tensor.and

```
Copy fn and(self: @Tensor, other: @Tensor) -> Tensor;
```

Computes the logical AND of two tensors element-wise. The input tensors must have either:

* Exactly the same shape
* The same number of dimensions and the length of each dimension is either a common length or 1.
*

Args

* self
* (@Tensor
*) - The first tensor to be compared
* other
* (@Tensor
*) - The second tensor to be compared
*

Panics

* Panics if the shapes are not equal or broadcastable
*

Returns

A new Tensor with the same shape as the broadcasted inputs.

Examples

```
Copy use core::array::{ArrayTrait,SpanTrait};

use orion::operators::tensor::{TensorTrait,Tensor,BoolTensor};

fn and_example() -> Tensor { let tensor_1 = TensorTrait::::new( shape: array![3,3].span(), data: array![false,true,false,false,false,true,true,false,true,false,false,true].span(), );

let tensor_2 = TensorTrait::::new( shape: array![3,3].span(), data: array![false,false,true,true,false,true,false,true,false,true,false,true].span(), );

return tensor_1.and(@tensor_2); }

                [false,false,false,false,false,true,false,false,false,false,false,true]
```

Last updated 1 month ago