### Syntax

The general syntax of a [formula](formula.docx) is as follows:

formula:
expression

expression:
( expression )
constant
prefix-operator expression
expression infix-operator expression
expression postfix-operator
cell-reference
function-call
name

where expression is an arbitrarily complex expression involving constants (§), operators (§), [cell](cell.docx) references (§), calls to functions (§), and names (§).

A token is the minimal lexical element of a formula. The categories of tokens are: constants (except for array-constant), operators, [cell](cell.docx) [references](references.docx), function names, names, and punctuators. The punctuators are:

* Left parenthesis (() and right parenthesis ()) used for expression grouping and in a function call.
* Comma (,) used in a function call and an array-constant.
* Left brace ({), right brace (}), and semicolon (;) used in an array-constant.

In a [formula](formula.docx), an arbitrary number of space characters (U+0020) can precede the first token or follow the final token. An arbitrary number of space characters can separate two adjacent tokens, except that no space characters shall separate a function-name from the left parenthesis (() that follows it. Such space characters have no effect on the semantics of a formula; however, such spaces shall be distinguished from the space operator (§).

All arithmetic terms in an expression are real numbers. [Example: In the expression 1/3, although the operands appear to be integers, they are, in fact, real numbers, and the result is 0.33…, not 0, as would result from integer division. end example]

As ranges of data are fundamental to spreadsheet calculations, many SpreadsheetML functions are able to take arrays as inputs and to return arrays as outputs. The way in which [formula](formula.docx) return values are returned into the [worksheet](worksheet.docx) differs based on whether the [formula](formula.docx) in a given [cell](cell.docx) was array entered or not, but intermediate calculations are always done for the full arrays when they are used as inputs. The differences with an array-entered [formula](formula.docx) are:

* No implicit intersection is performed on [cell](cell.docx) range inputs.
* The results of the formula’s calculation can be returned across multiple cells in the sheet.

When a range of [cell](cell.docx) [references](references.docx) is used in a [formula](formula.docx) that is array-entered in an area larger than that range, the excess cells take on a value of #N/A.

[Example: Here are some formulas taking array constants and ranges:

* (B2:B4\*C2:C4)+10.5 performs three calculations: (B2\*C2)+10.5, (B3\*C3)+10.5, and (B4\*C4)+10.5.
* SQRT({1,2,3,4}) returns 1 when entered normally.
* SQRT({1,2,3,4}) returns 1 when array-entered into a single [cell](cell.docx), but if it’s array-entered in four or more cells in a contiguous [row](row.docx), it will return 1, 1.41, 1.73, and 2 in the first four cells, respectively, and #N/A in any additional cells in the [horizontal](horizontal.docx) range for which it was array-entered. (For display purposes, the values returned have been truncated to two decimal places.)
* SUM(SQRT({1,2,3,4})) returns 6.14 when entered normally, since array calculations are always performed by the [SQRT](SQRT.docx) function, and the array output is understood as a valid input by the [SUM](SUM.docx) function.

With A1:A4 holding the values 1, 2, 3, and 4, respectively:

* SQRT(A1:A4) entered normally will do implicit intersection if it is in any of the rows 1–4, and return the [SQRT](SQRT.docx) of the number in the same row.
* SQRT(A1:A4) returns 1 when array-entered into a single [cell](cell.docx), since it does not do implicit intersection in this case. If it’s array-entered in multiple cells in a contiguous column, it will return 1, 1.41, 1.73, 2, #N/A, …, respectively, in the cells in its [vertical](vertical.docx) output range. end example]