Scalar Output

`torch.numel`

`torch.numel(input)`	
	Require
	• $\lvert\texttt{input}\rvert = (d_1, d_2, \ldots, d_k)$
	Guarantees
	• $d_1 \cdot d_2 \cdot \cdots \cdot d_k$를 스칼라로 반환

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \texttt{numel}(E) \Rightarrow e[1]e[2]\cdots e[k], c}$$

`torch.Tensor.dim`, `torch.Tensor.ndim`, `torch.Tensor.ndimension`

`a.dim(), a.ndim or a.ndimension()`	
	Require
	• $\lvert\texttt{a}\rvert = (d_1, d_2, \ldots, d_k)$
	Guarantees
	• k를 스칼라로 반환

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash E.\texttt{dim}() \Rightarrow \texttt{rank}(e), c}$$

$$\frac{\sigma \vdash E.\texttt{dim}() \Rightarrow k, c}{\sigma \vdash E.\texttt{ndim} \Rightarrow k, c}\ \text{alias}$$

$$\frac{\sigma \vdash E.\texttt{dim}() \Rightarrow k, c}{\sigma \vdash E.\texttt{ndimension}() \Rightarrow k, c}\ \text{alias}$$

`torch.eq`

`torch.eq(input, other)`	
	Require
	Guarantees
	• $input$과 $other$의 내용물과 shape이 모두 일치하면 $True$, 아니면 $False$

$$\frac{\sigma \vdash A \Rightarrow \_, c_a \qquad \sigma \vdash B \Rightarrow \_, c_b}{\sigma \vdash torch.\texttt{eq}(A, B) \Rightarrow bool, c_a \cup c_b}$$

Conversion To/From NonTensor

`torch.Tensor.tolist`

- 파이썬 리스트로 텐서를 바꿔주는 함수

`torch.as_tensor(numpy_list)`, `torch.from_numpy`

- 넘파이 배열이나 파이썬 리스트로부터 텐서를 만드는 함수