

tf.set_random_seed

```
set_random_seed(seed)
```

Defined in [tensorflow/python/framework/random_seed.py](#).

See the guide: [Constants, Sequences, and Random Values > Random Tensors](#)

Sets the graph-level random seed.

Operations that rely on a random seed actually derive it from two seeds: the graph-level and operation-level seeds. This sets the graph-level seed.

Its interactions with operation-level seeds is as follows:

1. If neither the graph-level nor the operation seed is set: A random seed is used for this op.
2. If the graph-level seed is set, but the operation seed is not: The system deterministically picks an operation seed in conjunction with the graph-level seed so that it gets a unique random sequence.
3. If the graph-level seed is not set, but the operation seed is set: A default graph-level seed and the specified operation seed are used to determine the random sequence.
4. If both the graph-level and the operation seed are set: Both seeds are used in conjunction to determine the random sequence.

To illustrate the user-visible effects, consider these examples:

To generate different sequences across sessions, set neither graph-level nor op-level seeds:

```
a = tf.random_uniform([1])
b = tf.random_normal([1])

print("Session 1")
with tf.Session() as sess1:
    print(sess1.run(a)) # generates 'A1'
    print(sess1.run(a)) # generates 'A2'
    print(sess1.run(b)) # generates 'B1'
    print(sess1.run(b)) # generates 'B2'

print("Session 2")
with tf.Session() as sess2:
    print(sess2.run(a)) # generates 'A3'
    print(sess2.run(a)) # generates 'A4'
    print(sess2.run(b)) # generates 'B3'
    print(sess2.run(b)) # generates 'B4'
```

To generate the same repeatable sequence for an op across sessions, set the seed for the op:

```

a = tf.random_uniform([1], seed=1)
b = tf.random_normal([1])

# Repeatedly running this block with the same graph will generate the same
# sequence of values for 'a', but different sequences of values for 'b'.
print("Session 1")
with tf.Session() as sess1:
    print(sess1.run(a)) # generates 'A1'
    print(sess1.run(a)) # generates 'A2'
    print(sess1.run(b)) # generates 'B1'
    print(sess1.run(b)) # generates 'B2'

print("Session 2")
with tf.Session() as sess2:
    print(sess2.run(a)) # generates 'A1'
    print(sess2.run(a)) # generates 'A2'
    print(sess2.run(b)) # generates 'B3'
    print(sess2.run(b)) # generates 'B4'

```

To make the random sequences generated by all ops be repeatable across sessions, set a graph-level seed:

```

tf.set_random_seed(1234)
a = tf.random_uniform([1])
b = tf.random_normal([1])

# Repeatedly running this block with the same graph will generate the same
# sequences of 'a' and 'b'.
print("Session 1")
with tf.Session() as sess1:
    print(sess1.run(a)) # generates 'A1'
    print(sess1.run(a)) # generates 'A2'
    print(sess1.run(b)) # generates 'B1'
    print(sess1.run(b)) # generates 'B2'

print("Session 2")
with tf.Session() as sess2:
    print(sess2.run(a)) # generates 'A1'
    print(sess2.run(a)) # generates 'A2'
    print(sess2.run(b)) # generates 'B1'
    print(sess2.run(b)) # generates 'B2'

```

Args:

- `seed`: integer.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

[Support](#)

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

[Terms](#) | [Privacy](#)