|  |  |
| --- | --- |
| **Variables** | |
| **cell** | The current cell. It has a few fields: 'value' and 'recon'. |
| **value** | The current cell's value. This is a shortcut for 'cell.value'. |
| **row** | The current row. It has 5 fields: 'flagged', 'starred', 'index', 'cells', and 'record'. |
| **cells** | The cells of the current row. This is a shortcut for 'row.cells'. A particular cell can be retrieved with 'cells.<column name>' if the <column name> is a single word, or with 'cells["<column name>"] otherwise. |
| **rowIndex** | The current row's index. This is a shortcut for 'row.index'. |
| **Shortcuts [ / \x/ ]** | |
| **/ \d /** | Any number [0-9] |
| **/ \D /** | Any character other than a digit (equal to [^0-9]) |
| **/ \s /** | Any whitespace character (space, tab, new line, carriage return) |
| **/ \S /** | Any character other than whitespace characters |
| **/ \w /** | Any character that can be part of a word |
| **/ \W/** | Any non-word character |
| **/ . /** | Any single character |
| **Functions** | |
| **abs** | (number d) |
| returns: number |
| Returns the absolute value of a number |
| **acos** | (number d) |
| returns: number |
| Returns the arc cosine of an angle, in the range 0 through PI |
| **and** | (boolean a, boolean b) |
| returns: boolean |
| ANDs two boolean values |
| **asin** | (number d) |
| returns: number |
| Returns the arc sine of an angle in the range of -PI/2 through PI/2 |
| **atan** | (number d) |
| returns: number |
| Returns the arc tangent of an angle in the range of -PI/2 through PI/2 |
| **atan2** | (number x, number y) |
| returns: number theta |
| Converts rectangular coordinates (x, y) to polar (r, theta) |
| **ceil** | (number d) |
| returns: number |
| Returns the ceiling of a number |
| **chomp** | (string str, string separator) |
| returns: string |
| Removes separator from the end of str if it's there, otherwise leave it alone. |
| **combin** | (number d) |
| returns: number |
| Returns the number of combinations for n elements as divided into k |

|  |  |
| --- | --- |
| **contains** | (string s, string frag) |
| returns: boolean |
| Returns whether s contains frag |
| **cos** | (number d) |
| returns: number |
| Returns the trigonometric cosine of an angle |
| **cosh** | (number d) |
| returns: number |
| Returns the hyperbolic cosine of a value |
| **cross** | (cell c, string projectName, string columnName) |
| returns: array |
| TODO |
| **datePart** | (date d, string part) |
| returns: date |
| Returns part of a date |
| **degrees** | (number d) |
| returns: number |
| Converts an angle from radians to degrees. |
| **diff** | (o1, o2, time unit (optional)) |
| returns: string for strings, number for dates |
| For strings, returns the portion where they differ. For dates, it returns the difference in given time units |
| **endsWith** | (string s, string sub) |
| returns: boolean |
| Returns whether s ends with sub |
| **escape** | (string s, string mode ['html','xml','csv','url','javascript']) |
| returns: string |
| Escapes a string depending on the given escaping mode. |
| **even** | (number d) |
| returns: number |
| Rounds the number up to the nearest even integer |
| **exp** | (number n) |
| returns: number |
| Returns e^n |
| **facetCount** | (choiceValue, string facetExpression, string columnName) |
| returns: number |
| Returns the facet count corresponding to the given choice value |
| **fact** | (number i) |
| returns: number |
| Returns the factorial of a number |
| **factn** | (number i) |
| returns: number |
| Returns the factorial of a number |
| **fingerprint** | (string s) |
| returns: string |
| Returns the fingerprint of s, a derived string that aims to be a more canonical form of it (this is mostly useful for finding clusters of strings related to the same information). |
| **floor** | (number d) |
| returns: number |
| Returns the floor of a number |
| **gcd** | (number d, number e) |
| returns: number |
| Returns the greatest common denominator of the two numbers |

|  |  |
| --- | --- |
| **get** | (o, number or string from, optional number to) |
| returns: Depends on actual arguments |
| If o has fields, returns the field named 'from' of o. If o is an array, returns o[from, to]. if o is a string, returns o.substring(from, to) |
| **hasField** | (o, string name) |
| returns: boolean |
| Returns whether o has field name |
| **htmlAttr** | (Element e, String s) |
| returns: String attribute Value |
| Selects a value from an attribute on an Html Element |
| **htmlText** | (Element e) |
| returns: String text |
| Selects the text from within an element (including all child elements) |
| **inc** | (date d, number value, string unit (default to 'hour')) |
| returns: date |
| Returns a date changed by the given amount in the given unit of time |
| **indexOf** | (string s, string sub) |
| returns: number |
| Returns the index of sub first ocurring in s |
| **innerHtml** | (Element e) |
| returns: String innerHtml |
| The innerHtml of an HTML element |
| **join** | (array a, string sep) |
| returns: string |
| Returns the string obtained by joining the array a with the separator sep |
| **jsonize** | (value) |
| returns: JSON literal value |
| Quotes a value as a JSON literal value |
| **lastIndexOf** | (string s, string sub) |
| returns: number |
| Returns the index of sub last ocurring in s |
| **lcm** | (number d, number e) |
| returns: number |
| Returns the greatest common denominator of the two numbers |
| **length** | (array or string o) |
| returns: number |
| Returns the length of o |
| **ln** | (number n) |
| returns: number |
| Returns the natural log of n |
| **log** | (number n) |
| returns: number |
| Returns the base 10 log of n |
| **match** | (string or regexp) |
| returns: array of strings |
| Returns an array of the groups matching the given regular expression |
| **max** | (number a, number b) |
| returns: number |
| Returns the greater of two numbers |
| **md5** | (string s) |
| returns: string |
| Returns the MD5 hash of s |

|  |  |
| --- | --- |
| **min** | (number a, number b) |
| returns: number |
| Returns the smaller of two numbers |
| **mod** | (number a, number b) |
| returns: number |
| Returns a modulus b |
| **mqlKeyQuote** | (string s) |
| returns: string |
| Unquotes a MQL key |
| **mqlKeyUnquote** | (string s) |
| returns: string |
| Quotes a string into a MQL key |
| **multinomial** | (one or more numbers) |
| returns: number |
| Calculates the multinomial of a series of numbers |
| **ngram** | (string s, number n) |
| returns: array of strings |
| Returns an array of the word ngrams of s |
| **ngramFingerprint** | (string s, number n) |
| returns: string |
| Returns the n-gram fingerprint of s |
| **not** | (boolean b) |
| returns: boolean |
| Returns the opposite of b |
| **now** | (undefined) |
| returns: date |
| Returns the current time |
| **odd** | (number d) |
| returns: number |
| Rounds the number up to the nearest even integer |
| **or** | (boolean a, boolean b) |
| returns: boolean |
| Returns a OR b |
| **ownText** | (Element e) |
| returns: String ownText |
| Gets the text owned by this HTML element only; does not get the combined text of all children. |
| **parseHtml** | (string s) |
| returns: HTML object |
| Parses a string as HTML |
| **parseJson** | (string s) |
| returns: JSON object |
| Parses a string as JSON |
| **partition** | (string s, string or regex frag, optional boolean omitFragment) |
| returns: array |
| Returns an array of strings [a,frag,b] where a is the string part before the first occurrence of frag in s and b is what's left. If omitFragment is true, frag is not returned. |
| **phonetic** | (string s, string encoding (optional, defaults to 'metaphone3')) |
| returns: string |
| Returns the a phonetic encoding of s (optionally indicating which encoding to use') |
| **pow** | (number a, number b) |
| returns: number |
| Returns a^b |
| **quotient** | (number numerator, number denominator) |
| returns: number |
| Returns the integer portion of a division |
| **radians** | (number d) |
| returns: number |
| Converts an angle in degrees to radians |
| **reinterpret** | (string s, string encoder) |
| returns: string |
| Returns s reinterpreted thru the given encoder. |
| **replace** | (string s, string or regex f, string r) |
| returns: string |
| Returns the string obtained by replacing f with r in s |
| **replaceChars** | (string s, string f, string r) |
| returns: string |
| Returns the string obtained by replacing all chars in f with the char in s at that same position |
| **reverse** | (array a) |
| returns: array |
| Reverses array a |
| **round** | (number n) |
| returns: number |
| Returns n rounded |
| **rpartition** | (string s, string or regex frag, optional boolean omitFragment) |
| returns: array |
| Returns an array of strings [a,frag,b] where a is the string part before the last occurrence of frag in s and b is what's left. If omitFragment is true, frag is not returned. |
| **select** | (Element e, String s) |
| returns: HTML Elements |
| Selects an element from an HTML elementn using selector syntax |
| **sha1** | (string s) |
| returns: string |
| Returns the SHA-1 hash of s |
| **sin** | (number d) |
| returns: number |
| Returns the trigonometric sine of an angle |
| **sinh** | (number d) |
| returns: number |
| Returns the hyperbolic sine of an angle |
| **slice** | (o, number from, optional number to) |
| returns: Depends on actual arguments |
| If o is an array, returns o[from, to]. if o is a string, returns o.substring(from, to) |
| **smartSplit** | (string s, optional string sep) |
| returns: array |
| Returns the array of strings obtained by splitting s with separator sep. Handles quotes properly. Guesses tab or comma separator if "sep" is not given. |
| **sort** | (array a) |
| returns: array |
| Sorts array a |
| **split** | (string s, string or regex sep, optional boolean preserveAllTokens) |
| returns: array |
| Returns the array of strings obtained by splitting s with separator sep. If preserveAllTokens is true, then empty segments are preserved. |

|  |  |
| --- | --- |
| **splitByCharType** | (string s) |
| returns: array |
| Returns the array of strings obtained by splitting s into substrings with the given lengths |
| **startsWith** | (string s, string sub) |
| returns: boolean |
| Returns whether s starts with sub |
| **strip** | (string s) |
| returns: string |
| Returns copy of the string, with leading and trailing whitespace omitted. |
| **substring** | (o, number from, optional number to) |
| returns: Depends on actual arguments |
| If o is an array, returns o[from, to]. if o is a string, returns o.substring(from, to) |
| **sum** | (array a) |
| returns: number |
| Sums numbers in array a |
| **tan** | (number d) |
| returns: number |
| Returns the trigonometric tangent of an angle |
| **tanh** | (number d) |
| returns: number |
| Returns the hyperbolic tangent of a value |
| **toDate** | (o, boolean month\_first / format1, format2, ... (all optional)) |
| returns: date |
| Returns o converted to a date object, you can hint if the day or the month is listed first, or give an ordered list of possible formats using this syntax: http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html |
| **toLowercase** | (string s) |
| returns: string |
| Returns s converted to lowercase |
| **toNumber** | (o) |
| returns: number |
| Returns o converted to a number |
| **toString** | (o, string format (optional)) |
| returns: string |
| Returns o converted to a string |
| **toTitlecase** | (string s) |
| returns: string |
| Returns s converted to titlecase |
| **toUppercase** | (string s) |
| returns: string |
| Returns s converted to uppercase |
| **trim** | (string s) |
| returns: string |
| Returns copy of the string, with leading and trailing whitespace omitted. |
| **type** | (object o) |
| returns: string |
| Returns the type of o |
| **unescape** | (string s, string mode ['html','xml','csv','url','javascript']) |
| returns: string |
| Unescapes all escaped parts of the string depending on the given escaping mode. |
| **unicode** | (string s) |
| returns: string |
| Returns an array of strings describing each character of s in their full unicode notation |

|  |  |
| --- | --- |
| **unicodeType** | (string s) |
| returns: string |
| Returns an array of strings describing each character of s in their full unicode notation |
| **uniques** | (array a) |
| returns: array |
| Returns array a with duplicates removed |
| **urlify** | (string s) |
| returns: string |
| replaces spaces with underscore |

|  |  |
| --- | --- |
| **Controls** | |
| **filter** | (expression a, variable v, expression test) |
| returns: array |
| Evaluates expression a to an array. Then for each array element, binds its value to variable name v, evaluates expression test which should return a boolean. If the boolean is true, pushes v onto the result array. |
| **forEach** | (expression a, variable v, expression e) |
| returns: array |
| Evaluates expression a to an array. Then for each array element, binds its value to variable name v, evaluates expression e, and pushes the result onto the result array. |
| **forEachIndex** | (expression a, variable i, variable v, expression e) |
| returns: array |
| Evaluates expression a to an array. Then for each array element, binds its index to variable i and its value to variable name v, evaluates expression e, and pushes the result onto the result array. |
| **forNonBlank** | (expression o, variable v, expression eNonBlank, expression eBlank) |
| returns: Depends on actual arguments |
| Evaluates expression o. If it is non-blank, binds its value to variable name v, evaluates expression eNonBlank and returns the result. Otherwise (if o evaluates to blank), evaluates expression eBlank and returns that result instead. |
| **forRange** | (number from, number to, number step, variable v, expression e) |
| returns: array |
| Iterates over the variable v starting at "from", incrementing by "step" each time while less than "to". At each iteration, evaluates expression e, and pushes the result onto the result array. |
| **if** | (expression o, expression eTrue, expression eFalse) |
| returns: Depends on actual arguments |
| Evaluates expression o. If it is true, evaluates expression eTrue and returns the result. Otherwise, evaluates expression eFalse and returns that result instead. |
| **isBlank** | (expression o) |
| returns: boolean |
| Returns whether o is null or an empty string |
| **isError** | (expression o) |
| returns: boolean |
| Returns whether o is an error |
| **isNonBlank** | (expression o) |
| returns: boolean |
| Returns whether o is not null and not an empty string |
| **isNotNull** | (expression o) |
| returns: boolean |
| Returns whether o is not null |
| **isNull** | (expression o) |
| returns: boolean |
| Returns whether o is null |
| **isNumeric** | (expression o) |
| returns: boolean |
| Returns whether o can represent a number |
| **with** | (expression o, variable v, expression e) |
| returns: Depends on actual arguments |
| Evaluates expression o and binds its value to variable name v. Then evaluates expression e and returns that result |

**CartoDB**

Ayuda CSS: [*http://www.w3schools.com/css/*](http://www.w3schools.com/css/)

Colores CSS: [*http://www.w3schools.com/cssref/css\_colors.asp*](http://www.w3schools.com/cssref/css_colors.asp)

Coordenadas geográficas: [*http://www.gps-coordinates.net/*](http://www.gps-coordinates.net/)

|  |  |
| --- | --- |
| **Atajos con Excel** | |
| **Limpiar** | Elimina todos los caracteres no imprimibles del texto. |
| =LIMPIAR(celda) |
| =CLEAN(cell) |
| **Espacios** | Elimina espacios sobrantes. |
| =ESPACIOS(celda) |
| =TRIM(cell) |
| **Sustituir** | Sustituye texto existente por una nueva sentencia. |
| =SUSTITUIR(celda,texto\_antiguo,texto\_nuevo) |
| =SUBSTITUTE(cell, old\_text,new\_text) |
| **Concatenar** | Une varias cadenas de texto en una sola. |
| =CONCATENAR(texto1,texto2,…) |
| =CONCATENATE(text1,text2,…) |
| **Si** | Comprueba si se cumple una condición dada, y devuelve “VERDAD” o “FALSO”. |
| =SI(condición,valor\_si\_cierto,valor\_si\_falso) |
| =IF(condition,value\_if\_right,value\_if\_false) |
| **Fecha a texto** | Convierte un valor a texto en el format especificado. |
| =TEXTO(celda,”formato”) |
| =TEXT(cell,”format”) |

Caracteres UNICODE: <https://en.wikipedia.org/wiki/List_of_Unicode_characters>

Caracteres UNICODE (otra tabla): <http://unicode-table.com/es/>

Caracteres ASCII: <http://www.theasciicode.com.ar/ascii-control-characters/unit-separator-ascii-code-31.html>