

# tensor.sin

## tensor.sin

...

```
Copy fnsin(self:@Tensor)->Tensor;
```

...

Computes the sine of all elements of the input tensor.

Args

- self
- (@Tensor
- ) - The input tensor.
- 

Returns

A newTensor of the same shape as the input tensor with the sine value of all elements in the input tensor.

Type Constraints

Constrain input and output types to fixed point tensors.

Example

...

```
Copy usecore::array::{ArrayTrait,SpanTrait};
```

```
useorion::operators::tensor::{TensorTrait,Tensor,FP8x23Tensor}; useorion::numbers::{FP8x23,FixedTrait};
```

```
fnsin_example()->Tensor { lettensor=TensorTrait::new( shape:array![3].span(), data:array![  
FixedTrait::new_unscaled(0,false), FixedTrait::new_unscaled(1,false), FixedTrait::new_unscaled(2,false) ] .span(), );
```

```
returntensor.sin(); }
```

```
[0,7058770,7627740] // The fixed point representation of // [0,0.8414...,0.9092...]
```

...

[Previous tensor.cumsum](#) [Next tensor.cos](#)

Last updated3 months ago