#### Functions

A function is a named [formula](formula.docx) that takes zero or more arguments, performs an operation, and, optionally, returns a result. A function call has the following form:

function-call:
function-name ( [ argument-list ] )

function-name:
predefined-function-name
user-defined-function-name

predefined-function-name:
[ABS](ABS.docx) | [ACOS](ACOS.docx) | [ACOSH](ACOSH.docx) | any of the other functions defined in §

user-defined-function-name:
letter [ user-defined-name-characters ]

user-defined-name-characters:
user-defined-name-characters user-defined-name-character

user-defined-name-character:
letter
any decimal digit 0–9
.

letter:
any Latin letter A–Z, a–z

argument-list:
argument
argument-list , argument

argument:
expression

predefined-function-names and user-defined-function-names are not case-sensitive.

A user-defined-function-name shall not have any of the following forms:

* [TRUE](TRUE.docx) or [FALSE](FALSE.docx)
* [name](name.docx)
* cell-reference

[Guidance: An implementation is encouraged to support user-defined-function-names at least as long as 255 characters. end guidance]

The semantics of a call to a function having a user-defined-function-name are unspecified.

[Example: Here are some function calls: PI(), POWER(A1,B3), and SUM(C6:C10). end example]

An argument to a function can be a call to a function. That is, function calls can nest. [Guidance An implementation is encouraged to support at least 64 levels of nested function calls. end guidance]

Some functions take a variable number of arguments. This is indicated in the [Syntax](Syntax.docx) sections of § by their having argument-list as all, or the trailing part, of their argument list. The total number of arguments that shall be passed to such functions is at least 1.

[Guidance An implementation is encouraged to support function calls having at least 255 arguments. end guidance]

Expressions can have one or more values. Scalar expressions designate a single value, and [cell](cell.docx) [references](references.docx) and array constants can designate multiple values. In the case of a multi-value expression, the way in which this is handled by a function when passed as an argument depends on a number of factors.

Most functions and operators expect either single- or multi-valued arguments and perform all of the array calculations whenever multi-valued arguments are present. [Example: SQRT({1;2;3;4}); see the examples in §. end example]

When a function expects a single-valued argument but a multi-valued expression is passed, an attempt can be made to convert that set of values to a single value. For an array value or constant, the value of the expression is the value of the first element within that array value or constant. For a [cell](cell.docx) range, the first element can be used, or implicit intersection can be performed—the exact behavior is unspecified.

When a function expects a multi-valued argument but a single-valued expression is passed, that single-valued argument is treated as a 1x1 array.

For rules on how deal with potential ambiguities between function names and defined names, see §.