Writing Conditional Statements

Introduction
Conditional statements are effective for changing and manipulating data of all types. Data can be mathematically changed, altered to reflect a categorization, or flagged for downstream filtering based on the criteria you specify. Using the conditional functions in the Formula tool, you can easily and effectively apply changes to your data.

Conditional Logic
Basic conditional statements in Designer are comprised of four parts, or "clauses": IF, THEN, ELSE and ENDIF. These four words create the framework for applying a logical change to your data.

A conditional statement begins with specifying the criteria to use for applying a change your data. That condition may test a numeric value, the presence of certain text or the status of a data cell in a particular column of data. These criteria are specified in the IF clause of a conditional statement.

If a condition is met, THEN apply a change to that data. Since the change being applied in the THEN clause is an output, it's important to make sure that the syntax, such as the use of quotes surrounding string data, matches the data type of the destination column of the data in Formula tool's configuration.

If a value is not changed by the logic in the THEN clause, what should happen? Should another function be applied? Should different text replace the cell's value? Or, should it be left as its original value? The ELSE clause will apply this logic to all values that do not meet the criteria specified in the IF clause.

The last clause is ENDIF, which indicates that the conditional function is complete.

A dataset contains information on website use, like the user accessing the site, the date and day of their visit, the length of their visit and their search content. Use a conditional statement in the Formula tool to create a new numeric column called [Weekend] to flag the sessions that took place on a Saturday or Sunday.

Create a conditional statement to apply the following logic: If a session occurred on a Saturday or a Sunday then assign a value of one (1) in the new column [Weekend]. Otherwise, assign a value of zero (0).

Click the Function Library and expand the Conditional function category. Select the first option from the list to insert the function into the Expression Editor.

The function includes the four clauses of a conditional statement. Insert the logical criteria and actions in the placeholders "c", "t", and "f".

Enter the condition to be evaluated in the placeholder "C". In this case, identify the values in the column [Day] which are equal to Saturday or Sunday. Because the values in the column [Day] are a string datatype, wrap the days of the week in quotes.
If a session occurred on a Saturday or a Sunday then assign a value of one (1) in the new column [Weekend]. Otherwise, assign a value of zero (0). In the placeholder "t", type the appropriate value that should be used if the value in the column [Day] is equal to Saturday or Sunday. Then, click Submit.

In the placeholder "f", type the value that should be used if the value in the column [Day] is not equal to Saturday or Sunday. Then, click Submit.

After running the workflow, a new column called [Weekend] contains ones and zeroes to flag our weekend days and weekdays.

Inline IF Function
In addition to IF-THEN statements, the Formula tool's conditional functions also include Inline IF statements, which can evaluate one set of criteria. You can think about Inline IF statements as shortened version of a conditional statement. An inline IF's syntax requires three pieces of information: a condition to evaluate, a value to apply if a value meets that condition, and a value to apply if a value does not meet the condition. Note that no "ENDIF" is required for an inline if statement.

In the column [Weekend Flag], enter the function IIF. In the Inline IF function, enter the Boolean expression to evaluate: Day = Saturday OR Day = Sunday. If the value in our dataset meets this condition, then we'll apply the change indicated by the placeholder "x": assign a value of 1. Otherwise, we'll apply the function indicated by the placeholder "y": assign a value of 0. After running the workflow, we can see that Saturday and Sundays have been assigned a value of 1 while weekdays have been assigned a value of 0.

Values equal to Saturday or Sunday should be assigned a value of 1. All others should be flagged as a zero. Drag the values into the correct placeholder in the expression.

After running the workflow, weekend days have been flagged with a one (1) and weekdays have been flagged with a zero (0).

Multi-Conditional Function
Conditional statements are not limited to two outcomes. Add an ELSEIF clause to evaluate additional conditions in more complex situations. ELSEIF clauses are always followed by a THEN clause to apply an action to values that meet the criteria. There is no limit to the number of ELSEIF clauses that can be used in a multi-conditional statement. An ELSE and ENDIF clause complete a multi-conditional statement.

Classify the durations of site visits as "Short", "Medium" and "Long" with a multi-conditional statement. Visits lasting 30 minutes or more are "Long". "Medium" visits last 5 to 30 minutes, and any session less than 5 minutes long is "Short".

Create a column called [Category] that is a string data type. Click the Function Library to access the conditional expressions. This time, click the second option for a multi-conditional statement to insert it into the expression editor.

Insert the first condition to test and action to apply: If a session duration is greater than or equal to 30 minutes, it is "Long".

The second condition to test is that a session lasted between 5 and 30 minutes. In the placeholder c2, type the expression [Duration of Visit] <30 AND [Duration of Visit] >= 5. If a value meets this condition, then it is "Medium".

The last condition that has not been evaluated is that a visit lasts less than five minutes. Because it is the only possible outcome based on our logic, all other visits are "Short".

Increase the number of conditions a statement can evaluate by typing sets of ENDIF and THEN clauses into the Expression Editor. When creating complex statements, it may be helpful to place each clause on a separate line in the Expression Editor to quickly spot gaps in logic or errors in syntax.

After running the workflow, a new column categorizes the length of session visits as short, medium or long.

Switch Function

Like a multi-conditional statement, the Switch function can evaluate multiple criteria. The Switch function runs a list of “cases” or conditions against a column of data and returns the corresponding value. Using the values in the column [Session Category], assign a number 1, 2 or 3 to categories Long, Medium and Short, respectively.

Create a new string column called [Category Code]. Insert the Switch function from the conditional function category. The Switch function requires a few parameters: the column of values to evaluate, a default case, and a series of cases and results, separated by commas. Insert the column [Category] into the placeholder "value". If no specified cases are met, set a default value. In this case, set the default to Null().

The Switch function matches Cases, or an existing value in a column, to a Result. Drag the values that represent the values of cases and results into the function below to complete the expression.

After running the workflow, the values 1, 2 and 3 populate the column [Category Code]. Because every specified case in the Switch function was found, the default value, Null(), was not used.