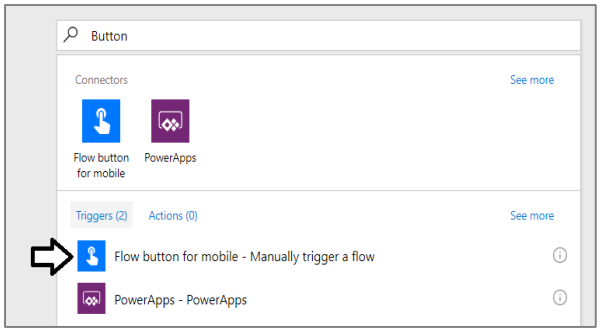
# Creating and Managing Flows

- Power Automate user portal used to manage & edit flows
  - Accessible through https://flow.microsoft.com
  - Flow can be created from blank or from template

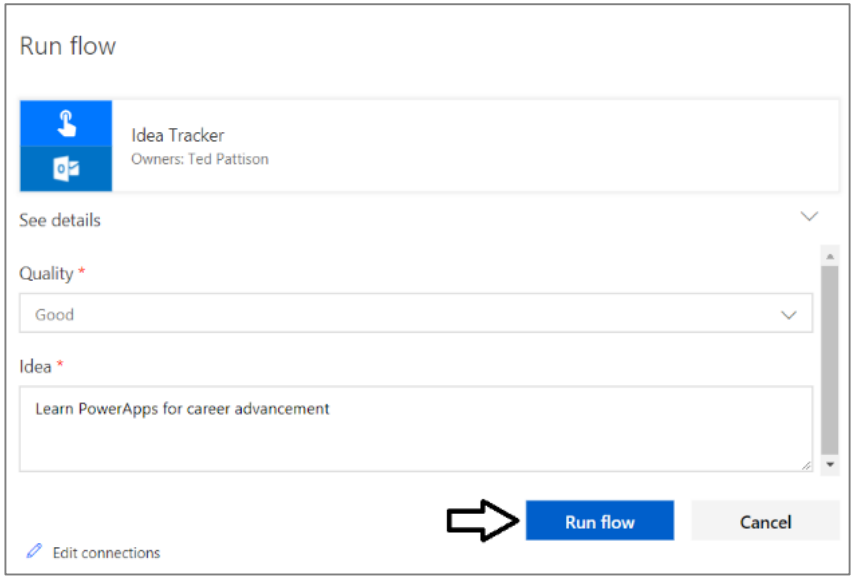


# Creating a Flow Trigger by Manual Button

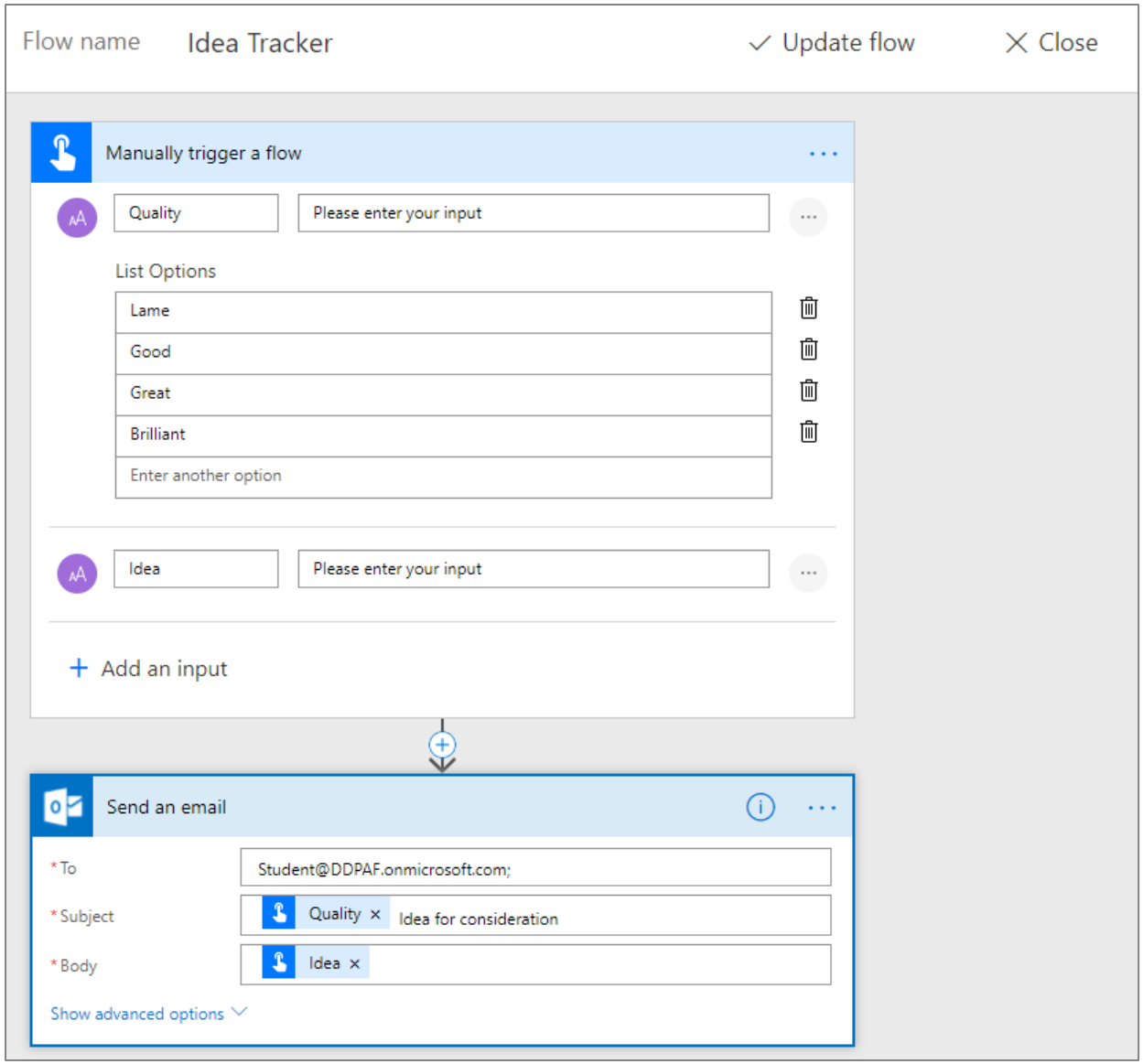
- Use Run button for Mobile as flow trigger




- Trigger can be extended with one or more input fields

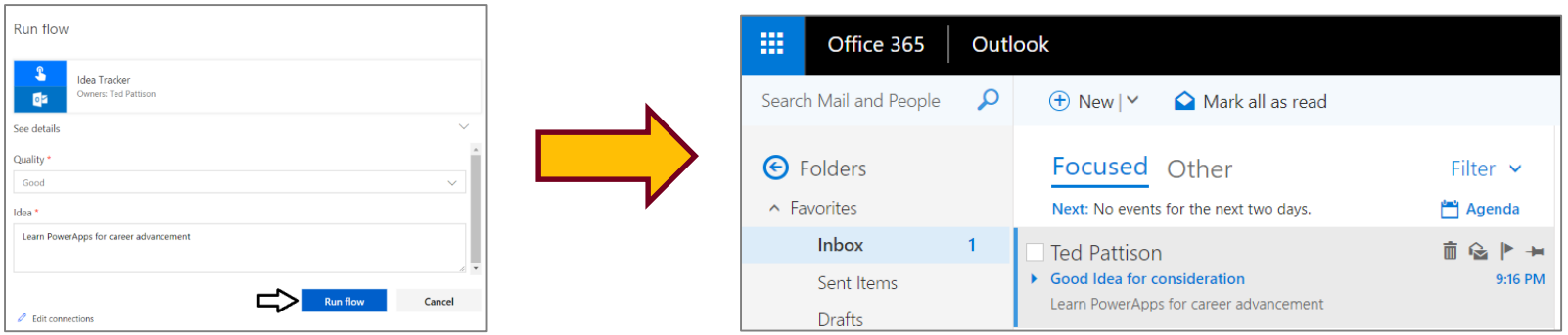


# Building Out The Idea Tracker Flow

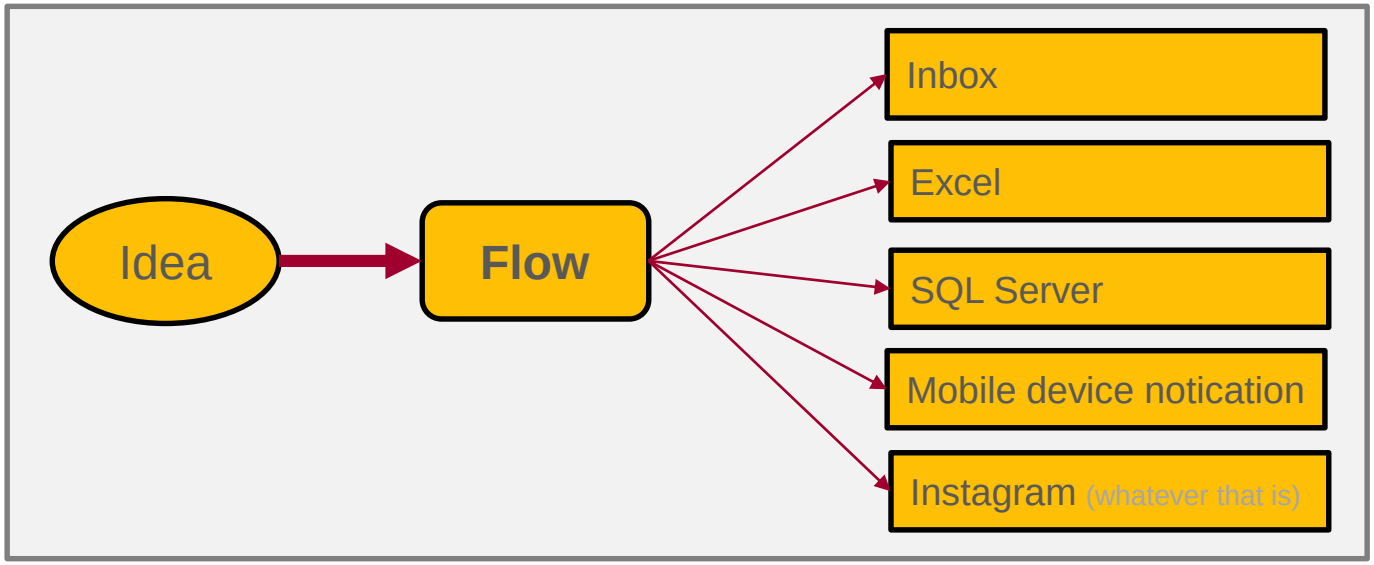


# Where Should You Write the Idea?

- You can track the idea by sending an email

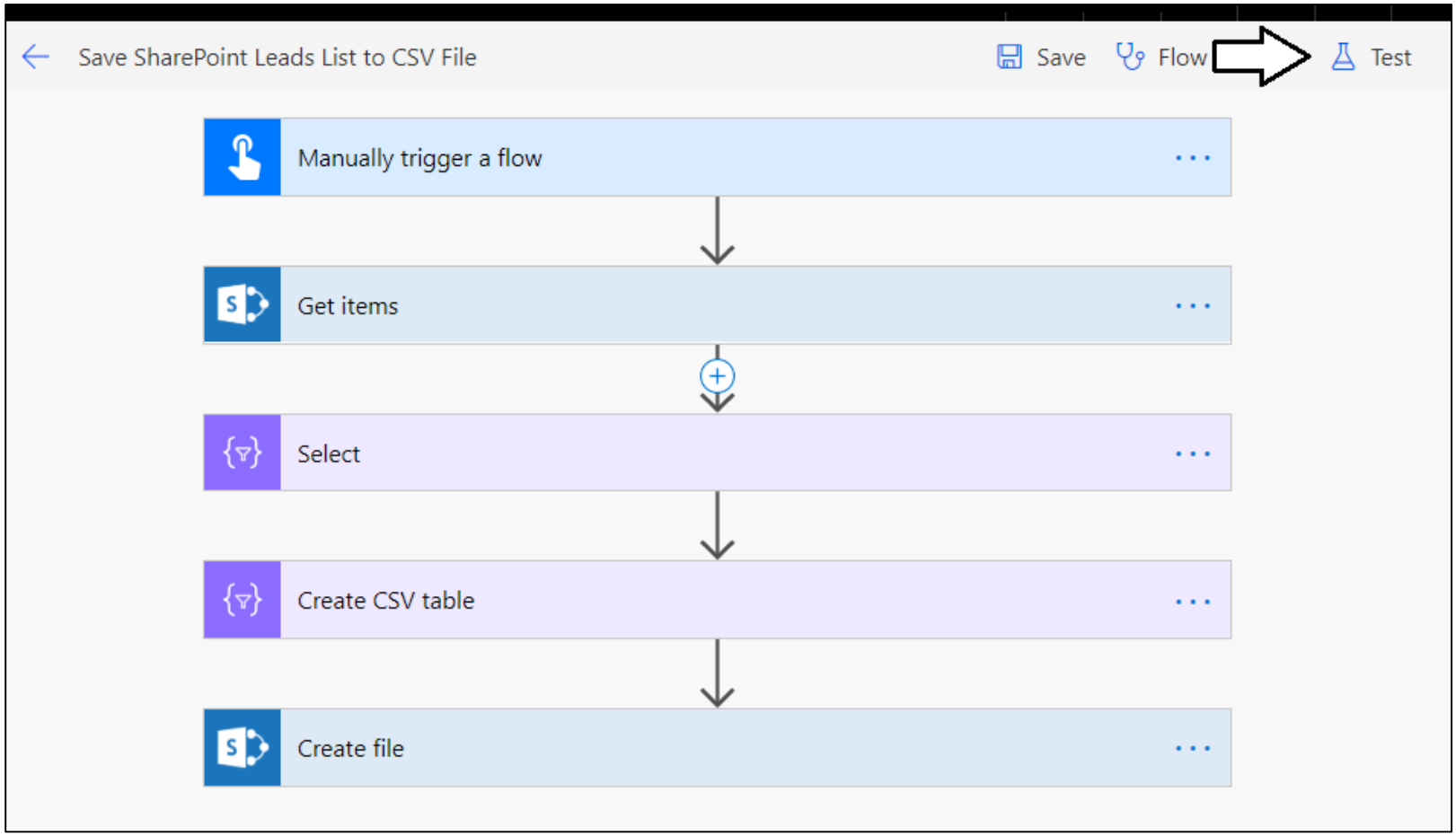



- Or get even more extravagant



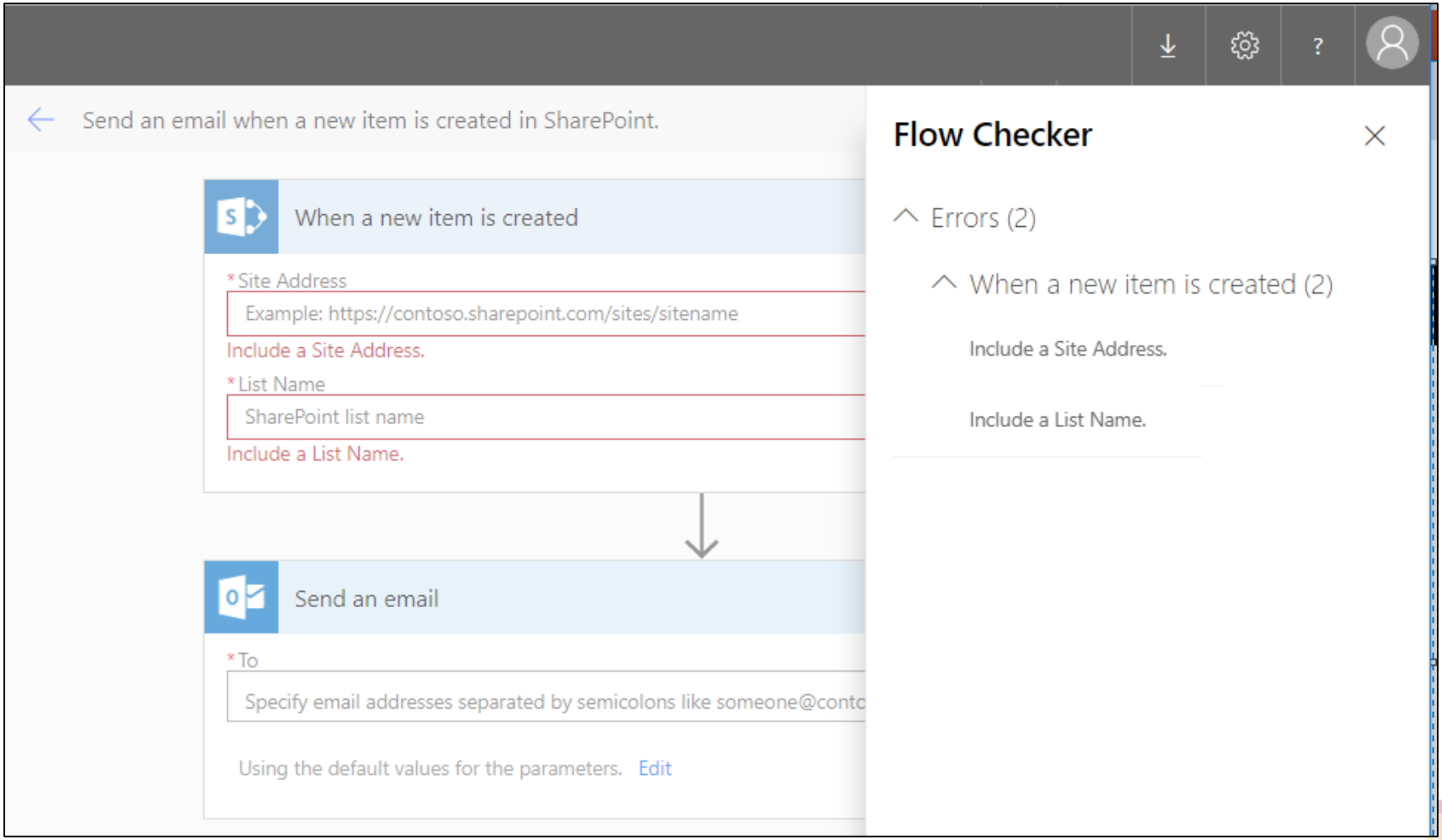
# Testing a Flow

- Flow can be run/tested from edit mode



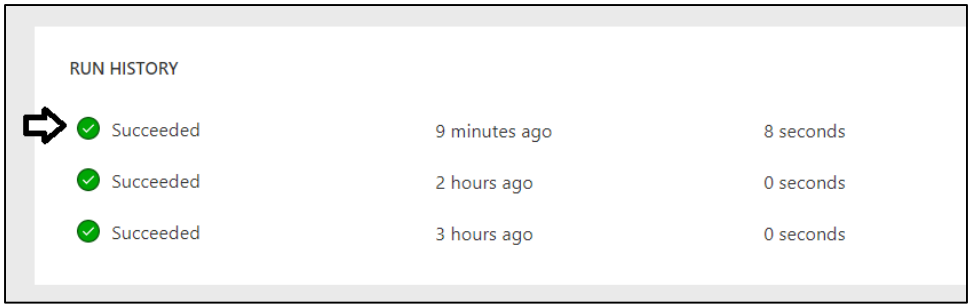
# Flow Checker

- Automatically checks flows for errors and omissions

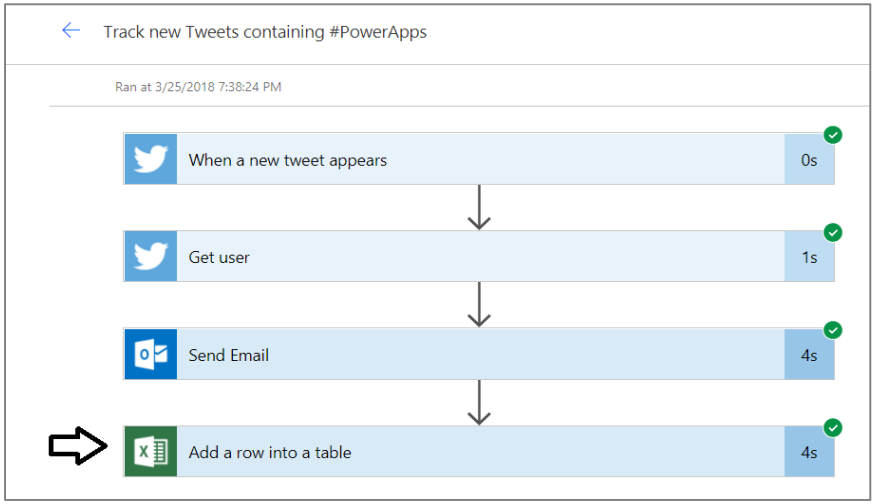


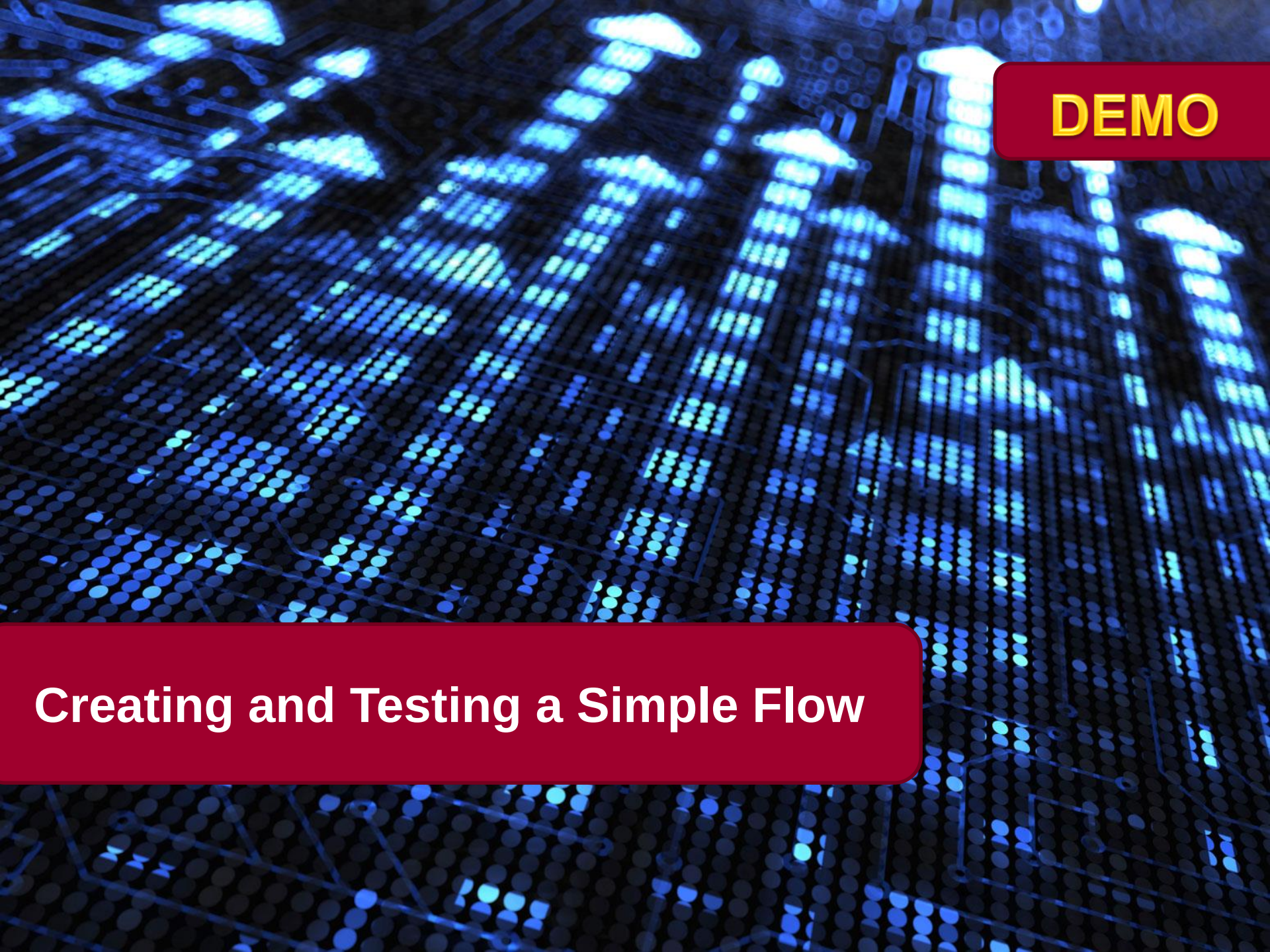
# Run History

- Flow provides history flows that have run




- Provides read-only view of data for auditing & monitoring



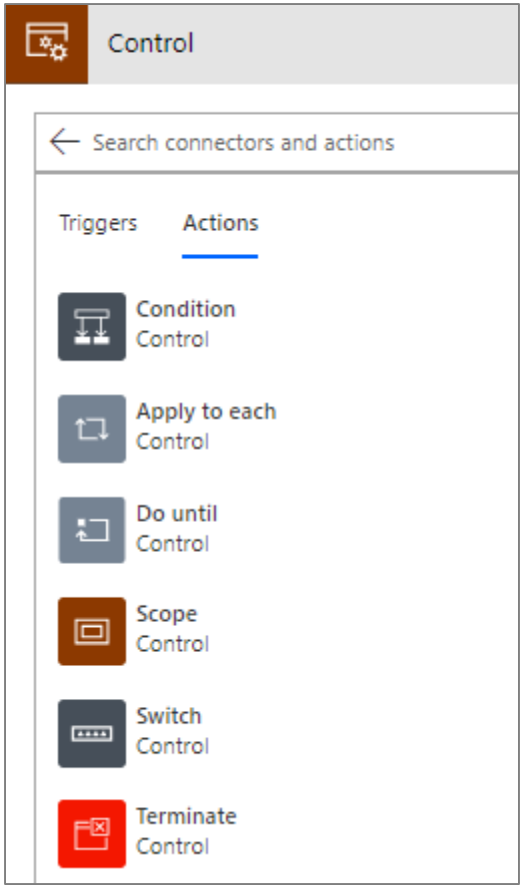

DEMO
Creating and Testing a Simple Flow

# Agenda

- ✓ Power Automate Fundamentals
- ✓ Creating and Testing Flows
- ➢ Using Control-of-Flow Actions
- Writing Flow Expressions
- Automating Document Generation

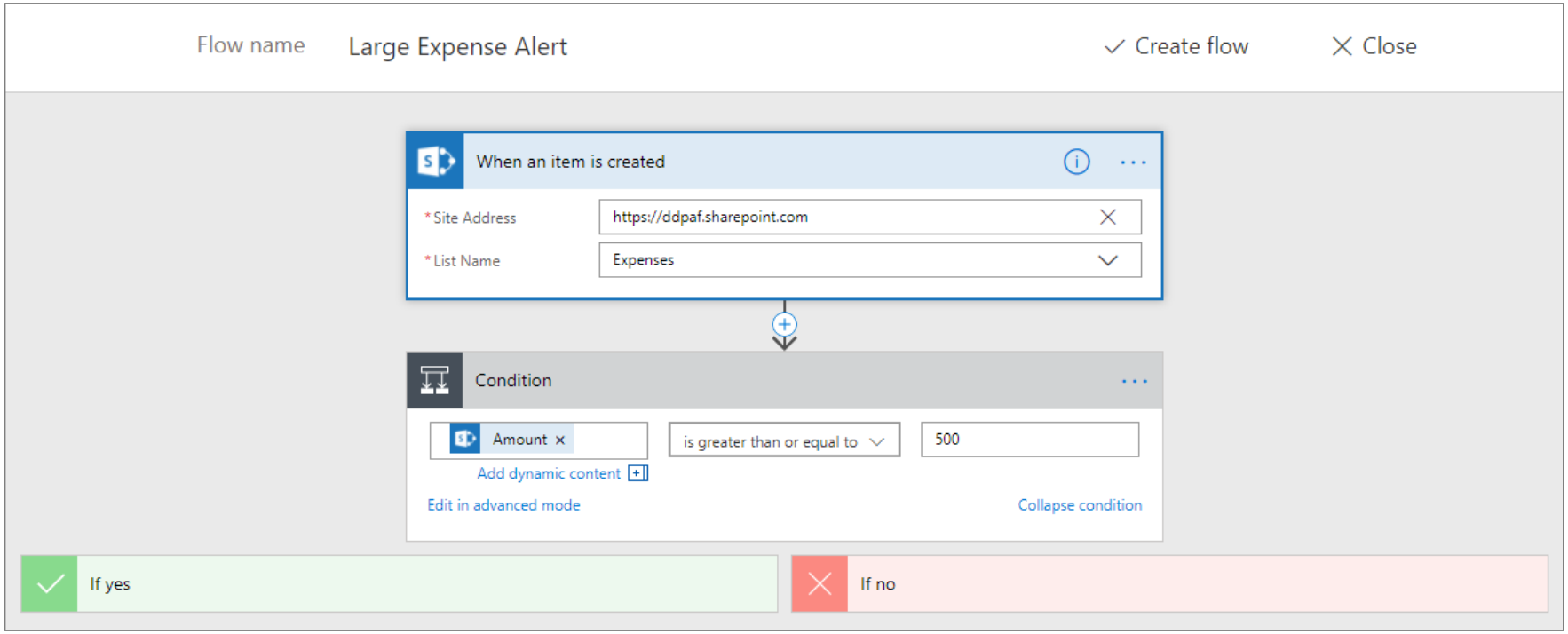
# Control of Flow

- Condition
  - Provides logical structure for If Then Else
- Apply to each
  - Enumerate through collection (e.g. list items)
- Do until
  - Repeat until condition changes
- Scope
  - Create an action container with a private scope
- Switch
  - Select Case flow
- Terminate
  - Completes a flow



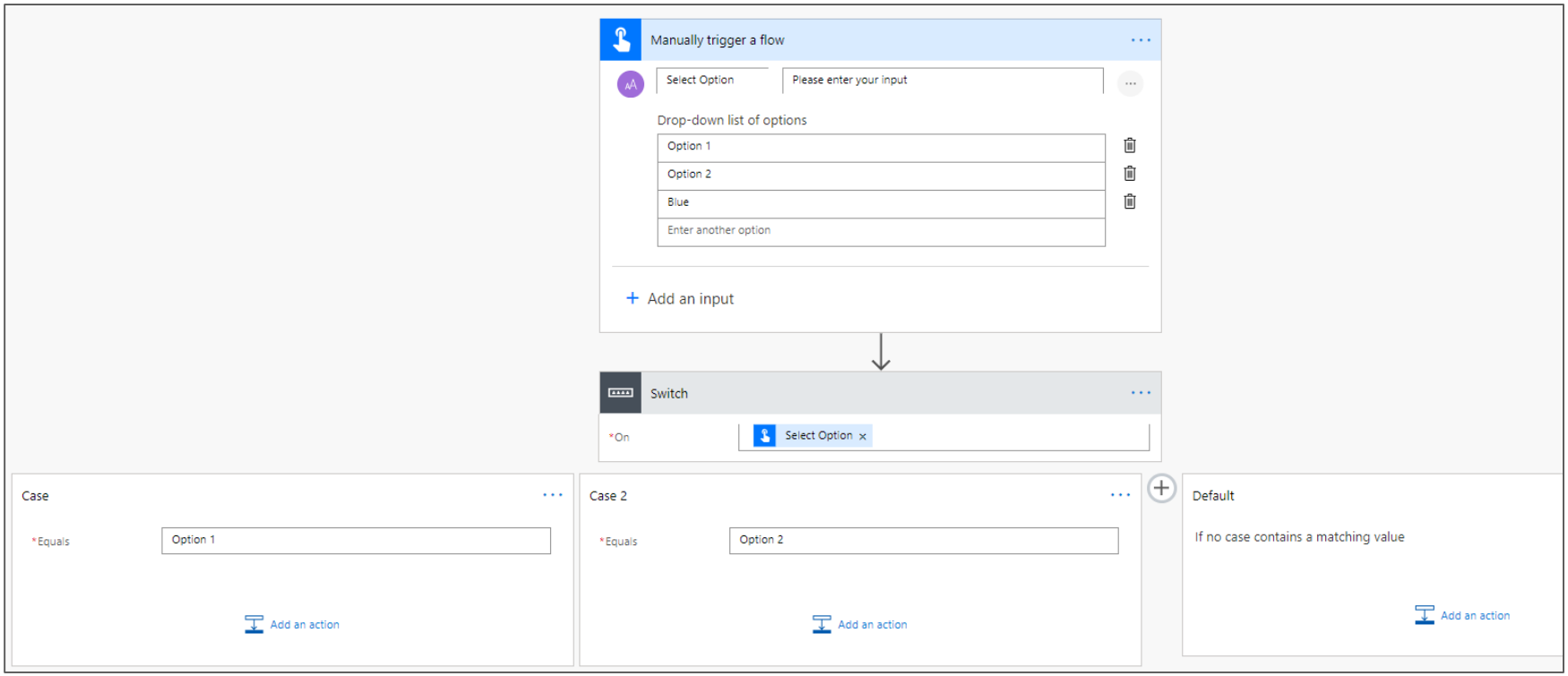
# Using a Condition

- Send alert email if expense amount greater than $500
  - Condition runs test which returns true or false
  - Condition provide **If yes** branch and **If no** branch



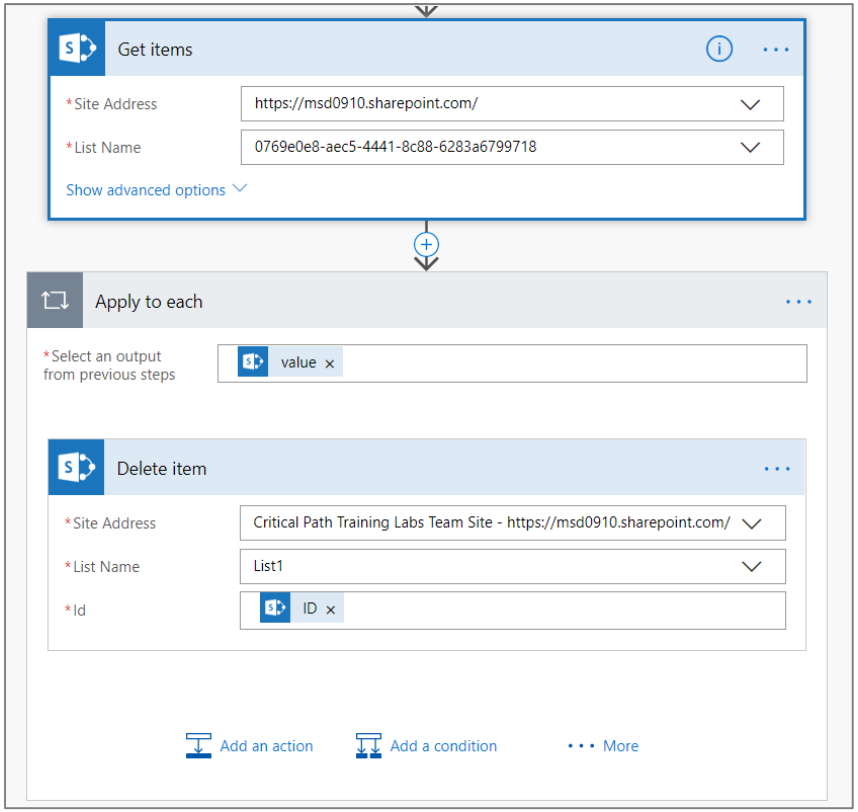
# Switch Action

- Switch actions provides cases
  - Each case represents separate execution path
  - Only one execution path will execute
  - Default case executes when no other case is matched



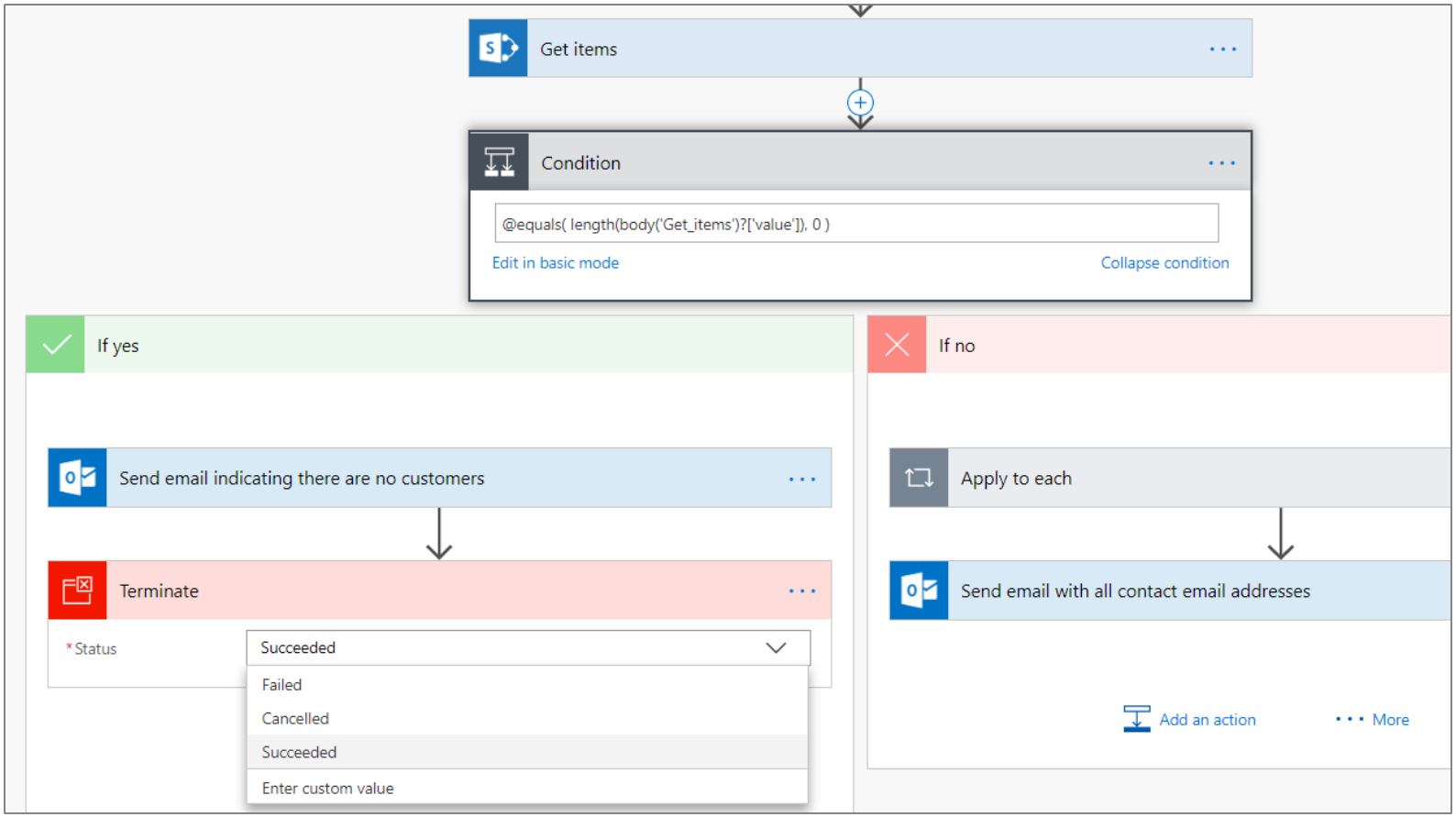
# Using Apply to Each

- Automatically added when list is used from output
- Destination step enumerates over list items



# Terminate action

- Used to stop a flow at any point
  - Terminate status can be set to Succeeded, Cancelled, Failed



# Agenda

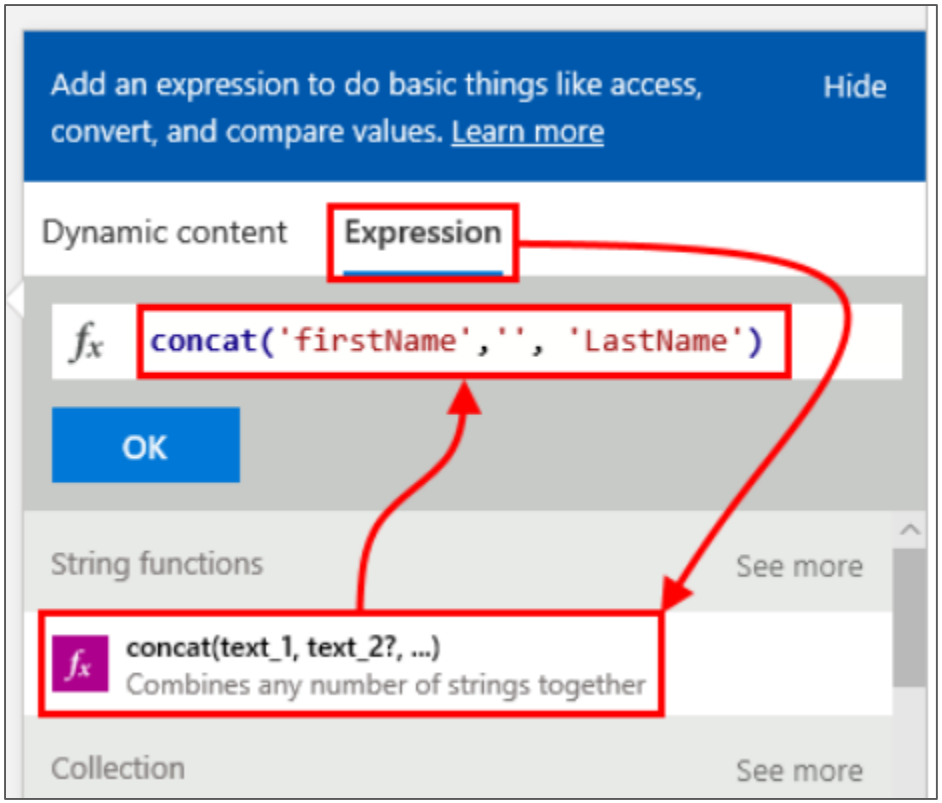
- ✓ Power Automate Fundamentals
- ✓ Creating and Testing Flows
- ✓ Using Control-of-Flow Actions
- ➢ Writing Flow Expressions
- Automating Document Generation

# Writing Flow Expressions

- Scenarios for writing Flow expressions
  - Perform string manipulation
  - Generate a GUID or a random number
  - Convert types
  - Perform simple inline calculations
  - Handling optional values
  - Writing conditional statements  using "If" statements
  - Working with arrays

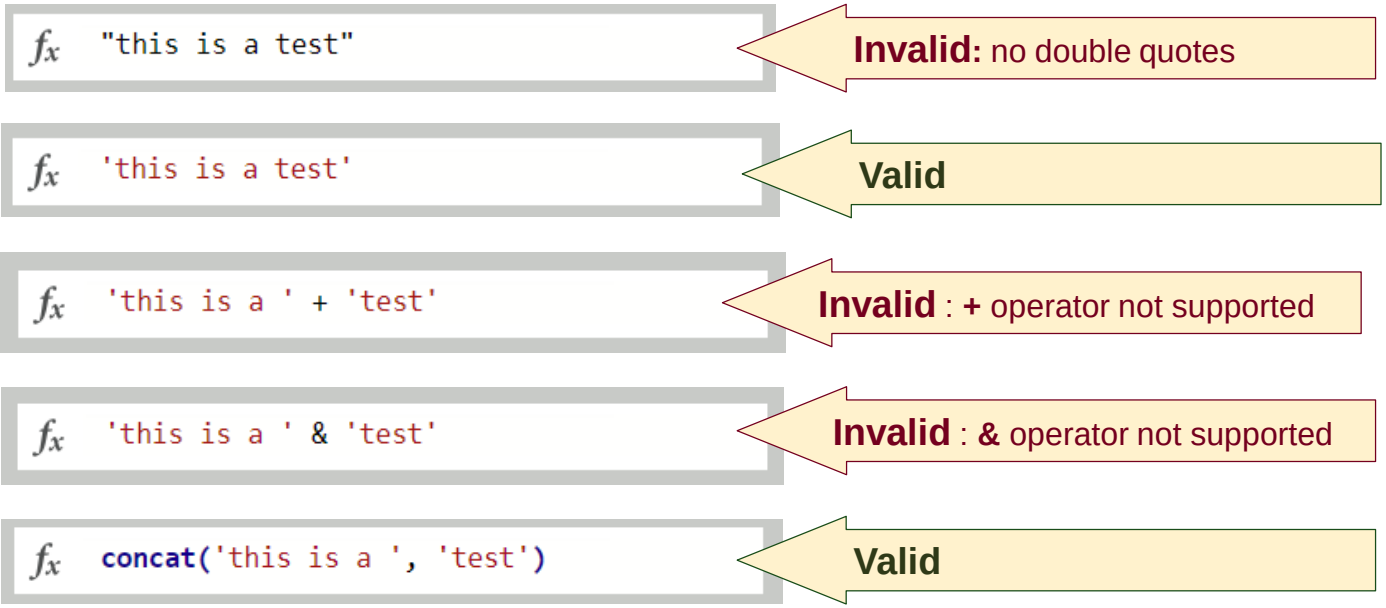
# Writing Expressions

- Expressions written in fx textbox
- Click OK to enter expressions



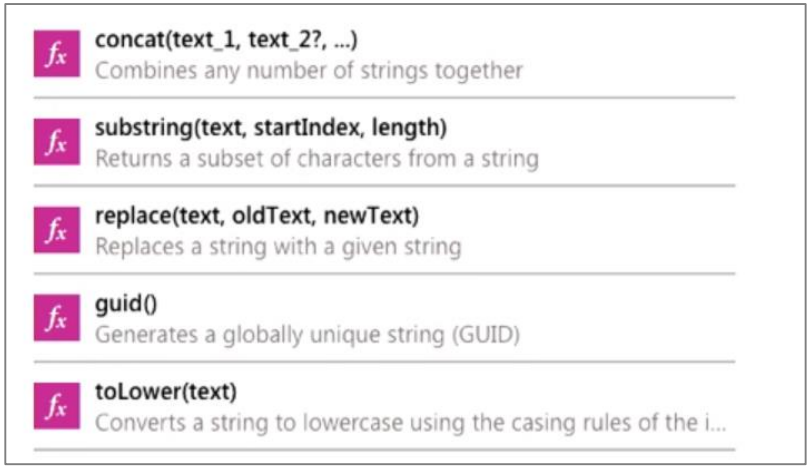
# Workflow Definition Language (WDL)

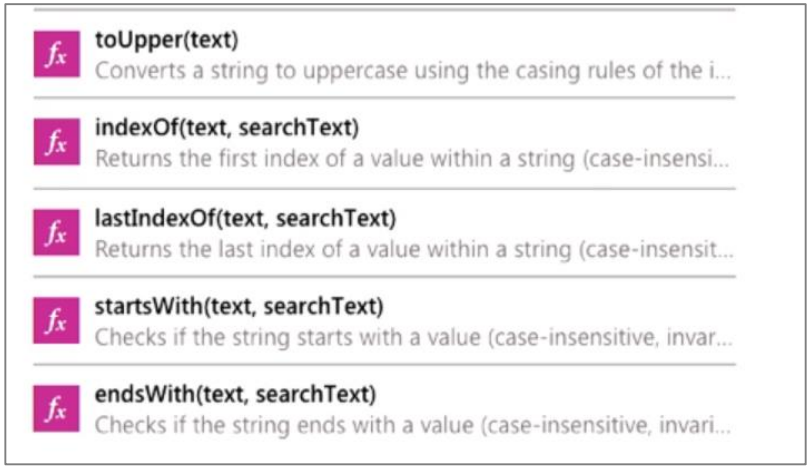
- Flow expressions written in Workflow Definition Language
  - Same language used in Azure Logic Apps
  - WDL is more powerful yet more complicated than PowerApps
  - WDL does not overload operators like PowerApps does
  - WDL requires single quotes instead of double quotes



# Working with Strings

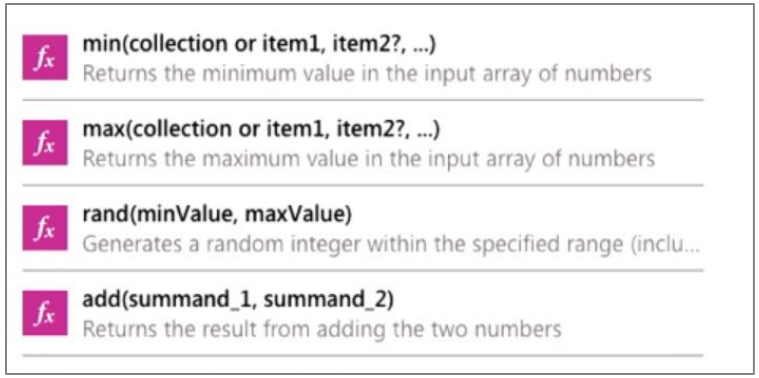
- Parse text together using **concat()**
- Parse out text using **substring()**
- Convert casing using **toLower()** and **toUpper()**
- Search string using **indexOf** and **startsWith()**
- Create new GUID identifier using **guid()**

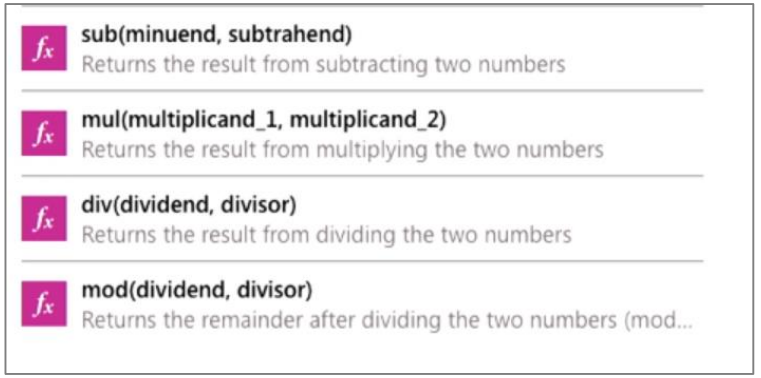


# Performing Arithmetic Operations

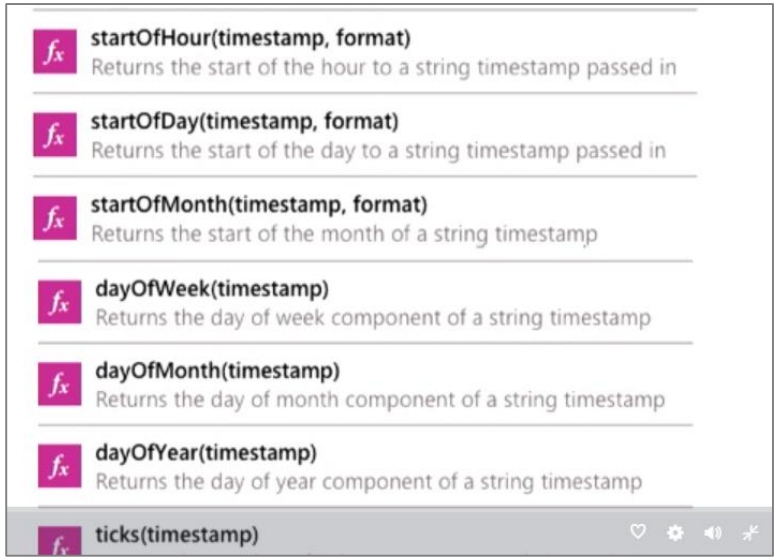
- You cannot use standard arithmetic operators
  - No support for familiar operators such as **+**, **-**, *******, **/**
  - This does not work: **2 + 2**
  - This works: **add(2, 2)**



# Working with Dates and Time

- Get Greenwich Meantime using **utcnow()**
- Use **add*()** functions to move time back/forward
- **convertTimeZone()** used to handle local times
- **formatDateTime()** used to format

# Understanding Arrays in Flow

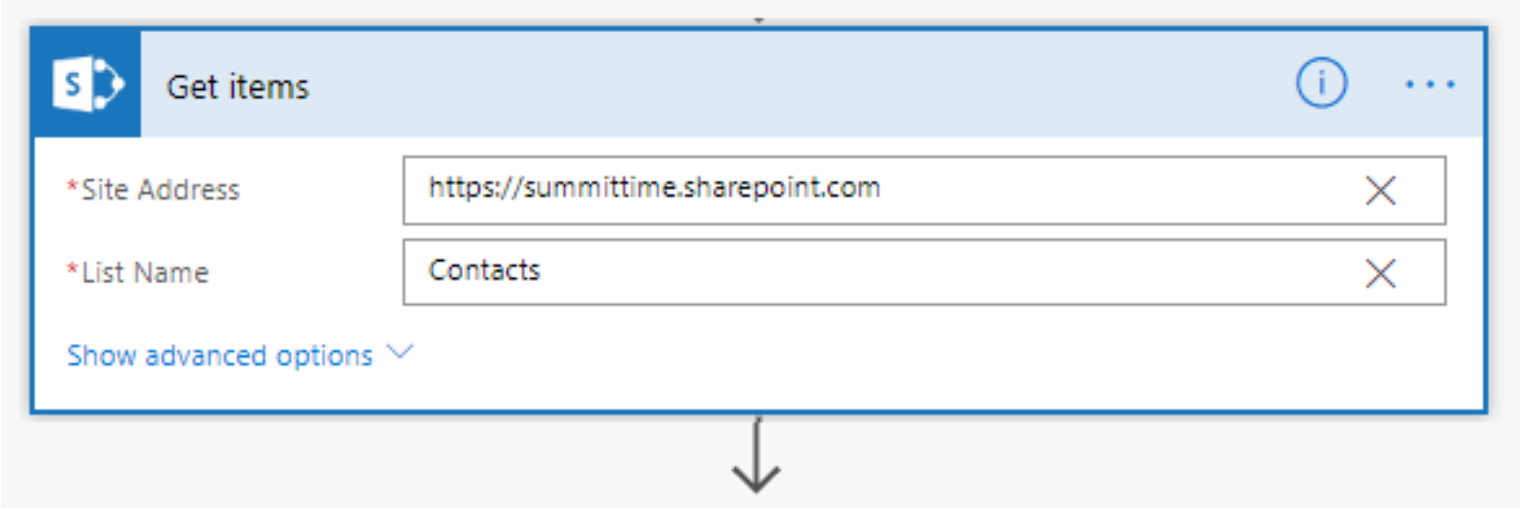
- Flow arrays are zero-based
  - Primitive value arrays

| | |
|---|---|
| 0 | Daugherty |
| 1 | Hernandez |
| 2 | Mack |
| 3 | Wiley |

  - Object arrays

| | Last Name | First Name | Company | Business Phone | Home Phone |
|---|---|---|---|---|---|
| 0 | Daugherty | Cindy | Wonka Industries | 1(337)111-4444 | 1(337)111-7777 |
| 1 | Hernandez | Zane | Vandelay Industries | 1(757)666-3333 | 1(757)777-1111 |
| 2 | Mack | Chang | Wonka Industries | 1(480)111-4444 | 1(480)777-0000 |
| 3 | Wiley | Ramona | Ecumena | 1(201)777-8888 | 1(201)777-2222 |

# Accessing an Array using ['value']

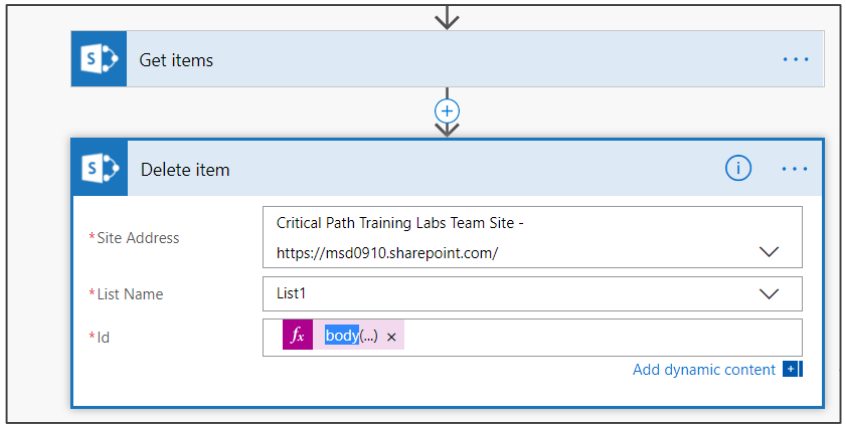


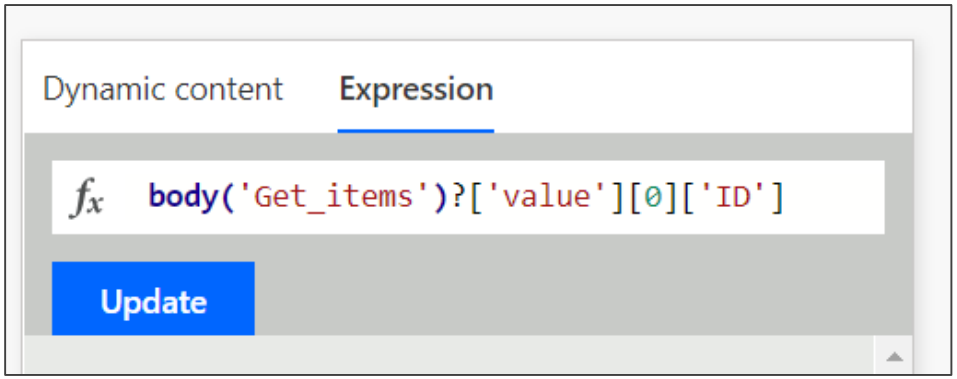

body('Get_items')?['value']

| | Last Name | First Name | Company | Business Phone | Home Phone |
|---|---|---|---|---|---|
| 0 | Daugherty | Cindy | Wonka Industries | 1(337)111-4444 | 1(337)111-7777 |
| 1 | Hernandez | Zane | Vandelay Industries | 1(757)666-3333 | 1(757)777-1111 |
| 2 | Mack | Chang | Wonka Industries | 1(480)111-4444 | 1(480)777-0000 |
| 3 | Wiley | Ramona | Ecumena | 1(201)777-8888 | 1(201)777-2222 |

# Retrieving List Items

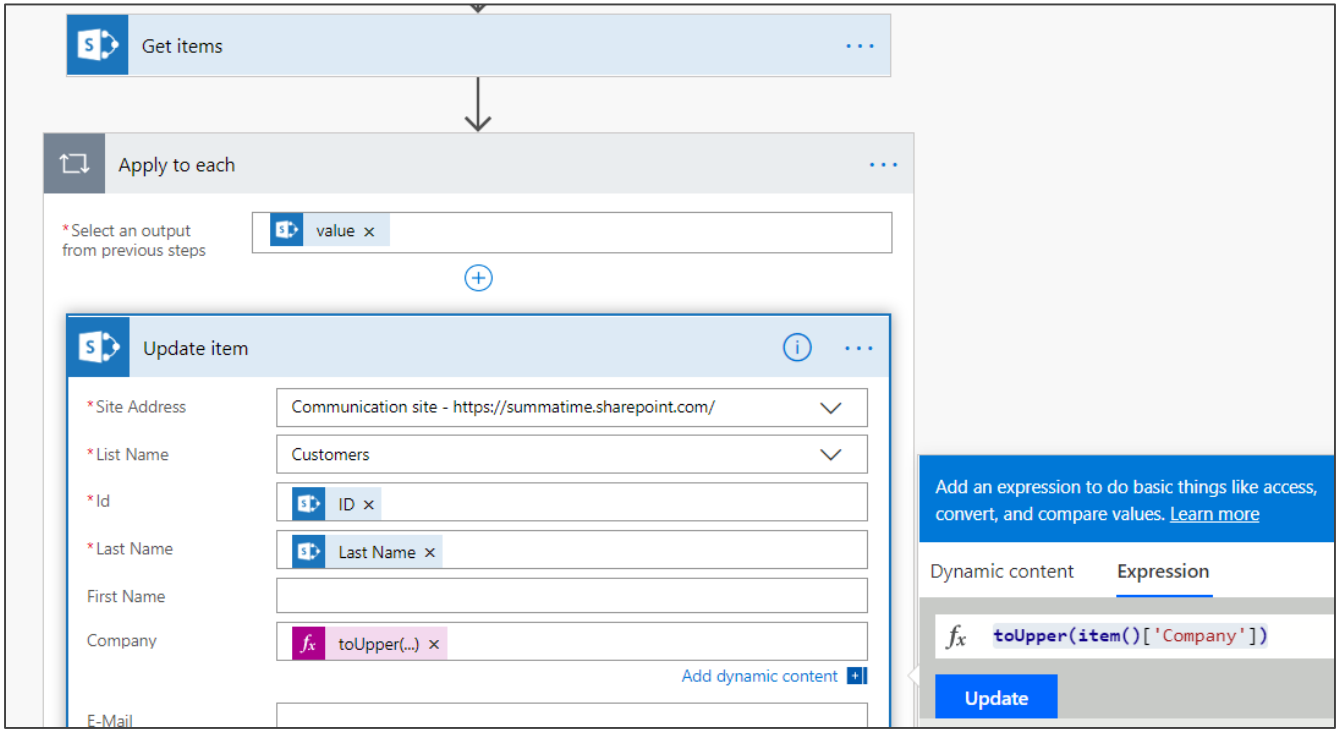
- Use **first()** and **last()** to get lead at head or tail
- Individual items retrieved using zero-based array syntax
  - SharePoint list item array - `body('Get_items')?['value']`
  - First list item in array - `body('Get_items')?['value'][0]`
  - Field value of first item - `body('Get_items')?['value'][0]['ID']`



# Writing Expressions with item()

- You can access current item inside flow loops
  - Use `item()` function provided by WDL
  - Example below formats all company names in upper case
  - Requires using `toUpper(item()['Company'])` in Apply to each loop



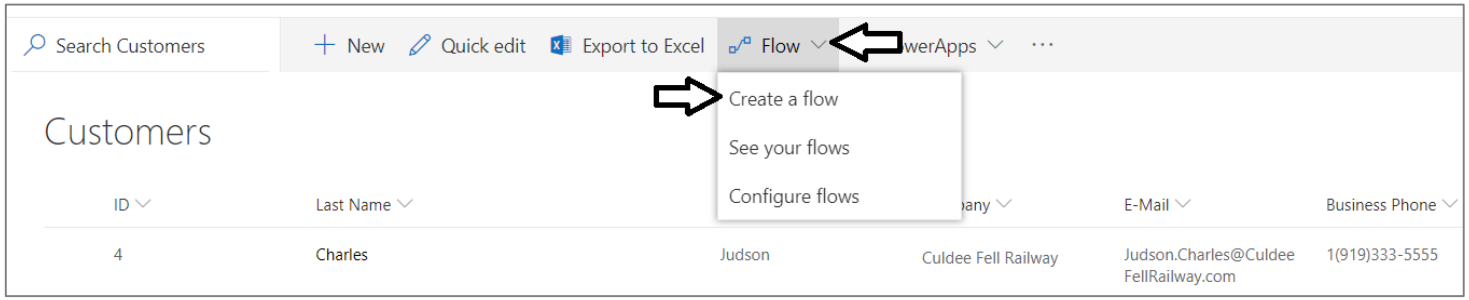

DEMO
Enumerating through SharePoint List Items to Perform Cleanup Operations

# Agenda

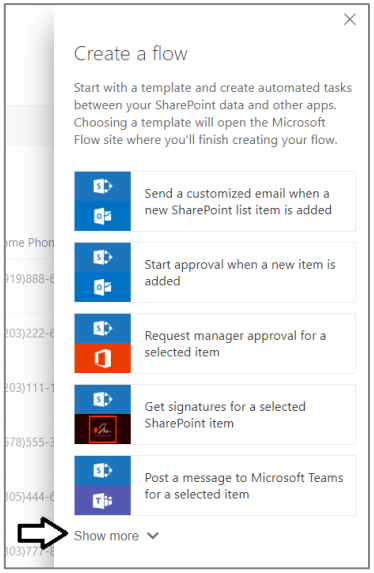
- ✓ Power Automate Fundamentals
- ✓ Creating and Testing Flows
- ✓ Using Control-of-Flow Actions
- ✓ Writing Flow Expressions
- ➢ Automating Document Generation

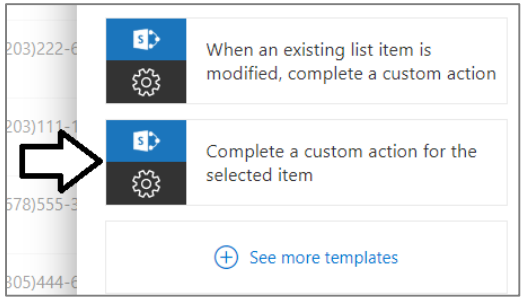
# Adding Flows for SharePoint Selected Item

- You can create a flow for a specific SharePoint list



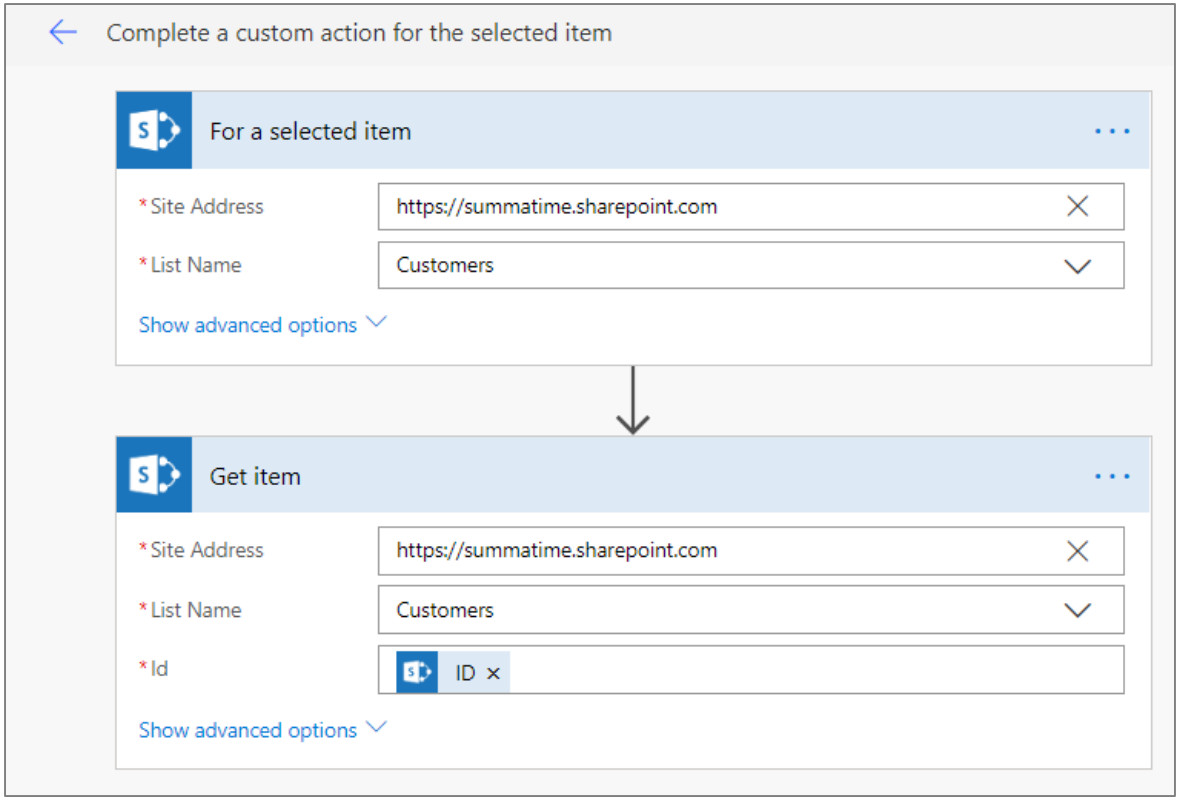

- Select option to Complete a custom action for the selected item



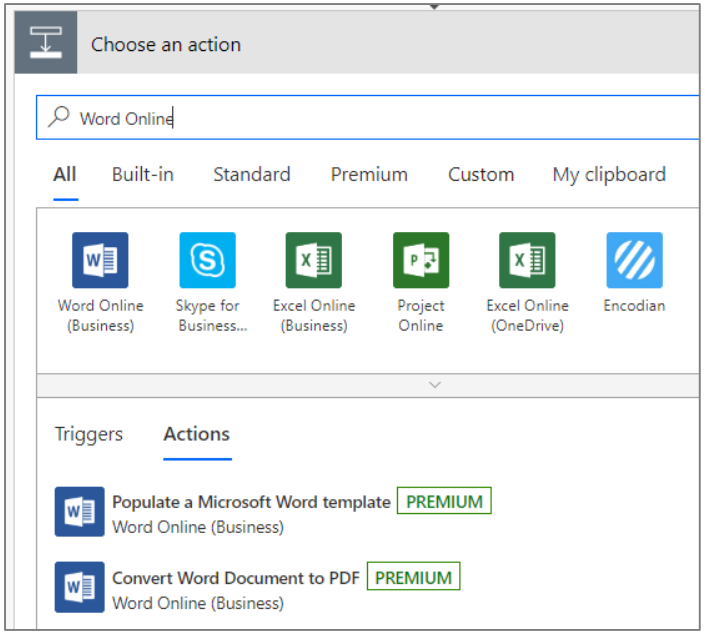
# Custom Action for Selected Item Flow

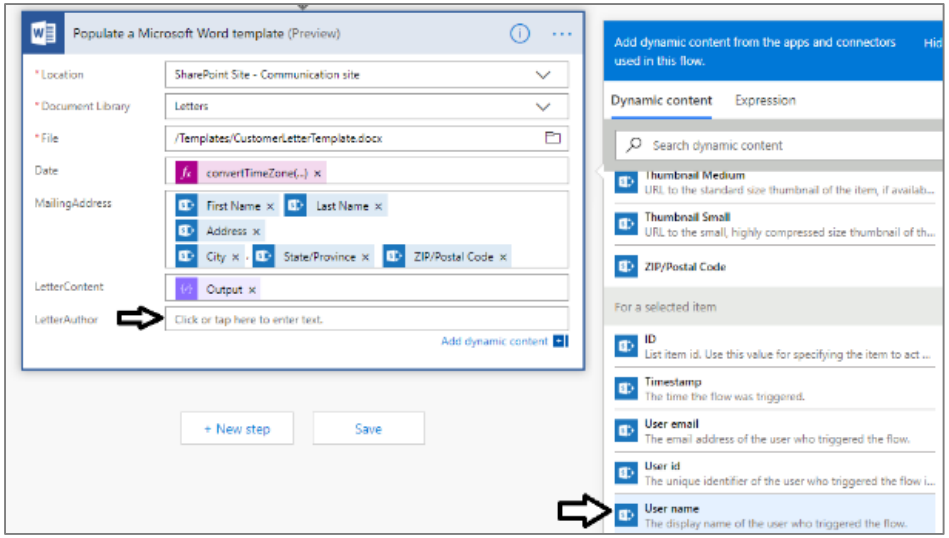
- Flow is initially created with a trigger and an action
  - Flow created with a **For a selected item** trigger
  - Flow also contain **Get item** action to get list item field values



# Word Online Connector

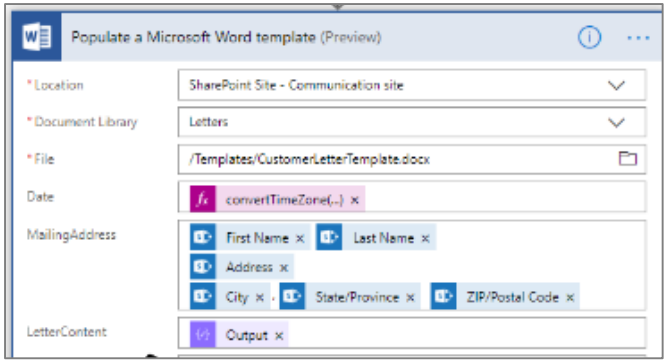
- Word Online connectors provides two valuable actions
  - Populate a Microsoft Word template
  - Convert Word Document to PDF



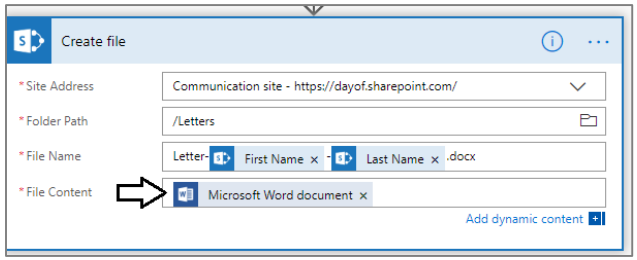



# Populate a Microsoft Word template

- Using Populate a Microsoft Word template action
  - Select Word document template with named content controls
  - Dynamically add content into named content controls

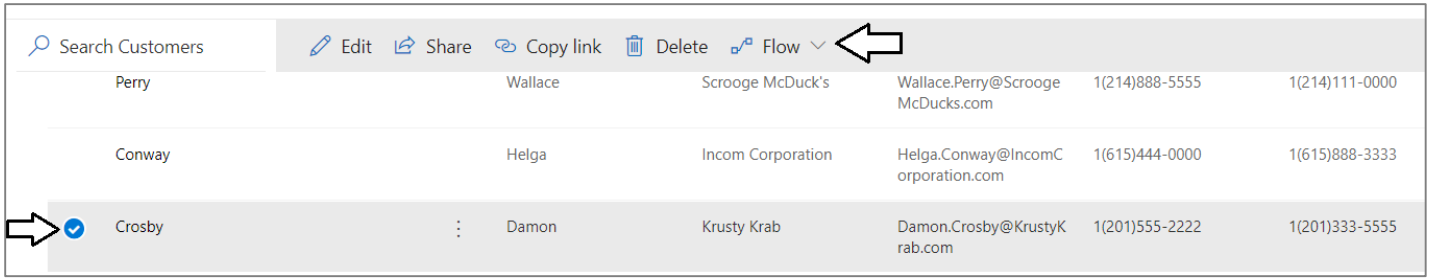



- Populate a Microsoft Word template doesn't create file
  - You must add another action to actually create the DOCX file

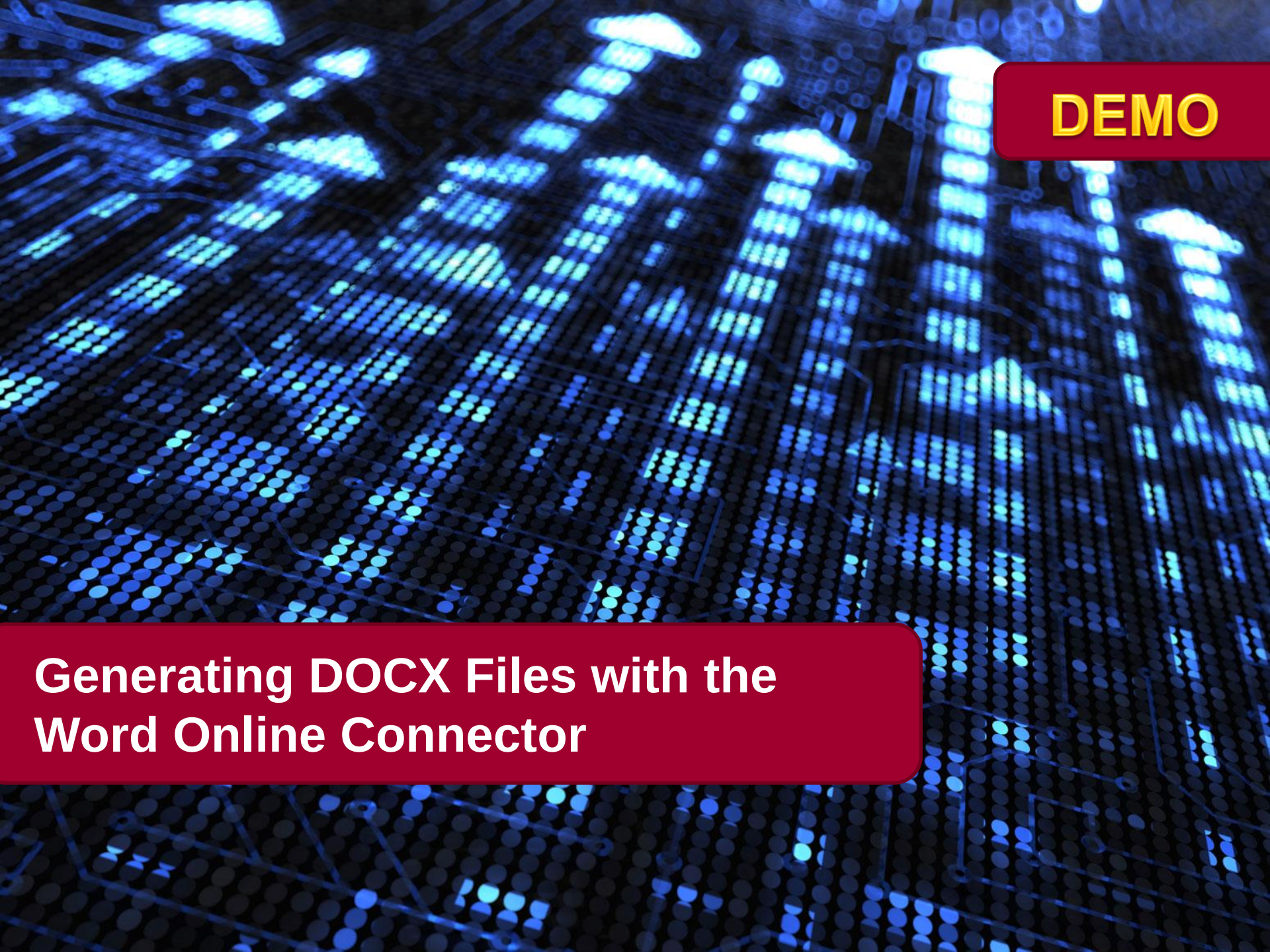


# Running a For a Selected Item Flow

- For a selected item flow runs on one item at a time
  - Make sure just one item is selected then drop down Flow menu




  - For a select item flows should appear in dropdown menu


DEMO
Generating DOCX Files with the
Word Online Connector

# Summary

- ✓ Power Automate Fundamentals
- ✓ Creating and Testing Flows
- ✓ Using Control-of-Flow Actions
- ✓ Writing Flow Expressions
- ✓ Automating Document Generation