Module: tf.compat.v1.debugging

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/debugging#top_of_page)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/debugging#functions)

Public API for tf.debugging namespace.

Functions

[Assert(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/Assert): Asserts that the given condition is true.

[assert\_all\_finite(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/verify_tensor_all_finite): Assert that the tensor does not contain any NaN's or Inf's.

[assert\_equal(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_equal): Assert the condition x == y holds element-wise.

[assert\_greater(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_greater): Assert the condition x > y holds element-wise.

[assert\_greater\_equal(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_greater_equal): Assert the condition x >= y holds element-wise.

[assert\_integer(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_integer): Assert that x is of integer dtype.

[assert\_less(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_less): Assert the condition x < y holds element-wise.

[assert\_less\_equal(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_less_equal): Assert the condition x <= y holds element-wise.

[assert\_near(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_near): Assert the condition x and y are close element-wise.

[assert\_negative(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_negative): Assert the condition x < 0 holds element-wise.

[assert\_non\_negative(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_non_negative): Assert the condition x >= 0 holds element-wise.

[assert\_non\_positive(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_non_positive): Assert the condition x <= 0 holds element-wise.

[assert\_none\_equal(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_none_equal): Assert the condition x != y holds for all elements.

[assert\_positive(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_positive): Assert the condition x > 0 holds element-wise.

[assert\_proper\_iterable(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/assert_proper_iterable): Static assert that values is a "proper" iterable.

[assert\_rank(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_rank): Assert x has rank equal to rank.

[assert\_rank\_at\_least(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_rank_at_least): Assert x has rank equal to rank or higher.

[assert\_rank\_in(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_rank_in): Assert x has rank in ranks.

[assert\_same\_float\_dtype(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/assert_same_float_dtype): Validate and return float type based on tensors and dtype.

[assert\_scalar(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_scalar): Asserts that the given tensor is a scalar (i.e. zero-dimensional).

[assert\_shapes(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/debugging/assert_shapes): Assert tensor shapes and dimension size relationships between tensors.

[assert\_type(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/assert_type): Statically asserts that the given Tensor is of the specified type.

[check\_numerics(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/check_numerics): Checks a tensor for NaN and Inf values.

[get\_log\_device\_placement(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/get_log_device_placement): Get if device placements are logged.

[is\_finite(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/math/is_finite): Returns which elements of x are finite.

[is\_inf(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/math/is_inf): Returns which elements of x are Inf.

[is\_nan(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/math/is_nan): Returns which elements of x are NaN.

[is\_non\_decreasing(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/math/is_non_decreasing): Returns True if x is non-decreasing.

[is\_numeric\_tensor(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/is_numeric_tensor): Returns True if the elements of tensor are numbers.

[is\_strictly\_increasing(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/math/is_strictly_increasing): Returns True if x is strictly increasing.

[set\_log\_device\_placement(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/debugging/set_log_device_placement): Set if device placements should be logged.

# tf.compat.v1.debugging.assert\_shapes

Assert tensor shapes and dimension size relationships between tensors.

tf.compat.v1.debugging.assert\_shapes(  
    shapes,  
    data=None,  
    summarize=None,  
    message=None,  
    name=None  
)

Defined in [python/ops/check\_ops.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/ops/check_ops.py).

This Op checks that a collection of tensors shape relationships satisfies given constraints.

#### Example:

tf.assert\_shapes({  
  x: ('N', 'Q'),  
  y: ('N', 'D'),  
  param: ('Q',),  
  scalar: ()  
})

Example of adding a dependency to an operation:

with tf.control\_dependencies([tf.assert\_shapes(shapes)]):  
  output = tf.matmul(x, y, transpose\_a=True)

If x, y, param or scalar does not have a shape that satisfies all specified constraints, message, as well as the first summarize entries of the first encountered violating tensor are printed, andInvalidArgumentError is raised.

Size entries in the specified shapes are checked against other entries by their **hash**, except: - a size entry is interpreted as an explicit size if it can be parsed as an integer primitive. - a size entry is interpreted as any size if it is None or '.'.

If the first entry of a shape is ... (type Ellipsis) or '\*' that indicates a variable number of outer dimensions of unspecified size, i.e. the constraint applies to the inner-most dimensions only.

Scalar tensors and specified shapes of length zero (excluding the 'inner-most' prefix) are both treated as having a single dimension of size one.

#### Args:

* **shapes**: dictionary with (Tensor to shape) items. A shape must be an iterable.
* **data**: The tensors to print out if the condition is False. Defaults to error message and first few entries of the violating tensor.
* **summarize**: Print this many entries of the tensor.
* **message**: A string to prefix to the default message.
* **name**: A name for this operation (optional). Defaults to "assert\_shapes".

#### Returns:

Op raising InvalidArgumentError unless all shape constraints are satisfied. If static checks determine all constraints are satisfied, a no\_op is returned.

#### Raises:

* **ValueError**: If static checks determine any shape constraint is violated.