2025-02-09

Eqalc

Convert math equations to functions.

v0.1.0 Tijme MIT

About

I created this package, because I thought it was very annoying to have to write down the equation in both math notation and code. This package allows you to write down the equation in math notation and convert it to a function.

Usage

- math-to-code()
- math-to-data()
- math-to-func()
- math-to-table()

math-to-code

Converts a math expression to code.

Example: `#math-to-code($x^2$)` will output `calc.pow(x,2)`.

Parameters

```
math-to-code(math: content) -> content
```

math `content`

The math expression.

math-to-data

Math to any data you might need.

Example: `#math-to-data($f(x)=x^2$)` will output:

```
#(
  func: (x => calc.pow(x,2)),
  str: "calc.pow(x,2)",
  x: "x",
  x-math: $x$,
  fx: "f(x)",
  fx-math: $f(x)$,
)
```

Parameters

```
math-to-data(math: content) ->
(func: function, str: string, x: string, x-math: content, fx: string, fx-math: content)
```

math `content`

The math expression.

math-to-func

Creates a function from a math expression.

Example: `#math-to-func($x^2$)` will output `#(x => calc.pow(x,2))`.

Parameters

```
math-to-func(math: content) -> function
```

math `content`

The math expression.

math-to-table

Creates a table of function values.

Example: `#math-to-table($x^2$, min: 1, max: 5, step: 1)` will output:

```
x    | 1 | 2 | 3 | 4 | 5 |
--------------------------
f(x) | 1 | 4 | 9 | 16| 25|
```

But in an actual table.

Parameters

```
math-to-table(
  math: content,
  min: integer,
  max: integer,
  step: integer,
  round: integer,
  name: content
) -> content
```

math `content`

The function to evaluate.

min `integer`

The minimum value of the domain.

Default: `0`

max `integer`

The maximum value of the domain.

Default: `5`

step `integer`

The step size.

Default: `1`

round `integer`

The integer of decimal places to round to.

Default: `2`

name `content`

The name of the function.

Default: `none`

Utility functions

- get-variable()
- math-to-str()

get-variable

Gets the main variable from a math expression.

Parameters

```
get-variable(math-str: string) -> string
```

math-str `string`

The math expression.

math-to-str

Converts math equations to strings.

Parameters

```
math-to-str(
  eq: content,
  get-first-part: boolean,
  depth: integer
) -> string
```

eq `content`

The math expression.

get-first-part `boolean`

Get the part before the equals sign. This is used to get the function name.

Default: `false`

depth `integer`

The depth of the recursion. This is used for debugging.

Default: `0`