Data Operations

1. Basic Arithmetic Operations
2. Relational Operations
3. Logical Operations
4. String Manipulations

1. Basic Arithmetic Operations:

Mathematical Operations - Addition (+), Subtraction (-), Multiplication (*), and Division (\), all work exactly the way you would expect for numeric data types. These operations don't work for the other data types such as Boolean, Object, DateTime etc. However, Addition operation can be used with String variables where it concatenates the two strings provided as input.

Additionally, we often use "Modulus" operator, which gives the remainder of division operation. It's denoted by mod keyword. Example: 25 mod 4 will give the output as 1.

Exponentiation uses the ^ Operator, as the following example demonstrates

```
z = 4 ^ 3
' The preceding statement sets z to 64 (the cube of 4)
```

2. Relational/Comparison Operations:

Relational Operations define the relationship one operand has to the other. In practical scenarios, these are mostly used in making decisions and the validity of the condition determine the further execution path of a workflow. They determine the equality and ordering.

The Relational/Comparison Operations are shown as below:

Operator	Result
=	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

The result of a relational operation is a "Boolean" value.

Sample Use-Case: Consider a sample case where you intend to check which among two numbers is greater and print the output in a message box window.

3. Logical Operations:

Logical operations are the ones that act on Boolean operands and form a resultant Boolean Value.

There are 3 Logical Operations and shown as below:

Operator	Result
AND	Logical AND
OR	Logical OR
NOT	Logical unary NOT

Ex: For any Boolean Expressions A, and B, the effect of each of these logical operations is shown below:

A	B	A AND B	A OR B	NOT A
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

This signifies that for an **"AND"** operation to be **TRUE**, **both** (or **all**) conditional expressions acting as operands **must be TRUE**. Even a **single failure results** is **FALSE** output.

On the contrary, if any operand is an **"OR'** operation hold **TRUE**, the **resultant** will be **TRUE** as well.

"NOT" operation simple inverts the validity criteria of a condition.

4. String Manipulations:

Strings are considered to be the simplest form of Arrays. A String is an Array of characters forming a text. There are various String Manipulating Functions supported with in UiPath.

Some of the important String Methods are described below:

Methods	Result
Contains	Checks for value within a String
Replace	Replaces a text in a string with a given value
Split	Splits string based on delimiter
Remove	Returns a new string in which a specified number of characters from the current string are deleted
Substring	Retrieves a substring from this instance
Join	Returns a string created by joining a number of substrings contained in an array
Left	Returns a string containing a specified number of characters from the left side of a string

Right	Returns a string containing a specified number of characters from the right side of a string
Trim	Trim specific part of a string
LTrim	Returns a string containing a copy of a specified string with no leading spaces
RTrim	Returns a string containing a copy of a specified string with no trailing spaces
ToLower/ToUpper	Converts string to lowercase / uppercase
EndsWith/StartsWith	Checks if a string starts / ends with particular characters

For more information you can refer the following URL's:

1. String Functions in VB.Net:

https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/functions/string-functions

2. String Methods in .Net Framework:

https://docs.microsoft.com/en-us/dotnet/api/system.string?view=netframework-4.8