

in

Last Modified on 25/05/2020 1:43 pm AEST

Returns true if some calculated value is equal to one of several options.

Keyword

```
search in (options)
```

```
search not in (options)
```

Arguments

Argument	Data Type
----------	-----------

search	any
---------------	-----

input	comma-separated list of single-value calculations
--------------	---

result	bool (true/false)
---------------	-------------------

Comments

The **in** keyword tests to see if the search calculation matches any of several possible options. It returns true if a match is found, and false if no match is found. The *search* and *input* will be converted to a common data type if possible.

The search input may be a list or single-value, and the result will be a bool (true/false) or list of bools respectively. The individual items specified in the list are typically constant values. They may also be calculations, but each must individually only return a single-value result.

Not in

The **not** keyword may be placed directly before the **in** keyword. This causes the opposite result to be returned, except if a null is provided (in which case false is still returned).

Case-insensitive

String comparisons are case insensitive for the **in** keyword, and for the ReadNow platform in general.

Nulls

Consistent with the way that nulls are treated in calculations (where a null value represents the idea that the value is unknown or unknowable):

1. If the search value is null, then an **in** calculation will return false.
2. If the search value is null, then a **not in** calculation will also return false.

Examples

```
[Project].[Project Tasks] in ('Documentation', 'Unit tests')
```

```
'A+' in (85, '79%', 'B+') /* all values are converted to the string data type to be a common type */
```

```
'readinow' in ('ReadiNow') /* returns true - string comparisons are case insensitive */
```

```
1 in (1, 2, 3) /* returns true */  
4 in (1, 2, 3) /* returns false */  
null in (1, 2, 3) /* returns false */  
null in (1, null) /* returns false, but may be reviewed in a future release */
```

```
1 not in (1, 2, 3) /* returns false */  
4 not in (1, 2, 3) /* returns true */  
null not in (1, 2, 3) /* returns false */  
null not in (1, null) /* returns false */
```

```
x in (  
    'Options can be',  
    'spread over',  
    'multiple lines'  
)
```

```
x in (  
    'Options' + 'can' + 'be',  
    [based on] + 'calculations'  
)
```