Amtiri

AMTIRI SCRIPT

Math

Generated on November 25, 2026

Math

Ceiling

`ceil` · `ceil(value: double, [module: double])`

Rounds up, or up to the nearest multiple of module when it is given.

Parameters

Name	Type	Description
`value`	`double`	The number to round up.
`module`	`double` (optional)	The step to round up to. Defaults to 1.

Returns `double`: The rounded-up value.

Examples

```
ceil(42.1) # 43.0
ceil(LEVEL, 5) # 45.0
```

Clamp

`clamp` · `clamp(value: double, min: double, max: double)`

Holds a value inside a range: anything below min reads as min, anything above max reads as max.

Parameters

Name	Type	Description
`value`	`double`	The number to confine.
`min`	`double`	The lowest value that may come out.
`max`	`double`	The highest value that may come out.

Returns `double`: The value, confined to the range.

Examples

```
clamp(LEVEL, 0, 40) # 40
clamp(-1, 0, 10) # 0
clamp(LEVEL, SETPOINTS.low, SETPOINTS.high) # 42.5
```

Cosine

`cos` · `cos(value: double)`

The cosine of an angle given in radians.

Parameters

Name	Type	Description
`value`	`double`	The angle, in radians.

Returns `double`: The cosine.

Examples

```
cos(0) # 1.0
cos(PI) # -1.0
```

Exponential

`exp` · `exp(value: double)`

e raised to the power of value - the inverse of ln.

Parameters

Name	Type	Description
`value`	`double`	The exponent.

Returns `double`: The exponential.

Examples

```
exp(0) # 1.0
exp(1) # 2.7182818284590455
```

Factorial

`fact` · `fact(value: int)`

The factorial of a whole number, value!.

Parameters

Name	Type	Description
`value`	`int`	The number whose factorial is taken.

Returns `double`: The factorial.

Examples

```
fact(5) # 120.0
```

Floor

`floor` · `floor(value: double, [module: double])`

Rounds down, or down to the nearest multiple of module when it is given.

Parameters

Name	Type	Description
`value`	`double`	The number to round down.
`module`	`double` (optional)	The step to round down to. Defaults to 1.

Returns `double`: The rounded-down value.

Examples

```
floor(LEVEL) # 42.0
floor(LEVEL, 5) # 40.0
floor(7.3, 2) # 6.0
```

Logarithm (base 10)

`log` · `log(value: double)`

The logarithm of a value in base 10.

Parameters

Name	Type	Description
`value`	`double`	The number whose logarithm is taken.

Returns `double`: The base-10 logarithm.

Examples

```
log(1000) # 3.0
```

Maximum

`max` · `max(value: double...)`

The largest of the values given.

Parameters

Name	Type	Description
`value`	`double` (1 or more)	A number to compare. Add as many as needed.

Returns `double`: The largest value.

Examples

```
max(LEVEL, 50, COUNT) # 50
max(1, 2.5) # 2.5
```

Minimum

`min` · `min(value: double...)`

The smallest of the values given.

Parameters

Name	Type	Description
`value`	`double` (1 or more)	A number to compare. Add as many as needed.

Returns `double`: The smallest value.

Examples

```
min(LEVEL, SETPOINTS.high) # 42.5
min(4, 2.5, 7) # 2.5
```

Modulus

`mod` · `mod(value: double, base: double)`

The remainder of dividing value by base - a wrapped counter, or an every-nth test.

Parameters

Name	Type	Description
`value`	`double`	The number to divide.
`base`	`double`	The divisor to wrap around.

Returns `double`: The remainder, with the sign of the dividend.

Examples

```
mod(17, 5) # 2.0
mod(COUNT, 2) # 1.0
```

Natural logarithm

`ln` · `ln(value: double)`

The logarithm of a value in base e.

Parameters

Name	Type	Description
`value`	`double`	The number whose logarithm is taken.

Returns `double`: The natural logarithm.

Examples

```
ln(E) # 1.0
ln(1) # 0.0
```

Random

`random` · `random()`

A fresh pseudo-random number between 0 inclusive and 1 exclusive on every evaluation. Never use it where a result has to be reproducible.

Returns `double`: A number in the range 0 to 1.

Examples

```
random() # *
random() < 1 # true
```

Round

`round` · `round(value: double, [module: double])`

Rounds to the nearest whole number, or to the nearest multiple of module when it is given - round(x, 0.5) snaps to halves. With no module the result is a whole number, not a decimal.

Parameters

Name	Type	Description
`value`	`double`	The number to round.
`module`	`double` (optional)	The step to round to. Defaults to 1, i.e. the nearest whole number.

Returns `double`: The rounded value: a whole number when no module is given, a decimal otherwise.

Examples

```
round(LEVEL) # 43
round(41.26, 0.5) # 41.5
round(LEVEL, 10) # 40.0
```

Scale

`scale` · `scale(value: double, originMin: double, originMax: double, destinationMin: double, destinationMax: double)`

Maps a value from one range onto another - the usual way to turn a raw sensor reading into engineering units. The result is not confined to the destination range; use scalein for that.

Parameters

Name	Type	Description
`value`	`double`	The reading to convert.
`originMin`	`double`	The lowest value the reading can take.
`originMax`	`double`	The highest value the reading can take.
`destinationMin`	`double`	The value originMin maps onto.
`destinationMax`	`double`	The value originMax maps onto.

Returns `double`: The value expressed in the destination range.

Examples

```
scale(12, 4, 20, 0, 100) # 50.0
scale(24, 4, 20, 0, 100) # 125.0
```

Scale (confined)

`scalein` · `scalein(value: double, originMin: double, originMax: double, destinationMin: double, destinationMax: double)`

Maps a value from one range onto another exactly as scale does, then confines the result to the destination range, whichever way round it runs - an out-of-range reading cannot drive an output past its limits.

Parameters

Name	Type	Description
`value`	`double`	The reading to convert.
`originMin`	`double`	The lowest value the reading can take.
`originMax`	`double`	The highest value the reading can take.
`destinationMin`	`double`	The value originMin maps onto, and one limit of the result.
`destinationMax`	`double`	The value originMax maps onto, and the other limit of the result.

Returns `double`: The value expressed in, and confined to, the destination range.

Examples

```
scalein(24, 4, 20, 0, 100) # 100.0
scalein(50, 0, 100, 100, 0) # 50.0
```

Sine

`sin` · `sin(value: double)`

The sine of an angle given in radians.

Parameters

Name	Type	Description
`value`	`double`	The angle, in radians.

Returns `double`: The sine.

Examples

```
sin(PI / 2) # 1.0
sin(0) # 0.0
```

Square root

`sqrt` · `sqrt(value: double)`

The square root of a value.

Parameters

Name	Type	Description
`value`	`double`	The number whose root is taken. A negative input yields a not-a-number result.

Returns `double`: The square root.

Examples

```
sqrt(16) # 4.0
sqrt(-1) # NaN
```

Tangent

`tan` · `tan(value: double)`

The tangent of an angle given in radians.

Parameters

Name	Type	Description
`value`	`double`	The angle, in radians.

Returns `double`: The tangent.

Examples

```
tan(0) # 0.0
tan(PI / 4) # 0.9999999999999999
```

This page describes Amtiri Script 5.5.0 as shipped with Data Orchester engine 6.22.0 and Site 1.27.0.