PowerApps Functions

If function

23099

Why

If is one of the most used functions in any programming language and PowerApps is not the exception.

Test for a specific condition. You can use the If function to execute a single condition.

Common examples include:

- To test if an argument is TRUE or FALSE
- To output a NUMBER
- To output some TEXT
- To output some CALCULATION

Overview

1. The If function tests if the first condition is met and returns the result. You can define an optional value when all conditions are false.
2. So an IF statement can have two results. The first result is if your comparison is True, the second if your comparison is False.
3. Nested If - It's called **nested** because you're basically putting an **If** Statement inside another IF Statement and possibly repeating that process multiple times.

PowerApps If Function Syntax

If(logical_test, value_if_true, [value_if_false])

There are 3 parts (arguments) to the If function:

- **logical_test** something, such as the value in a label or field.
- Specify what should happen if the test result is **true**.
- Specify what should happen if the test result is **false**.

Logical_test

Required. The value that you want to test. Formula(s) to test for true. Such formulas commonly contain comparison operators (such as <, >, and =) and test functions such as IsBlank and IsEmpty.

When using the IF function to construct a test, we can use the following logical operators:

Operator	Description
=	Equal To
<>	Not Equal To
>	Greater Than
<	Less Than
>=	Greater Than or Equal To
<=	Less Than or Equal To

value_if_true

Required. It is the value that is returned if *condition* evaluates to TRUE. The corresponding value to return for a condition that evaluates to true.

The **Value If True** argument can contain just about anything:

- A number

- Text wrapped in double quotes – “hello”
- A reference to another label or field – Label# or DataCardValue#
- A formula with another IF function.
- A formula with any other combination of functions.

value_if_false

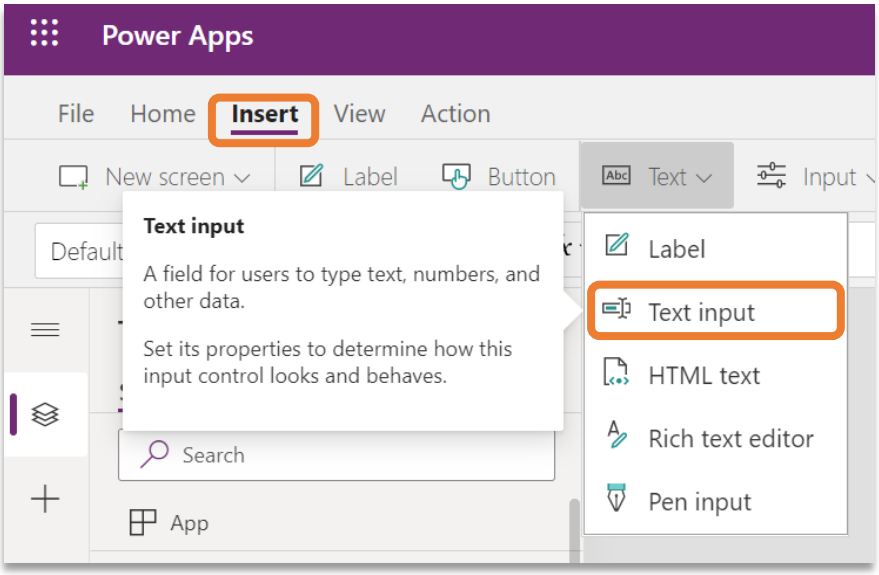
Optional. It is the value that is returned if *condition* evaluates to FALSE. The value to return if no condition evaluates to true. If you don’t specify this argument, a blank is returned.

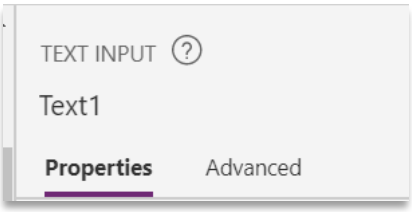
Scenario

1. You have a process where greater than or equal to 50 is a Pass and less than 50 is a fail.
2. You wish to create an automation to return Pass or Fail, depending on the number you input.
3. You are using PowerApps and navigate to a screen for editing records.

How?

1. Add **a Text input control**, and name it **Text1** if it doesn't have that name by default. Click **Insert**, Select **Text** and Choose **Text input**.



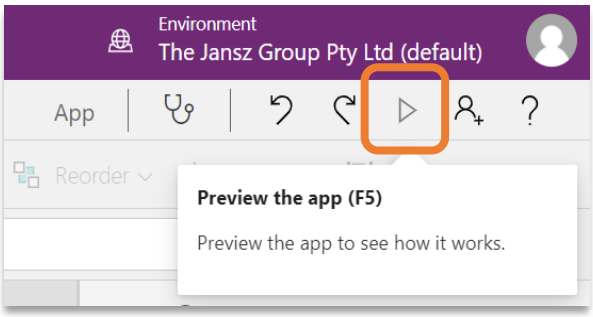


2. Add a **Label** control, Click **Insert**, Select **Text** and Choose **Text input**.

3. Set its **Text** property to this formula:

4. **If(Value(Text1.Text) >=50, "Pass", "Fail")**

5. Click **Preview the App** or press **F5** on keyboard



6. In Text1, type 60.

7. The Label control shows **Pass**, because the value of Text1 is more than 50.

8. In Text1, type 15.

9. The Label control shows **Fail**, because the value of Text1 is less than 50.

Nested If

The IF function in Excel can be nested, when you have multiple conditions to meet. The FALSE value is being replaced by another IF function to make a further test.

You can only nest 50 If statements. Once you reach this limit, think if you're overcomplicating the overall formula. Re-evaluate the conditions and combine validations into smaller chunks.

The first common use case scenario is to determine where an input value sits within a set of ranges, or bandings. When nesting IF functions, be sure to put the maximum tests first, or the results could be incorrect.

Scenario

1. You have a range of results from a test.
2. You wish to automatically assign grades based on a set method
3. You want to enter a numerical score to produce the grade.

Grades are assigned as follows:

- **A** - Score greater than 85
- **B** - Score between 84 and 75
- **C** - Score between 74 and 65
- **D** - Score between 64 and 50
- **F** - Score < 50

The English language version of your code reads something like this:

1. If the score is greater then 85 award an A,
2. If not: If the score is [not greater than 85 and] more than 75 award a B
3. If not: If the score is [not greater than 75 and] more than 65 award a C
4. If not: If the score is [not greater than 65 and] more than 50 award a D
5. Otherwise: award an F [Only scores less than 50 should be left]

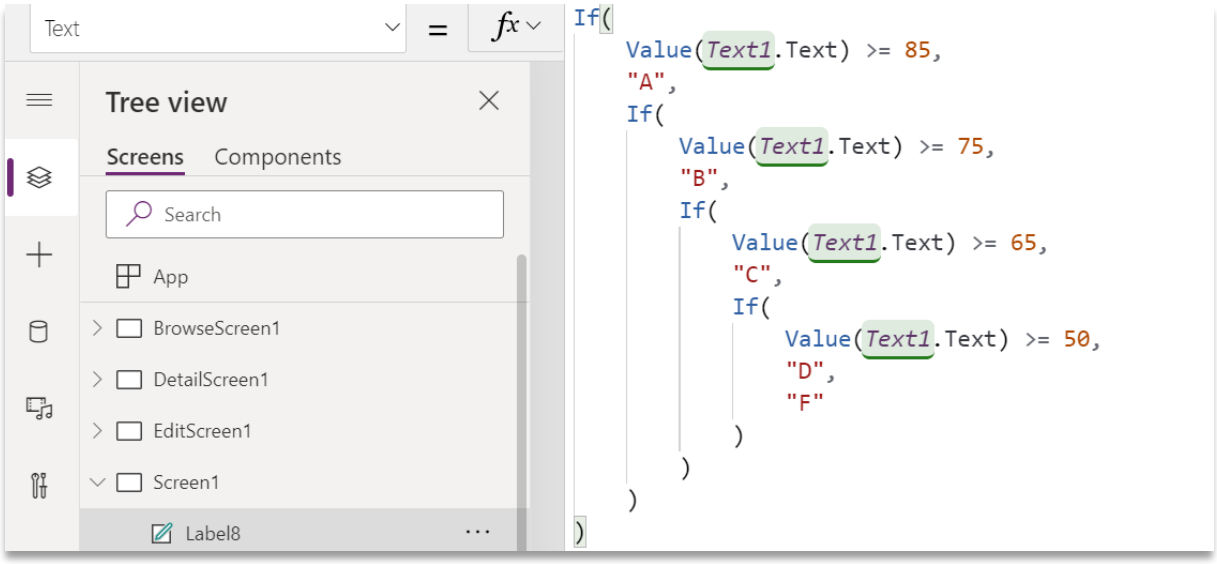
How?

10. From Previous Exercise you have added **a Text input control**, and name it **Text1**.

11. Add a **Label** control, Click **Insert**, Select **Text** and Choose **Text input**.

12. Set its **Text** property to this formula:

13. **If(Value(Text1.Text) >=85, "A", If(Value(Text1.Text) >=75,"B", If(Value(Text1.Text) >=65,"C", If(Value(Text1.Text) >=50,"D","F"))))**



14. Click **Preview the App** or press **F5** on keyboard and test out values