

# RAND Function

## Contents:

- *Basic Usage*
  - *Syntax and Arguments*
    - *int\_value*
  - *Examples*
    - *Example - Random values*
    - *Example - Type check functions*
- 

The `RAND` function generates a random real number between 0 and 1. The function accepts an optional integer parameter, which causes the same set of random numbers to be generated with each job execution.

- This function generates values of Decimal type with fifteen digits of precision after the decimal point. If you want to see all digits in the generated value, you might need to apply a different number format. See *NUMFORMAT Function*.
- New random numbers are generated within the browser, after each browser refresh, and between subsequent job executions.

Optionally, you can insert an integer as a parameter.

- When this value is present, this **seed value** is used as part of the random number generator such that its output is a set of pseudo-random values, which are consistent between job executions.
- When the browser is refreshed, the random numbers remain consistent when the seed value is present.
- This value must be a valid literal Integer value. Column references or functions returning Integer values are not supported.

## Basic Usage

### Example:

```
derive type:single value: RAND() as:'random'
```

**Output:** For each row, generate a random number between 0 and 1 in the new `random` function.

### Example with seed value:

```
derive type:single value: RAND(2) as:'random'
```

**Output:** For each row, generate a random number between 0 and 1 in the new `random` function. The generated random set of random values are consistent between job executions and are, in part, governed by the seed value 2.

## Syntax and Arguments

There are no arguments for this function.

```
derive type:single value: RAND([int_value])
```

Argument	Required?	Data Type	Description
int_value	N	integer	Integer literal

For more information on syntax standards, see *Language Documentation Syntax Notes*.

## int\_value

Optional Integer literal that is used to generate random numbers. Use of a seed value ensures consistency of output between job executions.

- Literal numeric values should not be quoted. Quoted values are treated as strings.
- Multiple columns and wildcards are not supported.

### Usage Notes:

Required?	Data Type	Example Value
No	Integer literal	14

### Examples

**Tip:** For additional examples, see *Common Tasks*.

### Example - Random values

In the following example, the `random` column is generated by the `RAND` function:

```
derive type:single value:RAND() as:'random'
```

source	random
A	0.516845703365675
B	0.71118736300207
C	0.758686066027118
D	0.640146255791255

### Example - Type check functions

The `RAND` function is typically used to introduce randomness of some kind in your data. In the following example, it is used to perform sampling within your wider dataset.

**Tip:** Keep in mind that for larger datasets the application displays only a sample of them. This method of randomization is applied to the full dataset during job execution.

### Source:

You want to extract a random sample of 20% of your set of orders for further study:

OrderId	Qty	ProdId
1001	30	Widgets
1002	10	Big Widgets
1003	5	Big Widgets

1004	40	Widgets
1005	80	Tiny Widgets
1006	20	Widgets
1007	100	Tiny Widgets

**Transform:**

You can use the following transform to generate a random integer from one to 10:

```
derive type:single value:round(RAND() * 10) as:'random'
```

You can now use the following transform to keep only the rows that contain random values that are in the top 20%, where the value is 9 or 10:

```
keep row:(random > 8)
```

**Results:**

**NOTE:** Since the results are randomized, your results might vary.

OrderId	Qty	ProdlId	random
1005	80	Tiny Widgets	9
1007	100	Tiny Widgets	10