

Values and types

Generated on October 14, 2026

Values and types

Error default

`evl` · `evl(value: any, default: any)`

Substitutes a default when evaluating a value raises an error, instead of letting the whole script fail. This is the guard for a variable that has never been written, which `nvl` cannot catch.

Parameters

Name	Type	Description
<code>value</code>	<code>any</code>	The expression to evaluate.
<code>default</code>	<code>any</code>	What to use instead when it fails.

Returns `any`: The value, or the default when it errored.

Examples

```
evl(NEVER_WRITTEN, 0) # 0
evl(LEVEL, 0) # 42.5
evl(REGISTERS[9], -1) # -1
```

Infinite default

`ivl` · `ivl(value: double, default: any)`

Substitutes a default when a value came out infinite - the guard to pair with a division whose divisor can reach zero.

Parameters

Name	Type	Description
<code>value</code>	<code>double</code>	The number to test.
<code>default</code>	<code>any</code>	What to use instead when it is infinite.

Returns `any`: The value, or the default when it is infinite.

Examples

```
ivl(LEVEL / 0, -1) # -1
ivl(LEVEL / 2, -1) # 21.25
```

Length

`length` · `length(value: any)`

How many items a list or record holds, or how many characters a text has. Yields null for anything else.

Parameters

Name	Type	Description
<code>value</code>	<code>any</code>	The list, record or text to measure.

Returns `int`: The item or character count, or null.

Examples

```
length(TAG) # 5
length(REGISTERS) # 4
length(SETPOINTS) # 2
length(LEVEL) # null
```

Not-a-number default

`inn` · `inn(value: double, default: any)`

Substitutes a default when a value is not a number: NaN, which is what a square root of a negative reading produces, or anything that is not numeric at all, such as text or null.

Parameters

Name	Type	Description
<code>value</code>	<code>double</code>	The number to test.
<code>default</code>	<code>any</code>	What to use instead when it is not a number.

Returns `any`: The value, or the default when it is not a number.

Examples

```
inn(sqrt(-1), 0) # 0
inn("abc", 0) # 0
inn(LEVEL, 0) # 42.5
```

Null default

`nvl` · `nvl(value: any, default: any)`

Substitutes a default when a value is null. It cannot guard a variable that has never been written: reading one is an error before `nvl` sees it, so use `evl` for that.

Parameters

Name	Type	Description
value	any	The value to test.
default	any	What to use instead when value is null.

Returns any : The value, or the default when it is null.

Examples

```
nv1(null, "none") # "none"
nv1(LEVEL, 0) # 42.5
nv1(NEVER_WRITTEN, 0) # error
```

To decimal

to_double · to_double(value: any)

Converts a value to a decimal number.

Parameters

Name	Type	Description
value	any	The value to convert.

Returns double : The decimal number.

Examples

```
to_double("12.5") # 12.5
to_double(COUNT) # 3.0
```

To text

to_string · to_string(value: any)

Converts a value to its text form.

Parameters

Name	Type	Description
value	any	The value to convert.

Returns string : The text form.

Examples

```

to_string(COUNT) # "3"
to_string(LEVEL) # "42.5"
to_string(PUMP_ON) # "true"

```

To whole number

to_int · to_int(value: any)

Converts a value to a whole number, truncating any decimals.

Parameters

Name	Type	Description
value	any	The value to convert.

Returns int : The whole number.

Examples

```

to_int(LEVEL) # 42
to_int("7") # 7
to_int(-2.7) # -2

```

Type of

type_of · type_of(value: any)

The name of a value's underlying type, for diagnosing a script that is receiving something other than what it expects.

Parameters

Name	Type	Description
value	any	The value to inspect.

Returns string : The type name, or null when the value is null.

Examples

```

type_of(LEVEL) # "Double"
type_of(COUNT) # "Integer"
type_of(START) # "LocalDateTime"
type_of(null) # null

```

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