



nordic.
summit

Elevate Your Power Automate Game by Using Expressions

Thank you to our Sponsors

Charter Sponsor



Microsoft

Platinum Sponsor



avanade

Organizing Sponsor



Thank you to our Sponsors

Gold Sponsors



Silver Sponsors



Online Sponsors



Fausto Capellan, Jr

- Microsoft MVP – Business Applications
- Power Platform Advocate
- Active Community Member
- Love Traveling



fausto@vecaent.com



[FaustoCapellanJr](https://www.linkedin.com/in/FaustoCapellanJr)



[Twitterhandle](#)



faustocapellan.com



Agenda

- Expressions Overview
- Types of Expressions
- Anatomy of an Expression
- Data Types and Content Conversion
- Expressions to Replace Actions
- Limitations
- Tips and Best Practices
- Demos









A woman with glasses and a striped shirt is looking at a laptop screen in a classroom setting. The background shows other laptops and people, suggesting a collaborative learning environment.

Expressions Overview

Expressions Overview

Set of functions that can be used to accomplish many different things in Power Automate

Expressions Overview

-  Convert data types
-  Manipulate strings
-  Retrieve and manipulate date and time
-  Generate data such as current time, GUIDs, or random numbers
-  Handle optional values
-  Apply conditional statements
-  Perform arithmetic functions
-  Flow leverages the [Azure Logic Apps Workflow Definition Language](#)

Types of Expressions

Basic

Contain a function with no parameters, or one or more parameters

```
utcNow()
```

```
workflow()
```

```
concat('Today is ', utcNow())
```

Advanced

Contain more than one function with one or more parameters

```
concat('Today is ', formatDateTime(utcNow(), 'MM-dd-yyyy'))
```

Anatomy of an Expression

Expressions are executed the same way as math operations: from inside out

```
addDays(startOfMonth(utcNow()), -1, 'MM-dd-yyyy')
```

1. Expression function: addDays()
2. No. of parameters: three separated by commas
3. Wrap static text in single quotes

A hand holding a black pen with a silver clip, pointing at a financial candlestick chart on a computer screen. The chart features green and red candlesticks, with a pink moving average line and a blue trend line. The background is a blurred blue screen displaying various financial data points and numbers. The text "Data Types and Content Conversion" is overlaid in a large, bold, black font on a semi-transparent white banner at the bottom of the image.

Data Types and Content Conversion

Data Types

- **String**

Represent alphanumeric data such as letters, numbers, spaces, symbols, punctuation marks.

- **Integer**

Whole numbers without decimals (can be either positive, negative, or zero).

- **Float**

Decimal number.

- **Boolean**

Represent two values: true or false.

Data Types (Continued...)

- **Arrays**

Represent a collection of elements, each selected by one or more indices (identifying keys) that can be computed at runtime during the execution of a program.

- **Dictionaries**

Represent a collection of key and value, and provides a mapping of the set keys with their values. Every key in the Dictionary must be unique and cannot be null.

- **Forms**

Contain elements called controls – textbox, dropdown, radio button - and each control manages a specific data type. These controls are used to display information to the user or receive information from the user.

Content Conversion

- Power Automate convert certain data types automatically
- Use expressions to manually convert data
- Conversion expressions are named according the data type to be converted

fx json(value)	Convert the input to a JSON type value
fx xml(value)	Covert the input to an Xml type value
fx int(value)	Convert the parameter to an integer
fx string(value)	Convert the parameter to a string
fx float(value)	Convert the parameter argument to a floating-point num...
fx bool(value)	Convert the parameter to a Boolean
fx base64(value)	Returns the base 64 representation of the input string
fx base64ToBinary(value)	Returns a binary representation of a base 64 encoded stri...
fx base64ToString(value)	Returns a string representation of a base 64 encoded stri...
fx binary(value)	Returns a binary representation of a value
fx dataUriToBinary(value)	Returns a binary representation of a data URI
fx dataUriToString(value)	Returns a string representation of a data URI
fx dataUri(value)	Returns a data URI of a value
fx decodeBase64(value)	Returns a string representation of an input based64 string
fx encodeURIComponent(value)	Url encodes the input string
fx decodeURIComponent(value)	Url decodes the input string
fx decodeDataUri(value)	Returns a binary representation of an input data URI string
fx uriComponent(value)	Returns a URI encoded representation of a value
fx uriComponentToBinary(value)	Returns a binary representation of a URI encoded string
fx uriComponentToString(value)	Returns a string representation of a URI encoded string
fx array(value)	Convert the input to an array
fx createArray(object_1, object_2?, ...)	Creates an array from the parameters

Content Conversion

		UI adds automatically									
To → From ↓	String	Base 64	Binary content	Data URI	URI comp.	Floating-point	Integer	Bool.	Array	JSON Object	XML content
String	Yes	base64()	binary()	dataUri()	uriComponent()	float()	int()	bool()	split() json()	json()	xml()
Base 64	base64ToString()	Yes	base64ToBinary()	*	*	*	*	*	*	*	*
Binary content	string()	base64()	Yes	dataUri()	uriComponent()	*	*	*	*	*	*
Data URI	dataUriToString()	*	dataUriToBinary()	Yes	*	*	*	*	*	*	*
URI comp.	uriComponentToString()	*	uriComponentToBinary()	*	Yes	*	*	*	*	*	*
Floating-point	Yes	base64()	binary()	dataUri()	uriComponent()	Yes	No	No	No	No	No
Integer	Yes	base64()	binary()	dataUri()	uriComponent()	Yes	Yes	No	No	No	No
Bool.	Yes	base64()	binary()	dataUri()	uriComponent()	No	No	Yes	No	No	No
Array	join() string()	*	*	*	*	No	No	No	Select Action	Select or Compose	xml()
JSON object	string()	*	*	*	*	No	No	No	Select or Compose	Compose Action	xml()
XML content	string()	*	*	*	*	No	No	No	xpath()	xpath()	Logic apps only

* First convert to/from a string and then use the expression required for normal strings



Expressions to Replace Actions

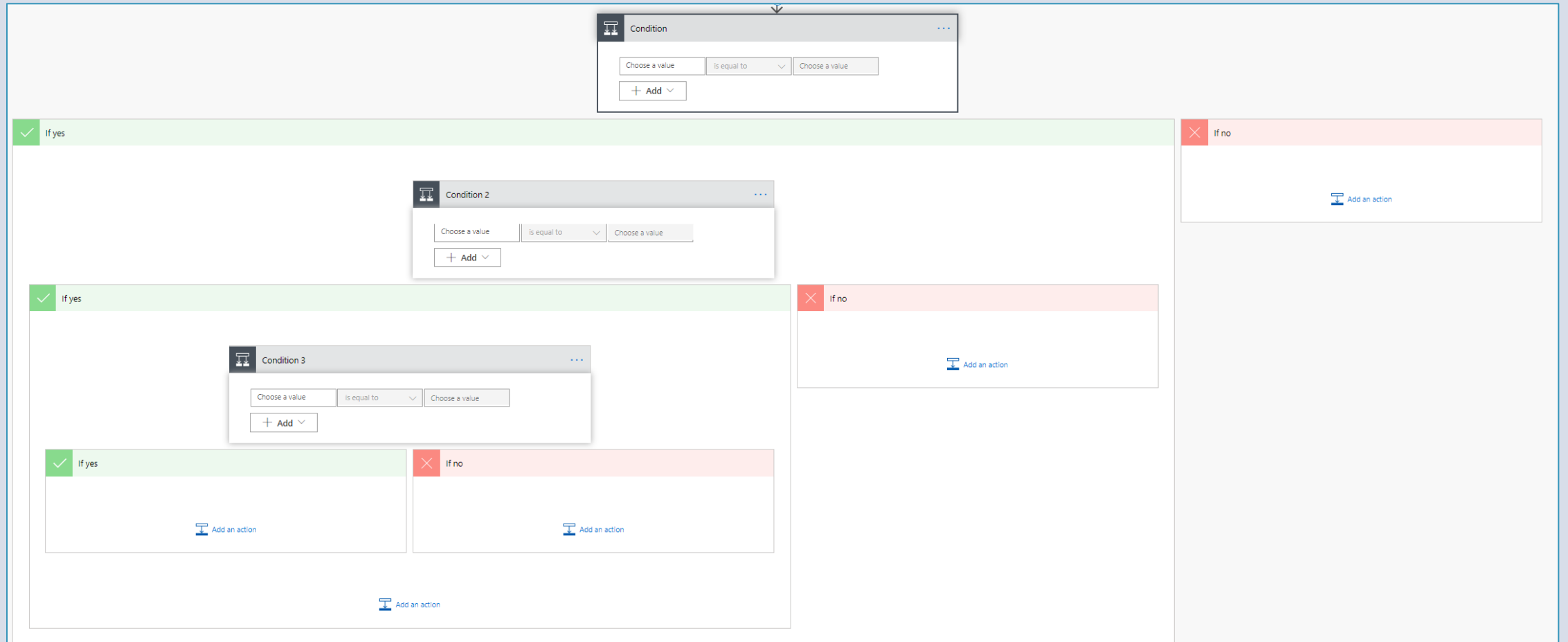
Expressions to Replace Actions

The if() expression

Why would you use the if() expression if you can use a condition?

- Avoid multiple nested conditions
- Avoiding multiple nested conditions prevents repeating the same set of actions
- There's a limit of 8 nested conditions

Expressions to Replace Actions



Expressions to Replace Actions

The `first()` expression

- Get the first letter from a string
- Get the first record from an array
- Avoid an Apply to each action when selecting a property from the first record
- Combined with the `split()` expression, get the left side of the string where the `split()` is executed
 - Note: This combination is the equivalent of the LEFT function in Excel

Expressions to Replace Actions

The last() expression

- Similar to the first() expression, except it starts from the right
- Get the last letter from a string
- Get the last record from an array
- Avoid an Apply to each action when selecting a property from the last record
- Combined with the split() expression, get the right side of the string where the split() is executed
 - Note: This combination is the equivalent of the RIGHT function in Excel

Expressions to Replace Actions

Date and Time expressions

- These functions are used to generate data such as current date and time
 - `utcNow()`
- Also used to manipulate date and time
 - `convertFromUtc()`
 - `convertTimeZone()`
 - `formatDateTime()`
 - `addDays()`

Date and time		See less
<code>fx</code>	<code>utcNow()</code> Returns the current timestamp as a string	
<code>fx</code>	<code>getFutureTime(interval, timeUnit, format?)</code> Returns a timestamp that is the current time plus the spe...	
<code>fx</code>	<code>getPastTime(interval, timeUnit, format?)</code> Returns a timestamp that is the current time minus the s...	
<code>fx</code>	<code>addToTime(timestamp, interval, timeUnit, format?)</code> Adds an integer number of a specified unit of time to a s...	
<code>fx</code>	<code>subtractFromTime(timestamp, interval, timeUnit, forma...</code> Subtracts an integer number of a specified unit of time fr...	
<code>fx</code>	<code>addSeconds(timestamp, seconds, format?)</code> Adds an integer number of seconds to a string timestamp...	
<code>fx</code>	<code>addMinutes(timestamp, minutes, format?)</code> Adds an integer number of minutes to a string timestamp...	
<code>fx</code>	<code>addHours(timestamp, hours, format?)</code> Adds an integer number of hours to a string timestamp ...	
<code>fx</code>	<code>addDays(timestamp, days, format?)</code> Adds an integer number of days to a string timestamp p...	
<code>fx</code>	<code>convertTimeZone(timestamp, sourceTimeZone, destina...</code> Converts a string timestamp passed in from a source tim...	
<code>fx</code>	<code>convertToUtc(timestamp, sourceTimeZone, format?)</code> Converts a string timestamp passed in from a source tim...	
<code>fx</code>	<code>convertFromUtc(timestamp, destinationTimeZone, for...</code> Converts a string timestamp passed in from a UTC to a ta...	
<code>fx</code>	<code>formatDateTime(timestamp, format)</code> Returns a string in date format	
<code>fx</code>	<code>startOfHour(timestamp, format)</code> Returns the start of the hour to a string timestamp passe...	
<code>fx</code>	<code>startOfDay(timestamp, format)</code> Returns the start of the day to a string timestamp passed...	
<code>fx</code>	<code>startOfMonth(timestamp, format)</code> Returns the start of the month of a string timestamp	
<code>fx</code>	<code>dayOfWeek(timestamp)</code> Returns the day of week component of a string timestamp	
<code>fx</code>	<code>dayOfMonth(timestamp)</code> Returns the day of month component of a string timesta...	
<code>fx</code>	<code>dayOfYear(timestamp)</code> Returns the day of year component of a string timestamp	
<code>fx</code>	<code>ticks(timestamp)</code> Returns the number of ticks (100 nanoseconds interval) si...	

Limitation

- Expression Evaluation Limit
 - A Flow expression can evaluate up to 131,072 characters
- Max Number of Characters per Expression
 - Flows allow a maximum of 8,192 characters per expression
- The expression editor is too small *

Tips and Best Practices

- Power Automate reads and processes date and time in Coordinated Universal Time (UTC)
- Rename actions with meaningful names
- Add comments to actions, especially if they contain expressions
- Filter data inside **Get** actions
- Use **Configure run after** to test or for error handling
- Use **Compose** actions instead of **Variables** for static values
- Replace spaces in action names with underscore when building expressions around said action

A close-up photograph of a group of people gathered around a silver laptop. One person's hand is resting on the keyboard, while another's hand points towards the screen. A third hand is visible on the right, also pointing. The background is blurred, showing other people in a professional setting. A semi-transparent white banner is overlaid at the bottom of the image.

Demos

Please give me feedback

- Remember to give my session feedback!
- Scan the QR code and find my session in the session list



Thank you to our Sponsors

Charter Sponsor



Platinum Sponsor



Organizing Sponsor



Gold Sponsors



Silver Sponsors



Online Sponsors





nordic.
summit