

# Formula Functions Quick Reference

Download the PDF of this article.

## In this Article

- [Introduction](#)
- [Functions](#)
- [Logic](#)
- [Text and Data](#)
- [Date and Time](#)
- [Math and Trig](#)
- [Statistical](#)
- [Information](#)

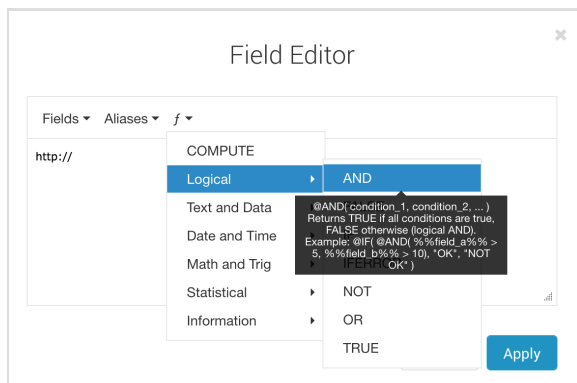
## Related Articles

### Introduction

This page is meant to serve as a quick reference guide for all of FormAssembly's formula functions.

If you are unfamiliar with FormAssembly formulas, please visit our [Smart Processing with Formulas](#) page for an introduction. For practical formula examples, please see our [Formulas for Common Uses](#) page.

You can also view an explanation for each formula function by hovering over an individual function in FormAssembly's [field editor](#).



## Functions

The engine supports most functions found in MS Excel. The function must be spelled in uppercase and start with the @ character.

**Please note:** Formulas cannot be used in connector repeating sections. Javascript calculations cannot be used in connectors. HTML cannot be used in formulas.

Here is a complete list of all FormAssembly functions, broken down by category:

## Logic

<b>@COMPUTE</b>	<pre>@COMPUTE(<i>expression</i>)</pre> <p>Performs arithmetic calculations on form fields.</p> <p>Example:</p> <pre>@COMPUTE(%%tfa_1%%+%%tfa_2%%)</pre>
<b>@AND</b>	<pre>@AND(<i>condition 1,condition 2</i>)</pre> <p>Returns TRUE if both conditions are true, FALSE otherwise (logical AND).</p> <p>Example:</p> <pre>@IF(@AND(%%tfa_1%%&gt;5,%%tfa_2%%&gt;10),"OK","NOT OK")</pre>
<b>@FALSE</b>	<pre>@FALSE()</pre> <p>Returns the logical value FALSE.</p> <div data-bbox="371 1171 1425 1279" style="background-color: #e0f2f7; padding: 5px;"><p>You may receive an error in your formula editor that this argument requires two arguments. It works exactly like @TRUE and does not require two arguments. This is being looked at to be fixed.</p></div> <p>Example:</p> <pre>@IF(@ISNUMBER(%%tfa_3%%)=@FALSE(),"It is not a number","It is a number")</pre> <p>Looks at the field and posts if it is a number or not a number.</p>
<b>@IF</b>	<pre>@IF(<i>condition,when_true,when_false</i>)</pre> <p>Performs a logical test and returns either the second parameter (<i>if true</i>) or the third parameter (<i>if false</i>).</p> <p>Example:</p> <pre>@IF(%%tfa_1%%&gt;5,"GOOD","NOT ENOUGH")</pre>

<p><b>@IFERROR</b></p>	<pre>@IFERROR(<i>value,value_if_error</i>)</pre> <p>Returns a value you specify if a formula evaluates to an error; otherwise, returns the result of the formula. Use the IFERROR function to trap and handle errors in a formula.</p> <p>Example:</p> <pre>@IFERROR(@DATEDIF(%%tfa_1%%,%%tfa_2%%,"D"),@DATEDIF(%%tfa_2%%,%%tfa_2%%,"D"))</pre> <p>Compare two dates to receive a difference. This formula compares first date to the second date and if there is an error, it swaps them and compares second to first.</p>
<p><b>@NOT</b></p>	<pre>@NOT(<i>condition</i>)</pre> <p>Returns TRUE if the condition is false, FALSE otherwise (logical NOT).</p> <p>Example:</p> <pre>@IF(@NOT(%%tfa_1%%&gt;5),"OK","NOT OK")</pre>
<p><b>@OR</b></p>	<pre>@OR(<i>condition 1,condition 2</i>)</pre> <p>Returns TRUE if at least one condition is true, FALSE if all conditions are false (logical OR).</p> <p>Example:</p> <pre>@IF(@OR(%%tfa_1%%&gt;5,%%tfa_2%%&gt;10),when_true,when_false)</pre>
<p><b>@TRUE</b></p>	<pre>@TRUE()</pre> <p>Returns the logical value TRUE.</p>

---

## Text and Data

<p><b>@ADDSLASHES</b></p>	<pre>@ADDSLASHES(%%tfa_XX%%)</pre> <p>Returns the string with backslashes before characters that need to be escaped. These characters are single quote ('), double quote ("), backslash (\) and NULL characters.</p> <p>Example:</p> <pre>@ADDSLASHES(%%tfa_1%%)</pre>
<p><b>@CHAR</b></p>	<pre>@CHAR(<i>number</i>)</pre> <p>Returns the character specified by a number. Number is a number between 1 and 255 specifying which character you want.</p> <p>Example:</p> <pre>@CHAR(65)</pre>
<p><b>@CODE</b></p>	<pre>@CODE(<i>text</i>)</pre> <p>Returns a numeric code for the first character in a text string. The returned code corresponds to the character set used by your computer.</p> <p>Example:</p> <pre>@CODE(%%tfa_1%%)</pre>
<p><b>@CONCATENATE</b></p>	<pre>@CONCATENATE(<i>string,string,...</i>)</pre> <p>Joins 2 or more strings together. Can also be used to send strings and field aliases together.</p> <p>Example:</p> <pre>@CONCATENATE(%%tfa_1%%," ",%%tfa_2%%)</pre>
<p><b>@CONTAINS</b></p>	<pre>@CONTAINS(<i>text,field</i>)</pre> <p>Returns true if the text is inside the field. Otherwise, returns false.</p> <p>Example:</p> <pre>@CONTAINS("@gmail.com",%%tfa_1%%)</pre>

<p><b>@FIND</b></p>	<pre>@FIND(text1,text2,start_position)</pre> <p>Returns the location of a substring of text (text1) in a case-sensitive string of text (text2). Returns #VALUE! if string not found.</p> <p>Example (Returns the domain part of an email address):</p> <pre>@MID(%%tfa_email%%,@COMPUTE(@FIND("@",%%tfa_email%%)+1), @COMPUTE(@FIND(".",%%tfa_email%%)-@FIND("@",%%tfa_email%%)-1))</pre>
<p><b>@LEFT</b></p>	<pre>@LEFT(text,number_of_characters)</pre> <p>Extracts a number of characters from a string, starting from the left.</p> <p>Example:</p> <pre>@LEFT(%%tfa_1%%,5)</pre>
<p><b>@LEN</b></p>	<pre>@LEN(text)</pre> <p>Returns the number of characters in a string.</p>
<p><b>@LOWER</b></p>	<pre>@LOWER(text)</pre> <p>Converts all uppercase letters in a text string to lowercase.</p>
<p><b>@MID</b></p>	<pre>@MID(text,start_position,number_of_characters)</pre> <p>Extracts a number of characters starting at any position.</p>
<p><b>@PROPER</b></p>	<pre>@PROPER(text)</pre> <p>Capitalizes the first letter in a text string and any other letters in text that follow any character other than a letter. Converts all other letters to lowercase letters.</p>
<p><b>@REPT</b></p>	<pre>@REPT(text,number_of_times)</pre> <p>Repeats text a given number of times. Use REPT to return a number of instances of a text string.</p>

<p><b>@RIGHT</b></p>	<pre>@RIGHT(<i>text,number_of_characters</i>)</pre> <p>Extracts a number of characters from a string, starting from the right.</p> <p>Example:</p> <pre>@RIGHT(%%tfa_1%%,5)</pre>
<p><b>@SEARCH</b></p>	<pre>@SEARCH(<i>find_text,within_text,start_num</i>)</pre> <p>Returns the number of the character at which a specific character or text string is first found, beginning with start_num (case-insensitive). Start_num is 1 by default.</p>
<p><b>@SUBSTITUTE</b></p>	<pre>@SUBSTITUTE(<i>base_string,match,match_replacement</i>)</pre> <p>Replaces all occurrences of match with match_replacement in base_string.</p>
<p><b>@TRIM</b></p>	<pre>@TRIM(<i>text</i>)</pre> <p>Strip whitespace from the beginning and end of a string.</p> <p>Example:</p> <pre>@TRIM(%%tfa_1%%)</pre>
<p><b>@UPPER</b></p>	<pre>@UPPER(<i>text</i>)</pre> <p>Converts text to uppercase.</p> <p>Example:</p> <pre>@UPPER(%%tfa_1%%)</pre>
<p><b>@URLENCODE</b></p>	<pre>@URLENCODE(<i>query_string_param</i>)</pre> <p>Returns the RFC3986-encoded version of the string passed in.</p> <p>Example:</p> <pre>http://www.tfaforms.com/123?tfa_Name=@URLENCODE("Mike Johnson") will output http://www.tfaforms.com/123?tfa_Name=Mike%20Johnson</pre>

<p><b>@URLDECODE</b></p>	<pre>@URLDECODE(<i>query_string_param</i>)</pre> <p>Decodes RFC3986-encoded version of the string passed in.</p> <p>Example:</p> <pre>http://www.tfaforms.com/123?tfa_Name=@URLDECODE("Mike%20Johnson") will output http://www.tfaforms.com/123?tfa_Name=Mike Johnson</pre>
<p><b>@YMDTODAY</b></p>	<pre>@YMDTODAY(<i>offset,format</i>)</pre> <p>Returns the current date (plus optional offset) in ISO8601/Salesforce Date format. Offset is date offset, e.g. "+1 days". Can be null. <a href="#">More information available here.</a></p> <p>Output will include time and be in the following format: 2017-09-05-04:00</p>
<p><b>@YMDNOW</b></p>	<pre>@YMDNOW(<i>offset,format</i>)</pre> <p>Returns the current date and time ( plus optional offset ) in ISO8601/Salesforce DateTime format. Can be set to Salesforce Date format with YMDNOW("", "Y-m-d"). <a href="#">More information available here.</a></p> <p>Output will include the time and be in the following format: 2017-09-05T17:16:30-04:00</p>

---

## Date and Time

<p><b>@DATEDIF</b></p>	<pre>@DATEDIF(<i>start_date,end_date,unit</i>)</pre> <p>Calculates the number of days, months, or years between two dates. Unit - the type of information that you want returned: "Y" - the number of complete years, "M" - months, "D" - days, "YD" - the difference between the days of start_date and end_date (the years of the dates are ignored), "MD" - the difference between the days in start_date and end_date (the months and years of the dates are ignored).</p> <p>Dates may be entered as text strings within quotation marks ("2001/1/30"), as serial numbers (36921 = January 30, 2001), or as the results of other formulas or functions (for example, DATEVALUE("2001/1/30")).</p>
------------------------	--

<p><b>@DATEVALUE</b></p>	<div data-bbox="424 165 1259 250" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@DATEVALUE(date_text)</p> </div> <p>Returns the serial number of the date represented by date_text. Use DATEVALUE to convert a date represented by text to a serial number.</p> <p>Example:</p> <div data-bbox="424 416 1259 519" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@DATEVALUE("8/22/2008") returns serial number of the text date = 39682.</p> </div>
<p><b>@DAY</b></p>	<div data-bbox="424 620 1259 705" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@DAY(serial_number)</p> </div> <p>Returns the day of a date, represented by a serial number. The day is given as an integer ranging from 1 to 31.</p>
<p><b>@HOUR</b></p>	<div data-bbox="424 828 1259 913" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@HOUR(serial_number)</p> </div> <p>Returns only the hour of a time value. The hour is given as an integer, ranging from 0 (12:00 A.M.) to 23 (11:00 P.M.).</p>
<p><b>@LOCALTODAY</b></p>	<div data-bbox="424 1037 1259 1122" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@LOCALTODAY()</p> </div> <p>Returns the current date according to your language and time-zone settings.</p>
<p><b>@LOCALNOW</b></p>	<div data-bbox="424 1209 1259 1294" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@LOCALNOW()</p> </div> <p>Returns the current date and time according to your language and time-zone settings.</p>
<p><b>@MINUTE</b></p>	<div data-bbox="424 1382 1259 1467" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@MINUTE(serial_number)</p> </div> <p>Returns the minutes of a time value. The minute is given as an integer, ranging from 0 to 59.</p>
<p><b>@MONTH</b></p>	<div data-bbox="424 1554 1259 1639" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@MONTH(serial_number)</p> </div> <p>Returns the month of a date represented by a serial number. The month is given as an integer, ranging from 1 (January) to 12 (December).</p>
<p><b>@NOW</b></p>	<div data-bbox="424 1762 1259 1848" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>@NOW()</p> </div> <p>Returns the current date as a timestamp. This can be passed to other date functions to extract the day, month or year.</p>



<p><b>@SECOND</b></p>	<div data-bbox="424 165 1246 250" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@SECOND(<i>serial_number</i>)</code></p> </div> <p>Returns the seconds of a time value. The second is given as an integer in the range 0 (zero) to 59.</p>
<p><b>@TIME</b></p>	<div data-bbox="424 456 1246 542" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@TIME(<i>hour,minute,second</i>)</code></p> </div> <p>Returns the decimal number for a particular time, ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 AM) to 23:59:59 (11:59:59 P.M.).</p>
<p><b>@TIMEVALUE</b></p>	<div data-bbox="424 663 1246 748" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@TIMEVALUE(<i>time_text</i>)</code></p> </div> <p>Returns the decimal number of the time represented by a text string. The decimal number is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).</p>
<p><b>@TODAY</b></p>	<div data-bbox="424 904 1246 990" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@TODAY()</code></p> </div> <p>Returns the serial number of the current date. The serial number is the date-time code used for date and time calculations. You can use the TODAY function only as a default value; you cannot use it in a calculated column.</p>
<p><b>@WEEKDAY</b></p>	<div data-bbox="424 1146 1246 1232" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@WEEKDAY(<i>serial_number,return_type</i>)</code></p> </div> <p>Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday), by default. <i>return_type</i> is a number that determines the type of return value: 1 - numbers 1 (Sunday) through 7 (Saturday), 2 - numbers 1 (Monday) through 7 (Sunday), 3 - numbers 0 (Monday) through 6 (Sunday).</p> <p>Example:</p> <div data-bbox="424 1464 1246 1581" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@WEEKDAY("2/14/2008",2)</code> returns day of the week, with numbers 1 (Monday) through 7 (Sunday) = 4.</p> </div>
<p><b>@YEAR</b></p>	<div data-bbox="424 1635 1246 1720" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><code>@YEAR(<i>serial_number</i>)</code></p> </div> <p>Returns the year value of a date represented by a serial number. The year is given as a 4-digit integer.</p>

**Note:** FormAssembly formulas honor the same date format parameters as PHP. More information [can be found here](#).

## Math and Trig

<b>@ABS</b>	<pre>@ABS(<i>number</i>)</pre> <p>Returns the absolute value of a number.</p>
<b>@CEILING</b>	<pre>@CEILING(<i>number</i>,<i>significance</i>)</pre> <p>Returns number rounded up, away from zero, to the nearest multiple of significance.</p> <p>Example:</p> <pre>@CEILING(2.5,1) This will return 3</pre>
<b>@DEGREES</b>	<pre>@DEGREES(<i>angle</i>)</pre> <p>Converts radians into degrees.</p>
<b>@EVEN</b>	<pre>@EVEN(<i>number</i>)</pre> <p>Returns number rounded up to the nearest even integer.</p> <p>Example:</p> <pre>@EVEN(1.5) This will return 2</pre>
<b>@FLOOR</b>	<pre>@FLOOR(<i>number</i>,<i>significance</i>)</pre> <p>Rounds number down, toward zero, to the nearest multiple of significance. Significance is the multiple to which you want to round.</p> <p>Example:</p> <pre>@FLOOR(2.5,1) This will return 2</pre>

<b>@INT</b>	<pre>@INT(<i>number</i>)</pre> <p>Rounds a number down to the nearest integer.</p> <p>Example:</p> <pre>@INT(8.9) This will return 8</pre>
<b>@MROUND</b>	<pre>@MROUND(<i>number,multiple</i>)</pre> <p>Returns a number rounded to the desired multiple.</p> <p>Example:</p> <pre>@MROUND(10,3) This will round 10 to a nearest multiple of 3, which is 9</pre>
<b>@ODD</b>	<pre>@ODD(<i>number</i>)</pre> <p>Returns number rounded up to the nearest odd integer.</p> <p>Example:</p> <pre>@ODD(1.5) This will round 1.5 up to the nearest odd integer, 3</pre>
<b>@PI</b>	<pre>@PI()</pre> <p>Returns the number 3.14159265358979, the mathematical constant pi, accurate to 15 digits.</p>
<b>@POWER</b>	<pre>@POWER(<i>number,power</i>)</pre> <p>Returns the result of a number raised to a power.</p> <p>Example:</p> <pre>@POWER(5,2) This will return 5 squared = 25</pre>

<p><b>@PRODUCT</b></p>	<pre>@PRODUCT(<i>number1,number2,...</i>)</pre> <p>Multiplies all the numbers given as arguments and returns the product.</p>
<p><b>@QUOTIENT</b></p>	<pre>@QUOTIENT(<i>numerator,denominator</i>)</pre> <p>Returns the integer portion of a division. Use this function when you want to discard the remainder of a division.</p>
<p><b>@RADIANS</b></p>	<pre>@RADIANS(<i>angle</i>)</pre> <p>Converts degrees to radians.</p>
<p><b>@RAND</b></p>	<pre>@RAND()</pre> <p>Returns a random number greater than or equal to 0 and less than 1. To generate a random real number between A and B, you can try custom code similar to <code>&lt;code&gt;RAND()*(b-a)+a&lt;/code&gt;</code>.</p> <p>Example:</p> <pre>@RAND()*100</pre> <p>Returns a random number greater or equal to 0 but less than 100</p>
<p><b>@RANDBETWEEN</b></p>	<pre>@RANDBETWEEN(<i>bottom,top</i>)</pre> <p>Returns a random number between the numbers you specify.</p> <p>Example:</p> <pre>@RANDBETWEEN(1,100)</pre> <p>Returns a random number between 1 and 100</p>
<p><b>@ROUND</b></p>	<pre>@ROUND(<i>number,decimal_places</i>)</pre> <p>Returns a number rounded to a specified number of decimal places.</p>

**@ROUNDDOWN**

```
@ROUNDDOWN(number,num_digits)
```

Rounds a number down, toward zero.

Example:

```
@ROUNDDOWN(3.2,0)  
Rounds 3.2 down to zero decimal places = 3
```

**@ROUNDUP**

```
@ROUNDUP(number,num_digits)
```

Rounds a number up, away from zero.

Example:

```
@ROUNDUP(3.2,0)  
Rounds 3.2 up to zero decimal places = 4
```

**@SIGN**

```
@SIGN(number)
```

Determines the sign of a number. Returns 1 if the number is positive, zero (0) if the number is 0, and -1 if the number is negative.

Example:

```
@SIGN(10)  
Returns sign of a positive number = 1
```

**@SQRT**

```
@SQRT(number)
```

Returns a positive square root.

**@SUM**

```
@SUM(number1,number2,... )
```

Adds all the numbers that you specify as arguments.

Example:

```
@SUM(%%tfa_1%%,%%tfa_2%%)
```

<b>@TRUNC</b>	<pre>@TRUNC(<i>number,num_digits</i>)</pre> <p>Truncates a number to an integer by removing the fractional part of the number. Num_digits is a number specifying the precision of the truncation.</p> <p>Example:</p> <pre>@TRUNC(8.9)</pre> <p>Returns integer part of 8.9 = 8</p>
---------------	---

## Statistical

<b>@MAX</b>	<pre>@MAX(<i>number1,number2,...</i>)</pre> <p>Returns the largest value from the numbers provided.</p>
<b>@MIN</b>	<pre>@MIN(<i>number1,number2,...</i>)</pre> <p>Returns the smallest value from the numbers provided.</p>

## Information

<b>@ISBLANK</b>	<pre>@ISBLANK(<i>value</i>)</pre> <p>Returns the logical value TRUE if value is empty; otherwise, it returns FALSE.</p>
<b>@ISERR</b>	<pre>@ISERR(<i>value</i>)</pre> <p>Returns the logical value TRUE if value is any error value except #N/A; otherwise, it returns FALSE.</p>
<b>@ISERROR</b>	<pre>@ISERROR(<i>value</i>)</pre> <p>Returns the logical value TRUE if value is any error value, such as #N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!; otherwise, it returns FALSE.</p>

<b>@ISEVEN</b>	<code>@ISEVEN(<i>number</i>)</code> Returns TRUE if number is even, or FALSE if number is odd.
<b>@ISNA</b>	<code>@ISNA(<i>value</i>)</code> Returns the logical value TRUE if value is #N/A (value not available) error value; otherwise, it returns FALSE.
<b>@ISNUMBER</b>	<code>@ISNUMBER(<i>value</i>)</code> Returns the logical value TRUE if value is a number; otherwise, it returns FALSE.
<b>@ISODD</b>	<code>@ISODD(<i>number</i>)</code> Returns TRUE if number is odd, or FALSE if number is even.
<b>@NA</b>	<code>@NA()</code> Returns the error value #N/A. #N/A is the error value that means "no value is available."

---