

Package: `tengen` (via `r-universe`)

June 7, 2026

Title Tensor Generics

Version 0.2.0

Description Generic functions and common objects for working with tensors.

License MIT + file LICENSE

URL <https://r-xla.github.io/tengen/>, <https://github.com/r-xla/tengen>

BugReports <https://github.com/r-xla/tengen/issues>

Imports `checkmate`, `cli`, `rlang`

Suggests `testthat` (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen `list(markdown = TRUE)`

RoxygenNote 7.3.3

Repository <https://r-xla.r-universe.dev>

Date/Publication 2026-04-08 06:53:16 UTC

RemoteUrl <https://github.com/r-xla/tengen>

RemoteRef v0.2.0

RemoteSha 55d1f433b5704a024ff16e3d0118cf718dba04b5

Contents

<code>as_array</code>	2
<code>as_dtype</code>	2
<code>as_raw</code>	3
<code>assert_dtype</code>	3
<code>BooleanType</code>	4
<code>DataType</code>	4
<code>device</code>	5
<code>dtype</code>	5
<code>FloatType</code>	6

IntegerType	6
is_dtype	7
ndims	7
nelts	8
shape	8
UIntegerType	9

Index 10

as_array	<i>As Array</i>
----------	-----------------

Description

Convert a value (usually a tensor from some framework) to an R array.

Usage

```
as_array(x, ...)
```

Arguments

x	(any) Object.
...	(any) Additional arguments.

Value

(any)

as_dtype	<i>Convert to DataType</i>
----------	----------------------------

Description

Convert to DataType.

Usage

```
as_dtype(x)
```

Arguments

x	(any) Object to convert. Can currently be a string (e.g. "i1", "i32", "f32", ...) or a DataType object.
---	--

Value

DataType

as_raw	<i>As Ray</i>
--------	---------------

Description

Convert a value (usually a tensor from some framework) to a `raw()` vector.

Usage

```
as_raw(x, ...)
```

Arguments

x	(any) Object.
...	(any) Additional arguments.

Value

(any)

assert_dtype	<i>Assert DataType</i>
--------------	------------------------

Description

Assert that an object is a [DataType](#).

Usage

```
assert_dtype(x, arg = rlang::caller_arg(x))
```

Arguments

x	(any) Object to check.
arg	(character(1)) Argument name for error messages.

Value

NULL (invisibly). Called for side effects.

BooleanType

BooleanType

Description

Represents the boolean type.

Usage

`BooleanType()`

Value

BooleanType

DataType

DataType Base Class

Description

DataType is the parent S3 class for all tensor data types. All data type classes inherit from DataType, enabling cross-type comparisons with == and != operators.

The specific data type classes are:

- [BooleanType\(\)](#) - Boolean (i1)
- [IntegerType\(\)](#) - Signed integers (i8, i16, i32, i64)
- [UIntegerType\(\)](#) - Unsigned integers (ui8, ui16, ui32, ui64)
- [FloatType\(\)](#) - Floating point (f32, f64)

Details

This is a virtual base class - you cannot create instances directly. Use the specific type constructors instead.

See Also

[BooleanType\(\)](#), [IntegerType\(\)](#), [UIntegerType\(\)](#), [FloatType\(\)](#)

device	<i>Device</i>
--------	---------------

Description

Get the device of a tensor.

Usage

```
device(x, ...)
```

Arguments

x	(any) Object.
...	(any) Additional arguments.

Value

(any)

dtype	<i>Data Type</i>
-------	------------------

Description

Get the data type of a tensor. The return type depends on the specific tensor implementation.

Usage

```
dtype(x, ...)
```

Arguments

x	(any) Object.
...	(any) Additional arguments.

Value

(any)

FloatType

FloatType

Description

Represents a floating point type with a given bit width.

Usage

FloatType(value)

Arguments

value (integer(1))

Value

FloatType

IntegerType

IntegerType (signed)

Description

Represents a signed integer type with a given bit width.

Usage

IntegerType(value)

Arguments

value (integer(1))

Value

IntegerType

`is_dtype`*Is DataType*

Description

Check if an object is a [DataType](#).

Usage

```
is_dtype(x)
```

Arguments

`x` (any)
Object to check.

Value

```
logical(1)
```

`ndims`*Number of Dimensions*

Description

Get the number of dimensions of a tensor.

Usage

```
ndims(x)
```

Arguments

`x` (any)
Object.

Value

```
(integer)
```

nelts	<i>Number of Elements</i>
-------	---------------------------

Description

Get the number of elements of a tensor.

Usage

```
nelts(x)
```

Arguments

x	(any) Object.
---	------------------

Value

(integer)

shape	<i>Shape</i>
-------	--------------

Description

Get the shape of a tensor.

Usage

```
shape(x, ...)
```

Arguments

x	(any) Object.
...	(any) Additional arguments.

Value

(integer())
The shape of the tensor.

UIntegerType	<i>UIntegerType</i>
--------------	---------------------

Description

Represents an unsigned integer type with a given bit width.

Usage

UIntegerType(value)

Arguments

value (integer(1))

Value

UIntegerType

Index

`as_array`, 2
`as_dtype`, 2
`as_raw`, 3
`assert_dtype`, 3

`BooleanType`, 4
`BooleanType()`, 4

`DataType`, 2, 3, 4, 7
`device`, 5
`dtype`, 5

`FloatType`, 6
`FloatType()`, 4

`IntegerType`, 6
`IntegerType()`, 4
`is_dtype`, 7

`ndims`, 7
`nelts`, 8

`shape`, 8

`UIntegerType`, 9
`UIntegerType()`, 4