Introduction to TensorFlow

Constants, variables, and placeholders

郭耀仁

大綱

- 常數
- 變數
- Placeholders

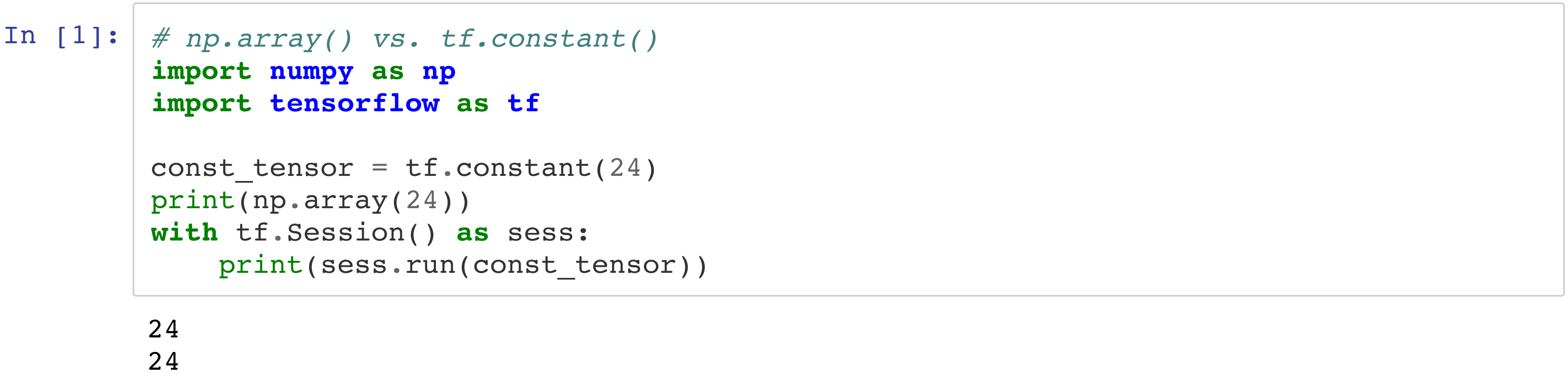
常數 Constants

如何宣告張量為常數

- `tf.constant()`
- `tf.zeros()`
- `tf.ones()`
- `tf.fill()`
- `tf.range()`
- `tf.random_normal()`
- `tf.random_uniform()`

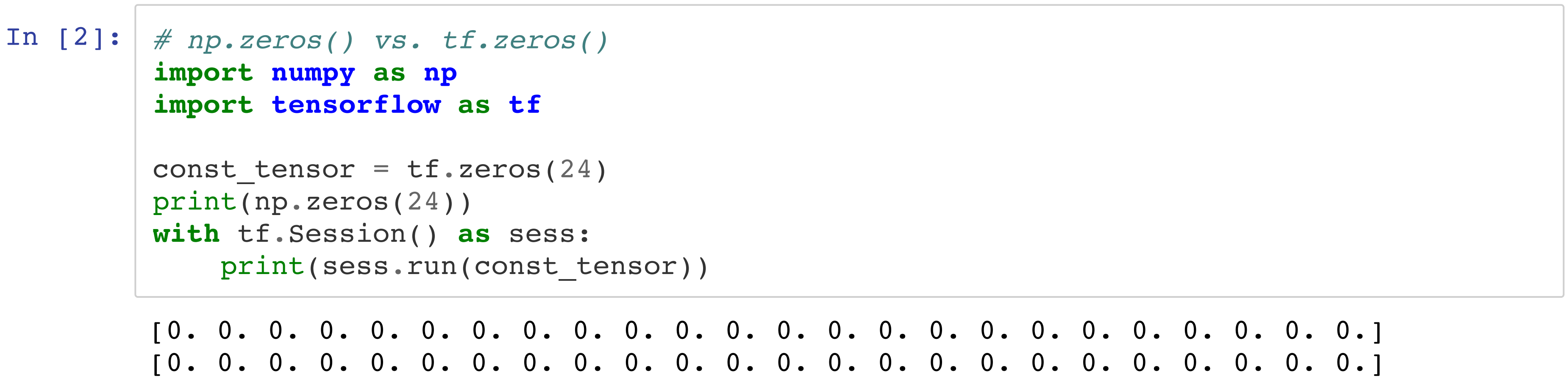
如何跟 NumPy 對應

NumPy	TensorFlow
`np.array()`	`tf.constant()`
`np.zeros()`	`tf.zeros()`
`np.ones()`	`tf.ones()`
`np.full()`	`tf.fill()`
`np.arange()`	`tf.range()`
`np.random.normal()`	`tf.random_normal()`
`np.random.uniform()`	`tf.random_uniform()`



```
In [1]:
# np.array() vs. tf.constant()
import numpy as np
import tensorflow as tf

const_tensor = tf.constant(24)
print(np.array(24))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

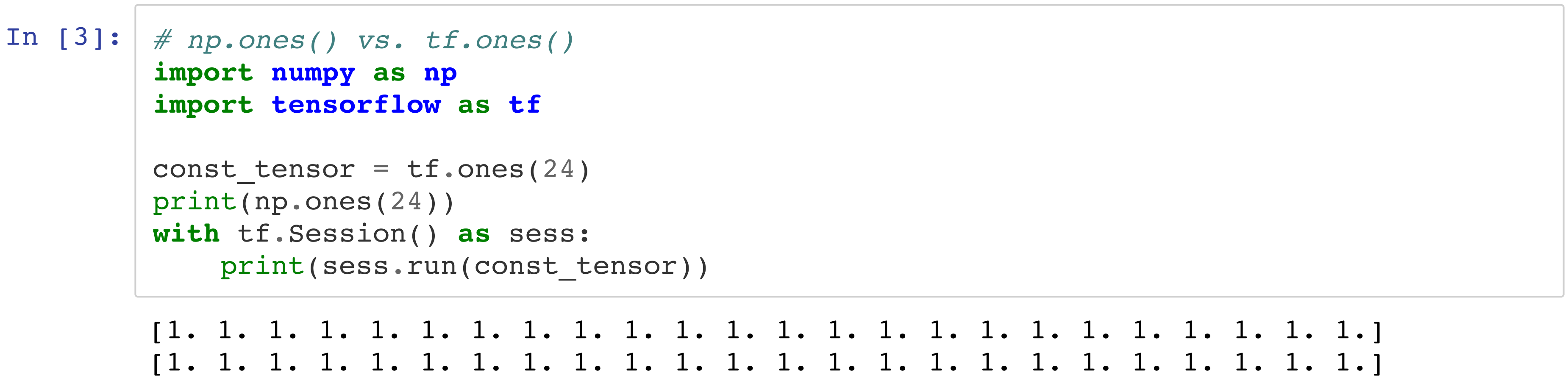
```
24
24
```



In [2]:

```
# np.zeros() vs. tf.zeros()
import numpy as np
import tensorflow as tf

const_tensor = tf.zeros(24)
print(np.zeros(24))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```



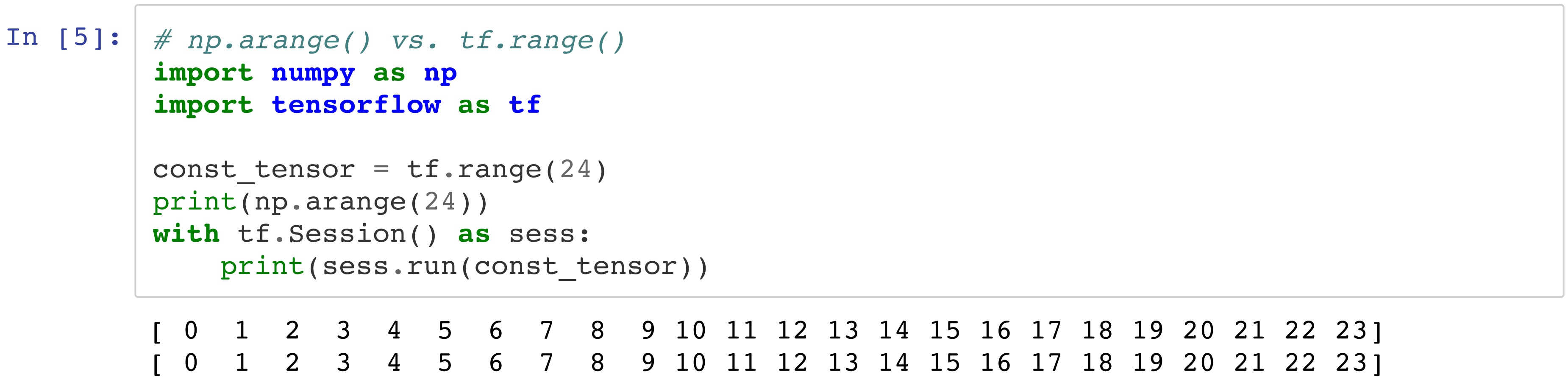
```python
# np.ones() vs. tf.ones()
import numpy as np
import tensorflow as tf

const_tensor = tf.ones(24)
print(np.ones(24))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [4]:

```
# np.full() vs. tf.fill()
import numpy as np
import tensorflow as tf

const_tensor = tf.fill((24,), 24)
print(np.full(24, (24,)))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

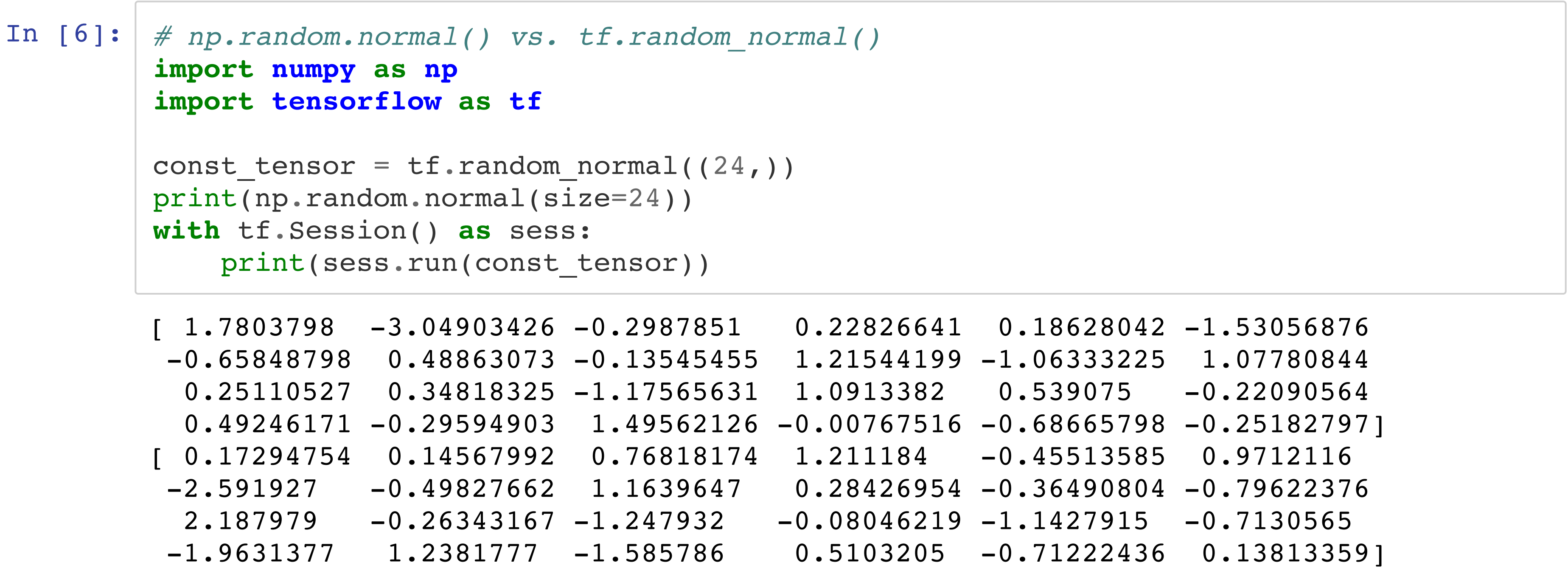
```
[24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24]
[24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24]
```



In [5]:

```
# np.arange() vs. tf.range()
import numpy as np
import tensorflow as tf

const_tensor = tf.range(24)
print(np.arange(24))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

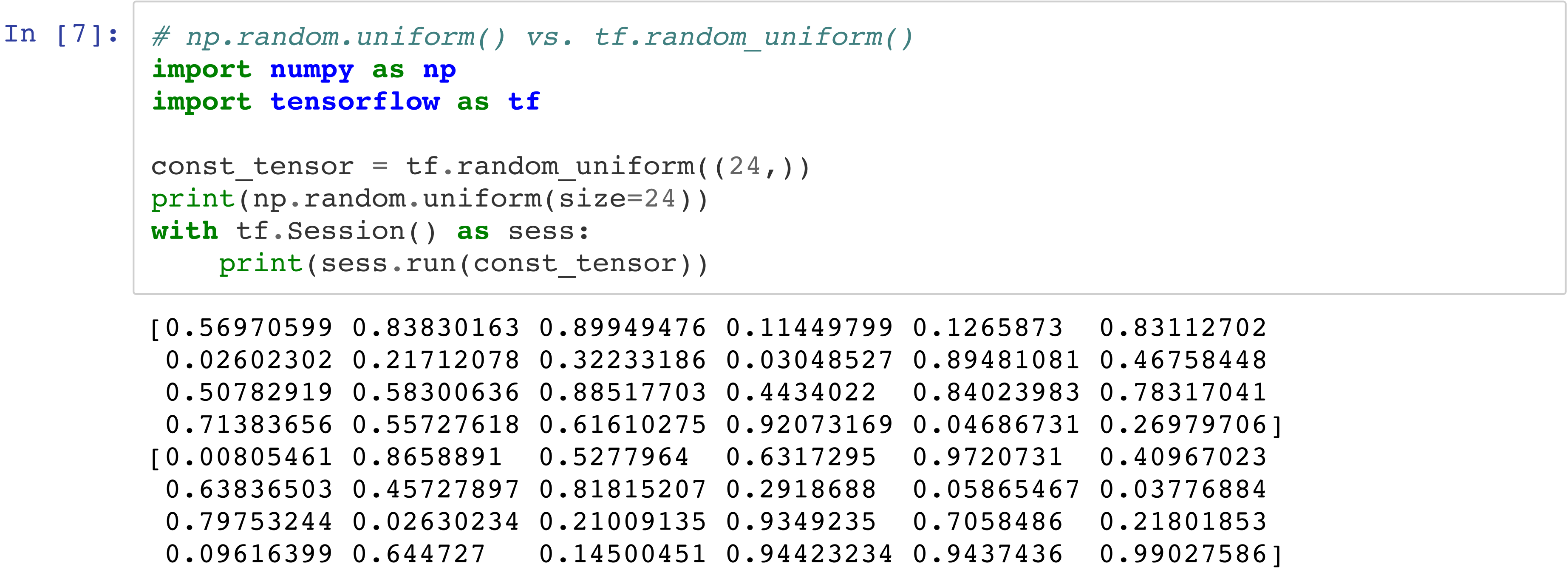
```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23]
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23]
```



In [6]:

```
# np.random.normal() vs. tf.random_normal()
import numpy as np
import tensorflow as tf

const_tensor = tf.random_normal((24,))
print(np.random.normal(size=24))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

```
[ 1.7803798  -3.04903426 -0.2987851   0.22826641  0.18628042 -1.53056876
 -0.65848798  0.48863073 -0.13545455  1.21544199 -1.06333225  1.07780844
  0.25110527  0.34818325 -1.17565631  1.0913382   0.539075   -0.22090564
  0.49246171 -0.29594903  1.49562126 -0.00767516 -0.68665798 -0.25182797]
[ 0.17294754  0.14567992  0.76818174  1.211184   -0.45513585  0.9712116
 -2.591927   -0.49827662  1.1639647   0.28426954 -0.36490804 -0.79622376
  2.187979   -0.26343167 -1.247932   -0.08046219 -1.1427915  -0.7130565
 -1.9631377   1.2381777  -1.585786    0.5103205  -0.71222436  0.13813359]
```



In [7]:

```
# np.random.uniform() vs. tf.random_uniform()
import numpy as np
import tensorflow as tf

const_tensor = tf.random_uniform((24,))
print(np.random.uniform(size=24))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

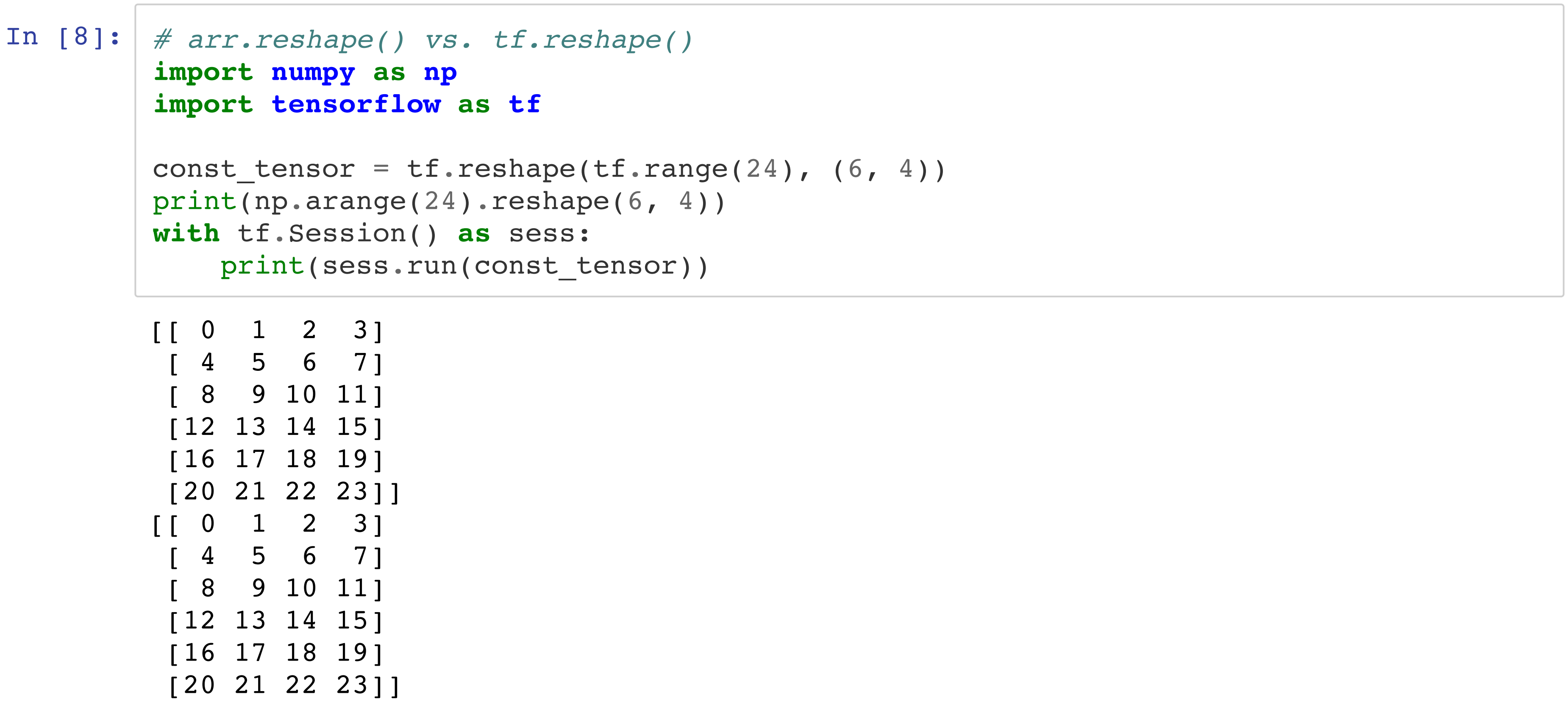
```
[0.56970599 0.83830163 0.89949476 0.11449799 0.1265873  0.83112702
 0.02602302 0.21712078 0.32233186 0.03048527 0.89481081 0.46758448
 0.50782919 0.58300636 0.88517703 0.4434022  0.84023983 0.78317041
 0.71383656 0.55727618 0.61610275 0.92073169 0.04686731 0.26979706]
[0.00805461 0.8658891  0.5277964  0.6317295  0.9720731  0.40967023
 0.63836503 0.45727897 0.81815207 0.2918688  0.05865467 0.03776884
 0.79753244 0.02630234 0.21009135 0.9349235  0.7058486  0.21801853
 0.09616399 0.644727   0.14500451 0.94423234 0.9437436  0.99027586]
```

如何使用 TensorFlow 處理矩陣

- `tf.reshape()`
- `tf.eye()`
- `tf.diag()`
- `tf.matrix_transpose()`
- `tf.matmul()`

如何跟 NumPy 對應

NumPy	TensorFlow
`arr.reshape()`	`tf.reshape()`
`np.eye()`	`tf.eye()`
`np.diag()`	`tf.diag()`
`np.transpose()`	`tf.matrix_transpose()`
`np.dot()`	`tf.matmul()`



In [8]:

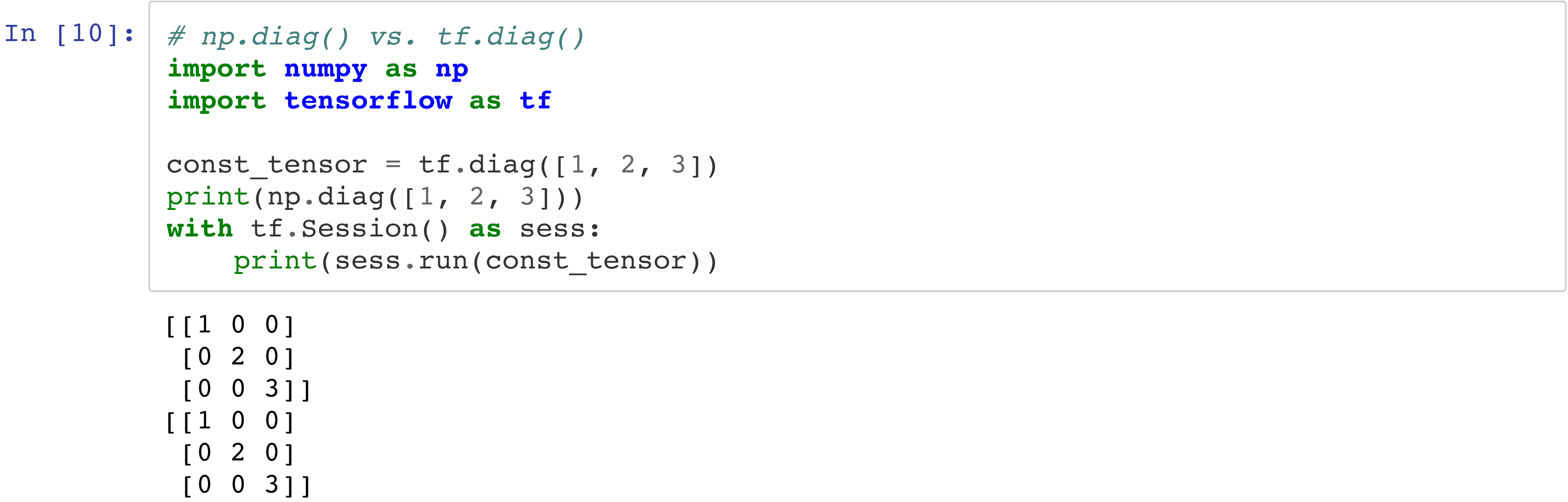
```
# arr.reshape() vs. tf.reshape()
import numpy as np
import tensorflow as tf

const_tensor = tf.reshape(tf.range(24), (6, 4))
print(np.arange(24).reshape(6, 4))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]]
```

In [9]:

```
# np.eye() vs. tf.eye()
import numpy as np
import tensorflow as tf

const_tensor = tf.eye(3)
print(np.eye(3))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```



In [10]:

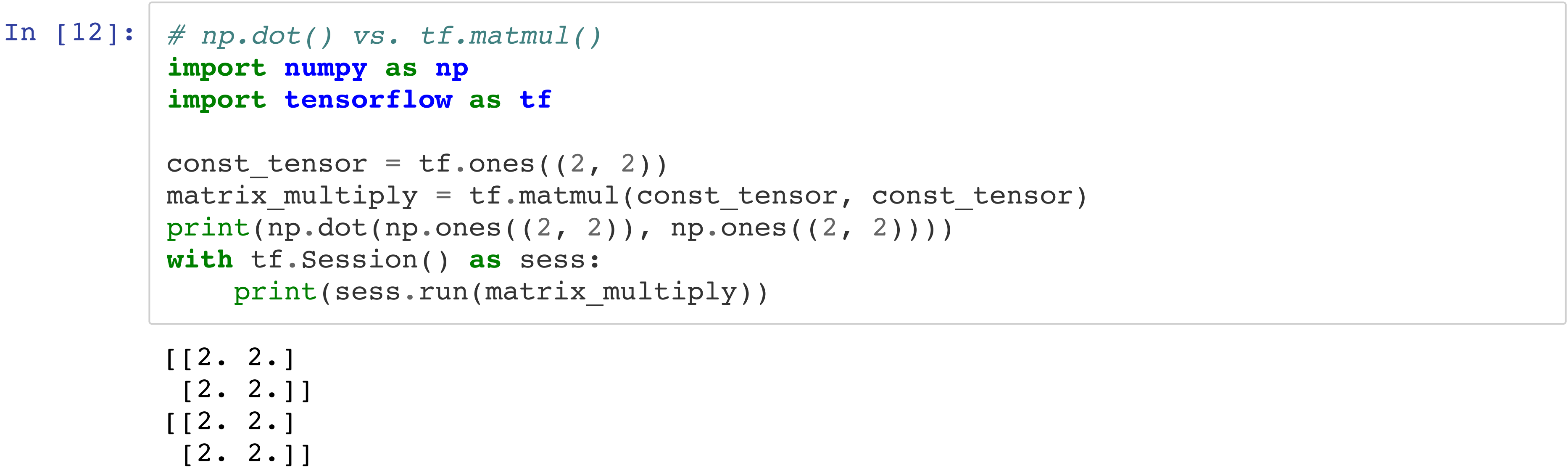
```python
# np.diag() vs. tf.diag()
import numpy as np
import tensorflow as tf

const_tensor = tf.diag([1, 2, 3])
print(np.diag([1, 2, 3]))
with tf.Session() as sess:
    print(sess.run(const_tensor))
```

```
[[1 0 0]
 [0 2 0]
 [0 0 3]]
[[1 0 0]
 [0 2 0]
 [0 0 3]]
```

In [11]:

```
# np.transpose() vs. tf.matrix_transpose()
import numpy as np
import tensorflow as tf

const_tensor = tf.ones((2, 4))
const_tensor_t = tf.matrix_transpose(const_tensor)
print(np.ones((2, 4)))
print(np.ones((2, 4)).T)
print("\n")
with tf.Session() as sess:
    print(sess.run(const_tensor))
    print(sess.run(const_tensor_t))
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]]
[[1. 1.]
 [1. 1.]
 [1. 1.]
 [1. 1.]]


[[1. 1. 1. 1.]
 [1. 1. 1. 1.]]
[[1. 1.]
 [1. 1.]
 [1. 1.]
 [1. 1.]]
```



```
# np.dot() vs. tf.matmul()
import numpy as np
import tensorflow as tf

const_tensor = tf.ones((2, 2))
matrix_multiply = tf.matmul(const_tensor, const_tensor)
print(np.dot(np.ones((2, 2)), np.ones((2, 2))))
with tf.Session() as sess:
    print(sess.run(matrix_multiply))
```

```
[[2. 2.]
 [2. 2.]]
[[2. 2.]
 [2. 2.]]
```

隨堂練習

使用 TensorFlow 常數張量計算 $u^T v$

$$u = \begin{bmatrix} 4 \\ -4 \\ -3 \end{bmatrix}$$

$$v = \begin{bmatrix} 4 \\ 2 \\ 4 \end{bmatrix}$$

隨堂練習

使用 TensorFlow 常數張量計算 $error$，其中 $error = y - y_{pred}$，而 $y_{pred} = X\theta$

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 5 & 8 \\ 6 & 10 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$y = \begin{bmatrix} 6 \\ 4 \\ 20 \\ 23 \end{bmatrix}$$

隨堂練習

使用 TensorFlow 常數張量計算 W 中的所有數值總和、各欄總和與各列總和

Hint: `tf.math.reduce_sum()`

$$W = \begin{bmatrix} 11 & 7 & 4 & 3 & 25 \\ 50 & 2 & 60 & 0 & 10 \end{bmatrix}$$

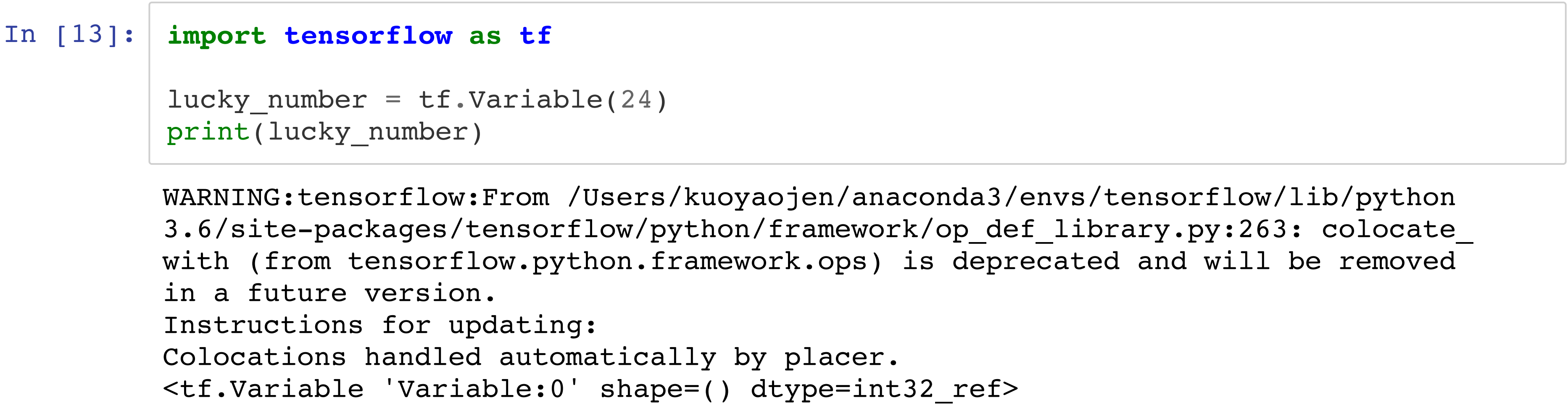
熟悉 NumPy 的 Python 使用者來說學習 TensorFlow 有很大優勢

https://www.numpy.org/devdocs/user/quickstart.html (https://www.numpy.org/devdocs/user/quickstart.html)

變數 Variables

以 tf.Variable() 宣告

變數張量的值可以更動，模型訓練過程要持續更新的參數會以變數張量宣告



In [13]:

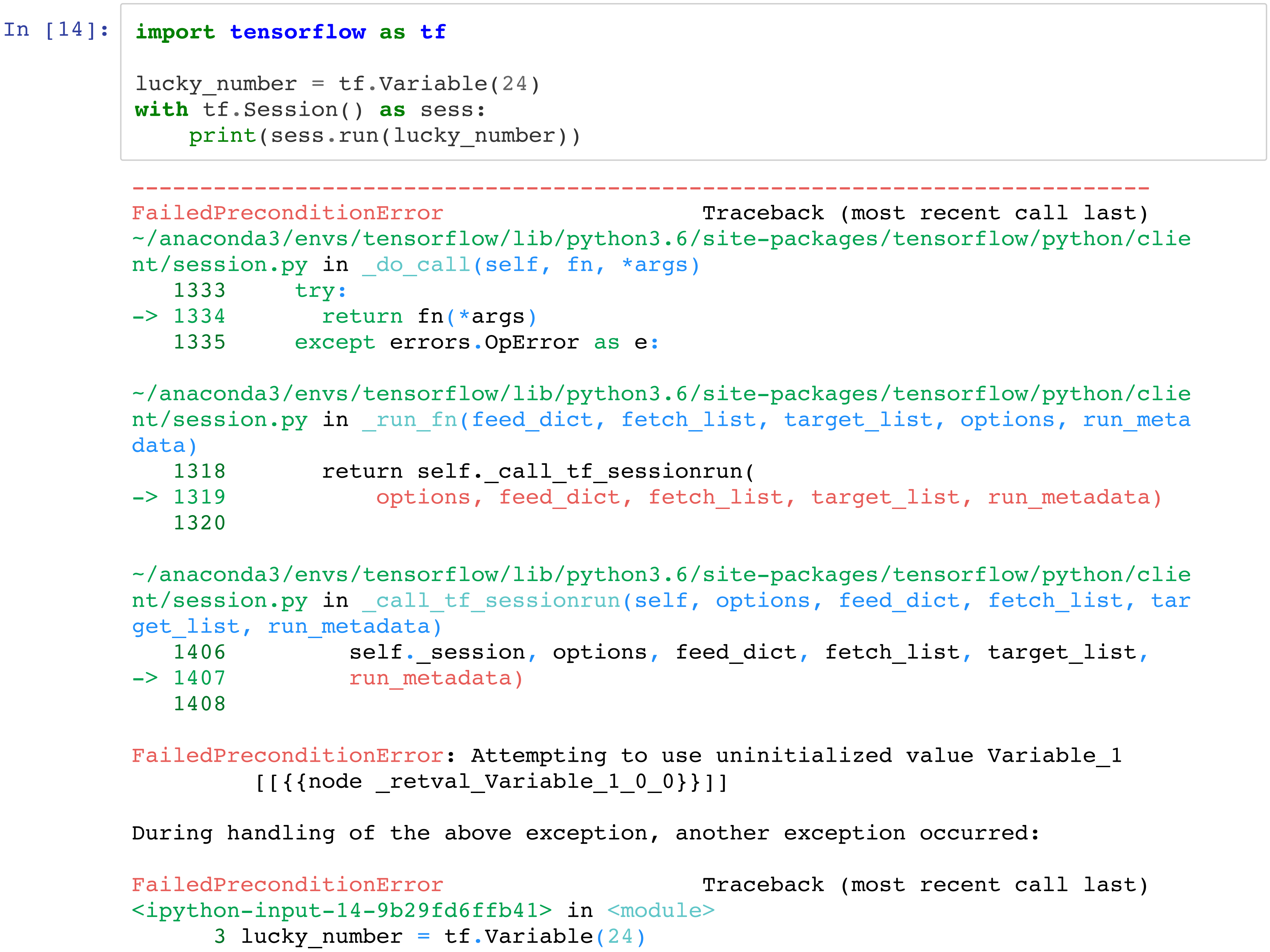
```
import tensorflow as tf

lucky_number = tf.Variable(24)
print(lucky_number)
```

```
WARNING:tensorflow:From /Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python
3.6/site-packages/tensorflow/python/framework/op_def_library.py:263: colocate_
with (from tensorflow.python.framework.ops) is deprecated and will be removed
in a future version.
Instructions for updating:
Colocations handled automatically by placer.
<tf.Variable 'Variable:0' shape=() dtype=int32_ref>
```

宣告變數張量不如常數張量那麼單純

- 宣告變數張量的初始值、類型與外觀
- 初始化變數張量

如果宣告的變數張量沒有經過初始化，將會得到錯誤



In [14]:

```
import tensorflow as tf

lucky_number = tf.Variable(24)
with tf.Session() as sess:
    print(sess.run(lucky_number))
```

```
---------------------------------------------------------------------------
FailedPreconditionError                   Traceback (most recent call last)
~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _do_call(self, fn, *args)
   1333     try:
-> 1334       return fn(*args)
   1335     except errors.OpError as e:

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _run_fn(feed_dict, fetch_list, target_list, options, run_meta
data)
   1318       return self._call_tf_sessionrun(
-> 1319           options, feed_dict, fetch_list, target_list, run_metadata)
   1320 

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _call_tf_sessionrun(self, options, feed_dict, fetch_list, tar
get_list, run_metadata)
   1406         self._session, options, feed_dict, fetch_list, target_list,
-> 1407         run_metadata)
   1408 

FailedPreconditionError: Attempting to use uninitialized value Variable_1
	 [[{{node _retval_Variable_1_0_0}}]]

During handling of the above exception, another exception occurred:

FailedPreconditionError                   Traceback (most recent call last)
<ipython-input-14-9b29fd6ffb41> in <module>
      3 lucky_number = tf.Variable(24)
      4 with tf.Session() as sess:
----> 5     print(sess.run(lucky_number))

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in run(self, fetches, feed_dict, options, run_metadata)
    927     try:
    928       result = self._run(None, fetches, feed_dict, options_ptr,
--> 929                          run_metadata_ptr)
    930       if run_metadata:
    931         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _run(self, handle, fetches, feed_dict, options, run_metadata)
   1150     if final_fetches or final_targets or (handle and feed_dict_tensor)
:
   1151       results = self._do_run(handle, final_targets, final_fetches,
-> 1152                              feed_dict_tensor, options, run_metadata)
   1153     else:
   1154       results = []

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _do_run(self, handle, target_list, fetch_list, feed_dict, opt
ions, run_metadata)
   1326     if handle is None:
   1327       return self._do_call(_run_fn, feeds, fetches, targets, options,
-> 1328                            run_metadata)
   1329     else:
   1330       return self._do_call(_prun_fn, handle, feeds, fetches)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _do_call(self, fn, *args)
   1346           pass
   1347       message = error_interpolation.interpolate(message, self._graph)
-> 1348       raise type(e)(node_def, op, message)
   1349 
   1350   def _extend_graph(self):
```

該如何修正呢?

將變數張量的 `initializer` 屬性放入 Session 中執行

```
In [15]: import tensorflow as tf

         lucky_number = tf.Variable(24)
         with tf.Session() as sess:
             sess.run(lucky_number.initializer)
             print(sess.run(lucky_number))
```

```
24
```

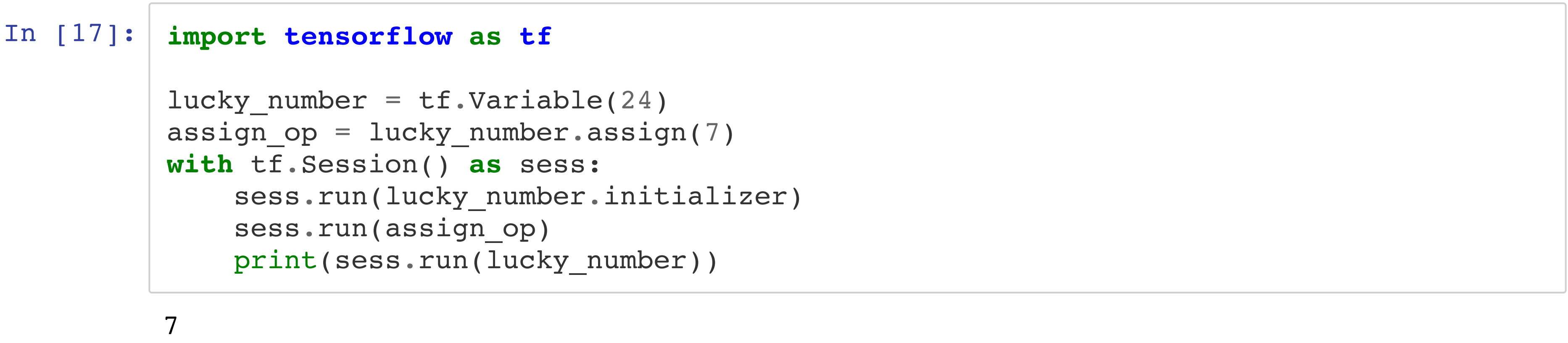
透過 .assign() 賦值

In [16]:

```
import tensorflow as tf

lucky_number = tf.Variable(24)
assign_op = lucky_number.assign(7)
with tf.Session() as sess:
    sess.run(lucky_number.initializer)
    # sess.run(assign_op)
    print(sess.run(lucky_number))
```

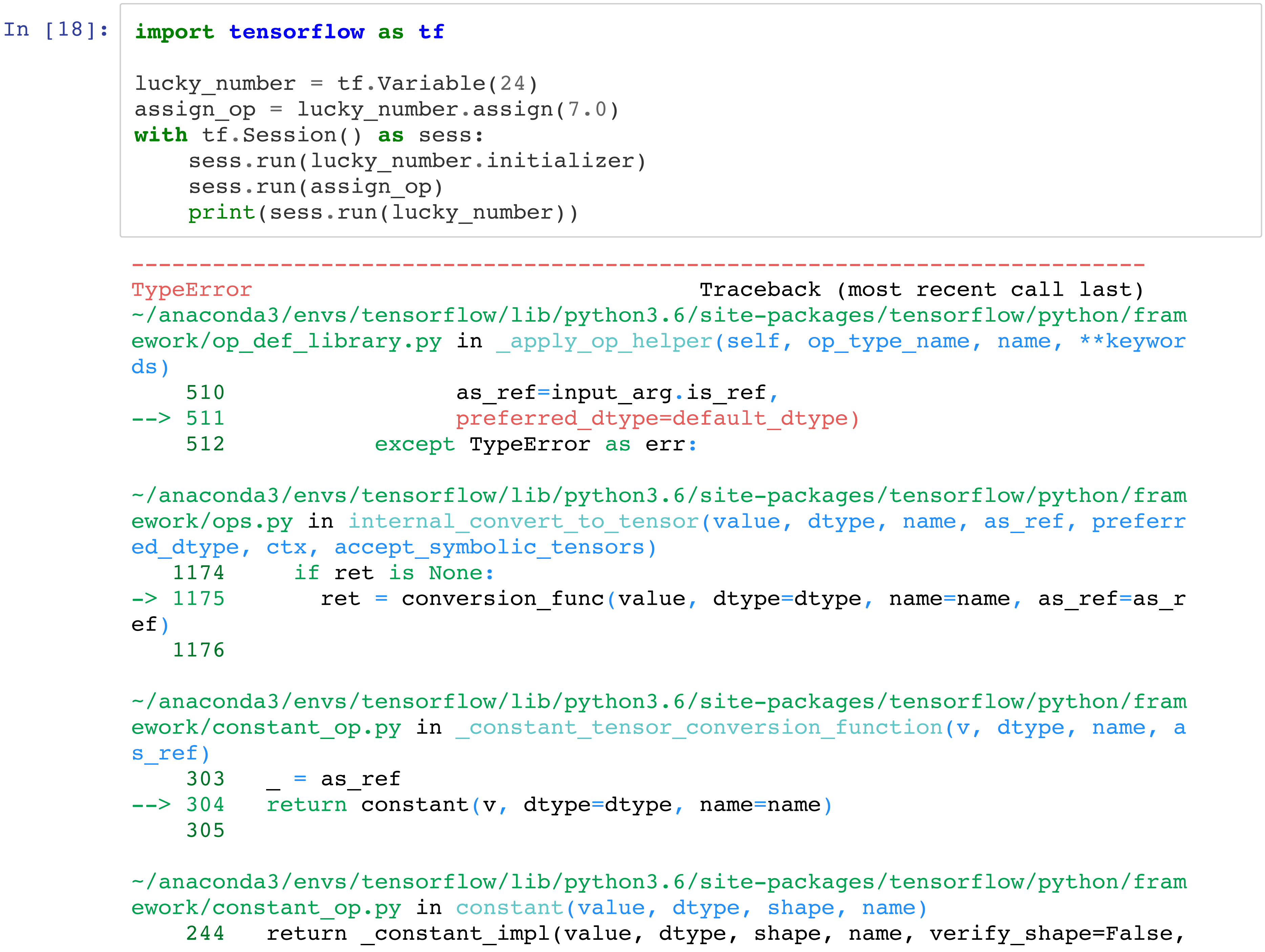
```
24
```

重新賦值也是一種運算（operation），必須放入 Session 執行



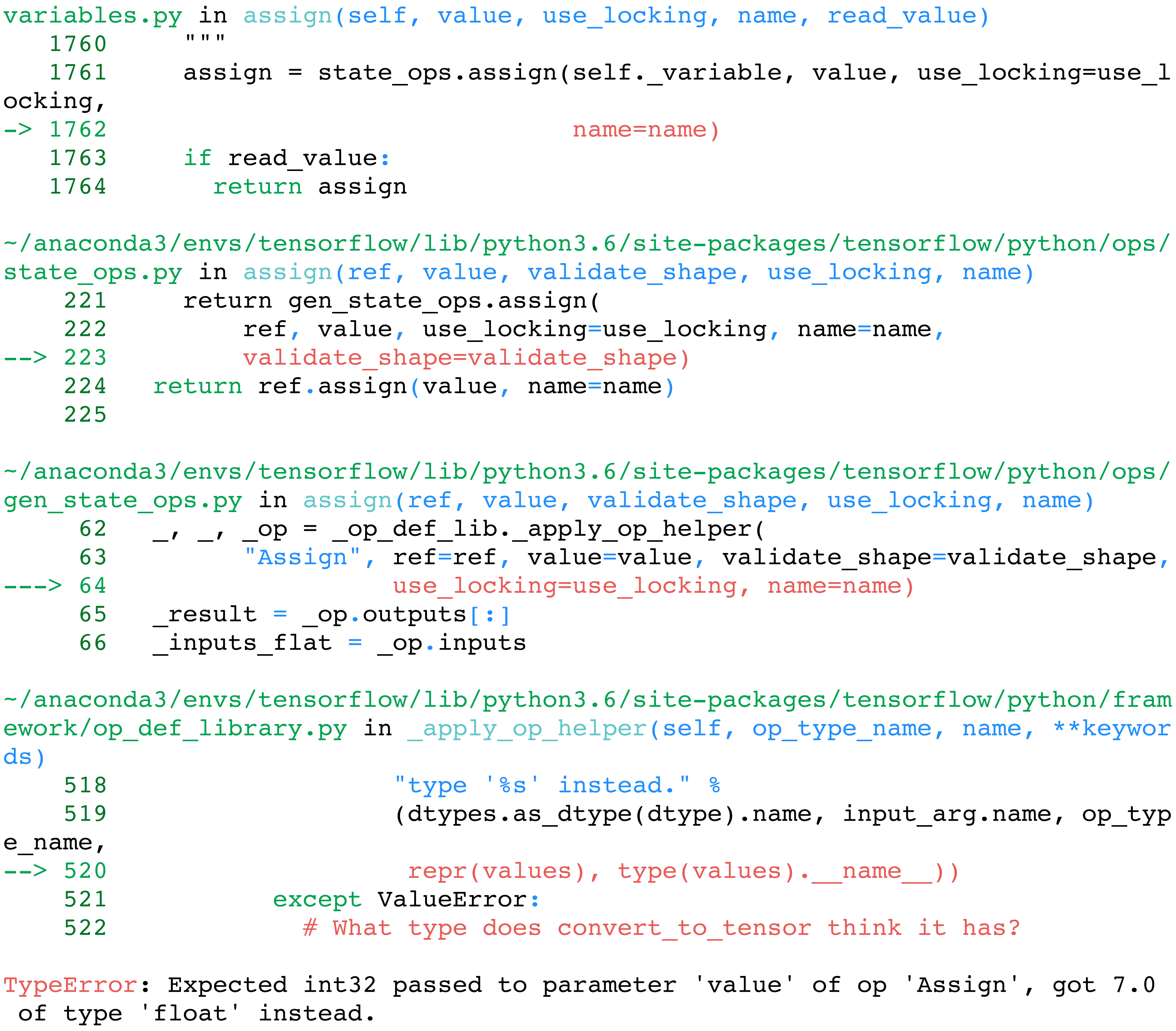
```
In [17]: import tensorflow as tf

         lucky_number = tf.Variable(24)
         assign_op = lucky_number.assign(7)
         with tf.Session() as sess:
             sess.run(lucky_number.initializer)
             sess.run(assign_op)
             print(sess.run(lucky_number))

         7
```

變數張量被宣告之後，重新賦值時必須要注意類型



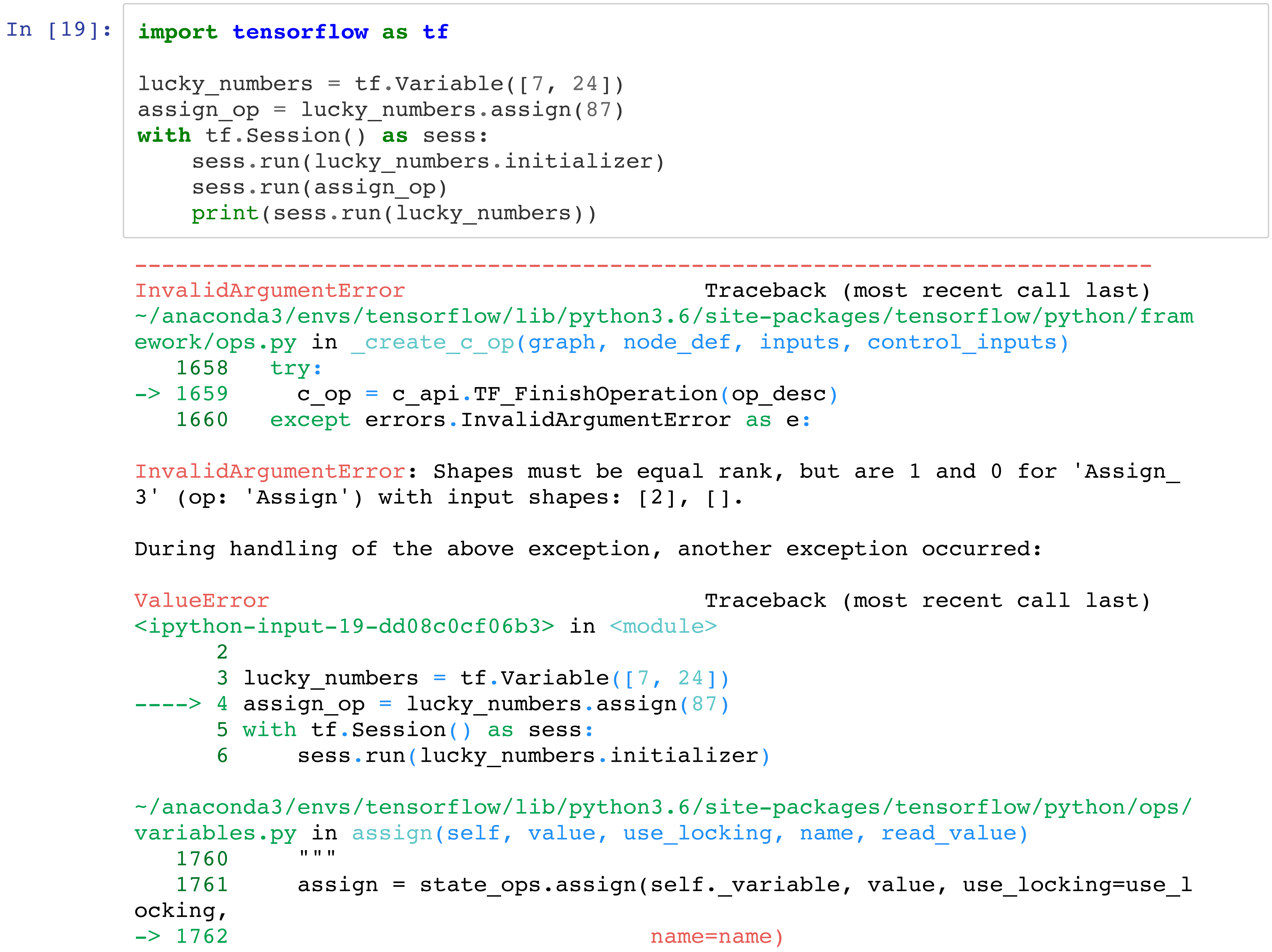
```
In [18]: import tensorflow as tf

         lucky_number = tf.Variable(24)
         assign_op = lucky_number.assign(7.0)
         with tf.Session() as sess:
             sess.run(lucky_number.initializer)
             sess.run(assign_op)
             print(sess.run(lucky_number))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/op_def_library.py in _apply_op_helper(self, op_type_name, name, **keywor
ds)
    510                 as_ref=input_arg.is_ref,
--> 511                 preferred_dtype=default_dtype)
    512           except TypeError as err:

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/ops.py in internal_convert_to_tensor(value, dtype, name, as_ref, preferr
ed_dtype, ctx, accept_symbolic_tensors)
   1174     if ret is None:
-> 1175       ret = conversion_func(value, dtype=dtype, name=name, as_ref=as_r
ef)
   1176 

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/constant_op.py in _constant_tensor_conversion_function(v, dtype, name, a
s_ref)
    303   _ = as_ref
--> 304   return constant(v, dtype=dtype, name=name)
    305 

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/constant_op.py in constant(value, dtype, shape, name)
    244   return _constant_impl(value, dtype, shape, name, verify_shape=False,
--> 245                               allow_broadcast=True)
    246

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/constant_op.py in _constant_impl(value, dtype, shape, name, verify_shap
e, allow_broadcast)
    282                 value, dtype=dtype, shape=shape, verify_shape=verify_shape,
--> 283                 allow_broadcast=allow_broadcast))
    284         dtype_value = attr_value_pb2.AttrValue(type=tensor_value.tensor.dtyp
e)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/tensor_util.py in make_tensor_proto(values, dtype, shape, verify_shape,
 allow_broadcast)
    465     else:
--> 466       _AssertCompatible(values, dtype)
    467       nparray = np.array(values, dtype=np_dt)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/tensor_util.py in _AssertCompatible(values, dtype)
    370       raise TypeError("Expected %s, got %s of type '%s' instead." %
--> 371                       (dtype.name, repr(mismatch), type(mismatch).__na
me__))
    372

TypeError: Expected int32, got 7.0 of type 'float' instead.

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call last)
<ipython-input-18-13fde7641243> in <module>
      2
      3 lucky_number = tf.Variable(24)
----> 4 assign_op = lucky_number.assign(7.0)
      5 with tf.Session() as sess:
      6     sess.run(lucky_number.initializer)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/ops/
```



```
variables.py in assign(self, value, use_locking, name, read_value)
   1760     """
   1761     assign = state_ops.assign(self._variable, value, use_locking=use_l
ocking,
-> 1762                               name=name)
   1763     if read_value:
   1764       return assign

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/ops/
state_ops.py in assign(ref, value, validate_shape, use_locking, name)
    221     return gen_state_ops.assign(
    222         ref, value, use_locking=use_locking, name=name,
--> 223         validate_shape=validate_shape)
    224   return ref.assign(value, name=name)
    225 

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/ops/
gen_state_ops.py in assign(ref, value, validate_shape, use_locking, name)
     62   _, _, _op = _op_def_lib._apply_op_helper(
     63         "Assign", ref=ref, value=value, validate_shape=validate_shape,
---> 64                   use_locking=use_locking, name=name)
     65   _result = _op.outputs[:]
     66   _inputs_flat = _op.inputs

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/op_def_library.py in _apply_op_helper(self, op_type_name, name, **keywor
ds)
    518                   "type '%s' instead." %
    519                   (dtypes.as_dtype(dtype).name, input_arg.name, op_typ
e_name,
--> 520                    repr(values), type(values).__name__))
    521           except ValueError:
    522             # What type does convert_to_tensor think it has?

TypeError: Expected int32 passed to parameter 'value' of op 'Assign', got 7.0
 of type 'float' instead.
```

變數張量被宣告之後，重新賦值時必須要注意外觀



In [19]:

```python
import tensorflow as tf

lucky_numbers = tf.Variable([7, 24])
assign_op = lucky_numbers.assign(87)
with tf.Session() as sess:
    sess.run(lucky_numbers.initializer)
    sess.run(assign_op)
    print(sess.run(lucky_numbers))
```

```
---------------------------------------------------------------------------
InvalidArgumentError                      Traceback (most recent call last)
~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/ops.py in _create_c_op(graph, node_def, inputs, control_inputs)
   1658   try:
-> 1659     c_op = c_api.TF_FinishOperation(op_desc)
   1660   except errors.InvalidArgumentError as e:

InvalidArgumentError: Shapes must be equal rank, but are 1 and 0 for 'Assign_
3' (op: 'Assign') with input shapes: [2], [].

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
<ipython-input-19-dd08c0cf06b3> in <module>
      2 
      3 lucky_numbers = tf.Variable([7, 24])
----> 4 assign_op = lucky_numbers.assign(87)
      5 with tf.Session() as sess:
      6     sess.run(lucky_numbers.initializer)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/ops/
variables.py in assign(self, value, use_locking, name, read_value)
   1760     """
   1761     assign = state_ops.assign(self._variable, value, use_locking=use_l
ocking,
-> 1762                               name=name)
```

```
   1763     if read_value:
   1764       return assign

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/ops/
state_ops.py in assign(ref, value, validate_shape, use_locking, name)
    221     return gen_state_ops.assign(
    222         ref, value, use_locking=use_locking, name=name,
--> 223         validate_shape=validate_shape)
    224   return ref.assign(value, name=name)
    225

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/ops/
gen_state_ops.py in assign(ref, value, validate_shape, use_locking, name)
     62   _, _, _op = _op_def_lib._apply_op_helper(
     63         "Assign", ref=ref, value=value, validate_shape=validate_shape,
---> 64                   use_locking=use_locking, name=name)
     65   _result = _op.outputs[:]
     66   _inputs_flat = _op.inputs

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/op_def_library.py in _apply_op_helper(self, op_type_name, name, **keywor
ds)
    786         op = g.create_op(op_type_name, inputs, output_types, name=scop
e,
    787                          input_types=input_types, attrs=attr_protos,
--> 788                          op_def=op_def)
    789       return output_structure, op_def.is_stateful, op
    790

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/uti
l/deprecation.py in new_func(*args, **kwargs)
    505                 'in a future version' if date is None else ('after %s'
% date),
    506                 instructions)
--> 507       return func(*args, **kwargs)
    508
    509     doc = _add_deprecated_arg_notice_to_docstring(
~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/ops.py in create_op(***failed resolving arguments***)
   3298             input_types=input_types,
   3299             original_op=self._default_original_op,
-> 3300             op_def=op_def)
   3301         self._create_op_helper(ret, compute_device=compute_device)
   3302       return ret

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/ops.py in __init__(self, node_def, g, inputs, output_types, control_inpu
ts, input_types, original_op, op_def)
   1821             op_def, inputs, node_def.attr)
   1822         self._c_op = _create_c_op(self._graph, node_def, grouped_inputs,
-> 1823                                   control_input_ops)
   1824 
   1825       # Initialize self._outputs.

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/fram
ework/ops.py in _create_c_op(graph, node_def, inputs, control_inputs)
   1660     except errors.InvalidArgumentError as e:
   1661       # Convert to ValueError for backwards compatibility.
-> 1662       raise ValueError(str(e))
   1663 
   1664     return c_op

ValueError: Shapes must be equal rank, but are 1 and 0 for 'Assign_3' (op: 'As
sign') with input shapes: [2], [].
```

TensorFlow 外觀的註記與 ndarray 相同

- 零維：`()`
- 一維：`(m,)`
- 二維：`(m, n)`
- 三維：`(q, m, n)`

In [20]:

```
import numpy as np

zero_d = np.array(24)
one_d = np.arange(24)
two_d = np.arange(24).reshape(6, 4)
three_d = np.arange(24).reshape(2, 3, 4)
print(zero_d.shape)
print(one_d.shape)
print(two_d.shape)
print(three_d.shape)
```

```
()
(24,)
(6, 4)
(2, 3, 4)
```

可以使用 .get_shape() 確認 Tensor 外觀

In [21]:

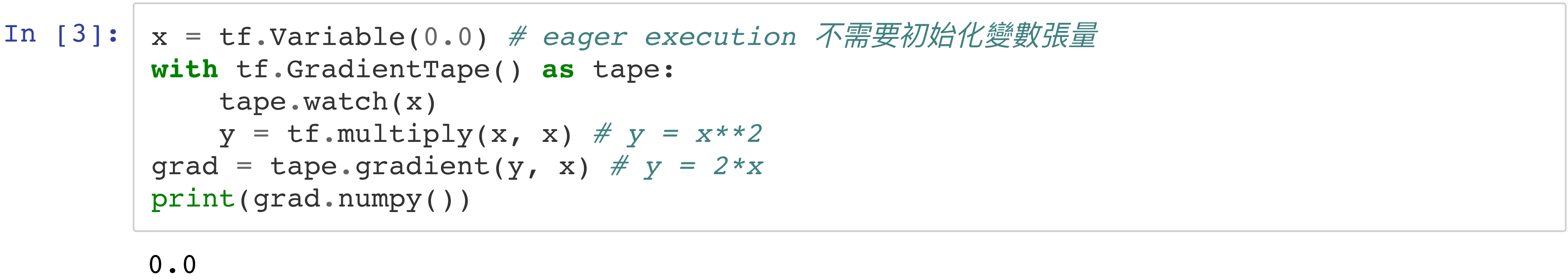
```
import tensorflow as tf

zero_d = tf.Variable(24)
one_d = tf.reshape(tf.Variable(tf.range(24)), (24,))
two_d = tf.reshape(tf.Variable(tf.range(24)), (6, 4))
three_d = tf.reshape(tf.Variable(tf.range(24)), (2, 3, 4))
print(zero_d.get_shape())
print(one_d.get_shape())
print(two_d.get_shape())
print(three_d.get_shape())
```

```
()
(24,)
(6, 4)
(2, 3, 4)
```

tf.GradientTape() 計算特定資料點的斜率

```
In [2]: # 如果想要跑以下三個儲存格，必須重新啟動 Kernel
        import tensorflow as tf

        tf.enable_eager_execution()
        x = tf.Variable(1.0) # eager execution 不需要初始化變數張量
        with tf.GradientTape() as tape:
            tape.watch(x)
            y = tf.multiply(x, x) # y = x**2
        grad = tape.gradient(y, x) # y = 2*x
        print(grad.numpy())
```

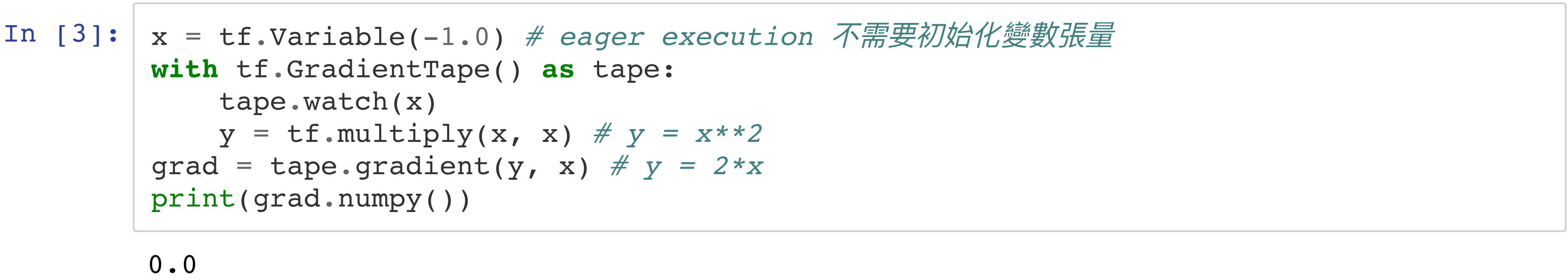
```
2.0
```



In [3]:

```
x = tf.Variable(0.0) # eager execution 不需要初始化變數張量
with tf.GradientTape() as tape:
    tape.watch(x)
    y = tf.multiply(x, x) # y = x**2
grad = tape.gradient(y, x) # y = 2*x
print(grad.numpy())
```

```
0.0
```



In [3]:

```
x = tf.Variable(-1.0) # eager execution 不需要初始化變數張量
with tf.GradientTape() as tape:
    tape.watch(x)
    y = tf.multiply(x, x) # y = x**2
grad = tape.gradient(y, x) # y = 2*x
print(grad.numpy())
```

```
0.0
```

隨堂練習

以 `tf.GradientTape()` 與 eager execution 計算 $y = 2x - 1$ 在 x 為 1, 0, -1 的梯度

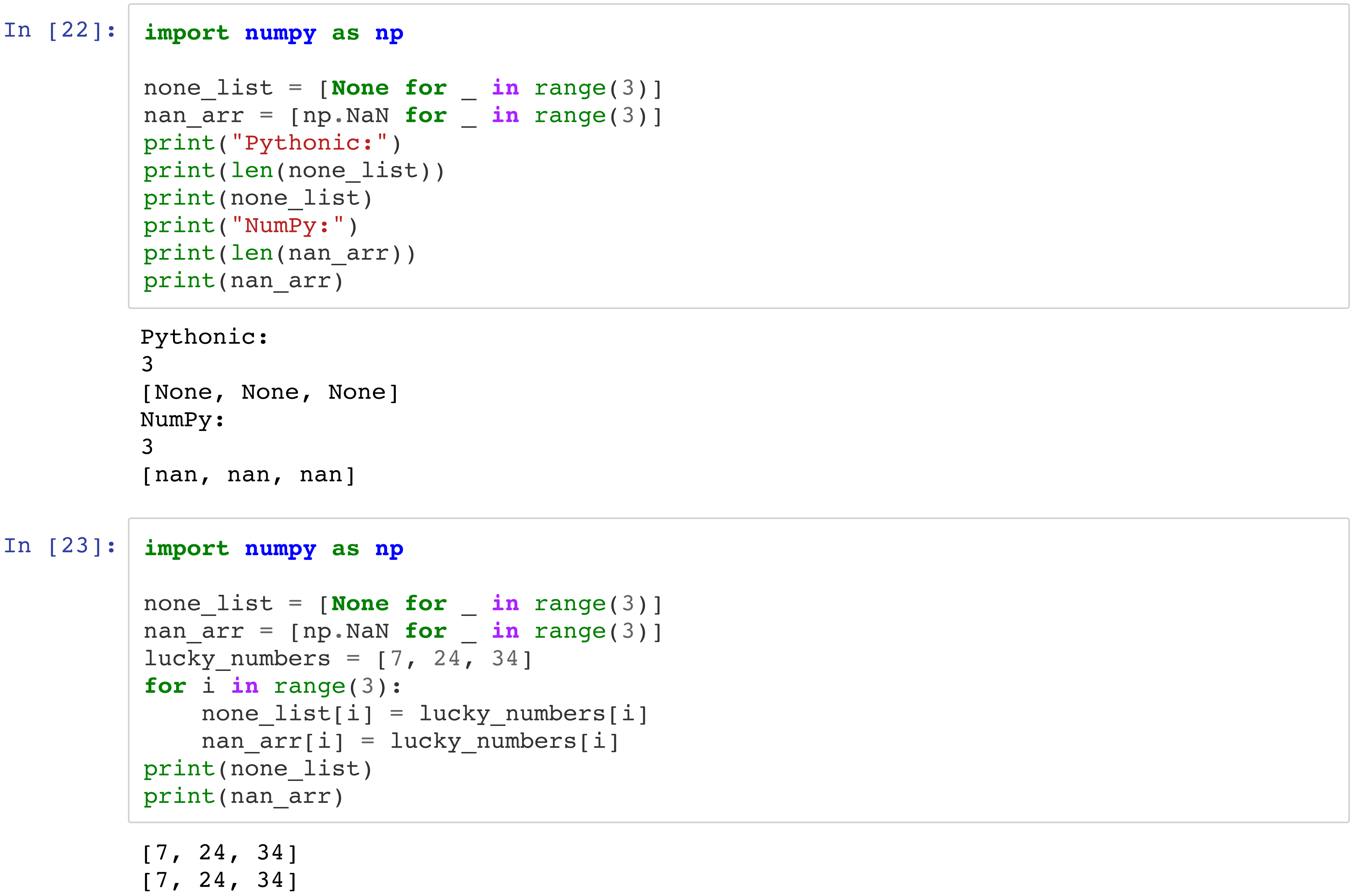
Placeholders

以 **`tf.placeholder()`** 宣告

Placeholder 作為 TensorFlow 模型的資料輸入口

- Feed dict 資料要以 Python dict 的格式餵入
- Fetch 是模型運算的輸出，類型是 `ndarray`

可以把 Placeholder 想成像是 `None` 或 `np.NaN`



In [22]:

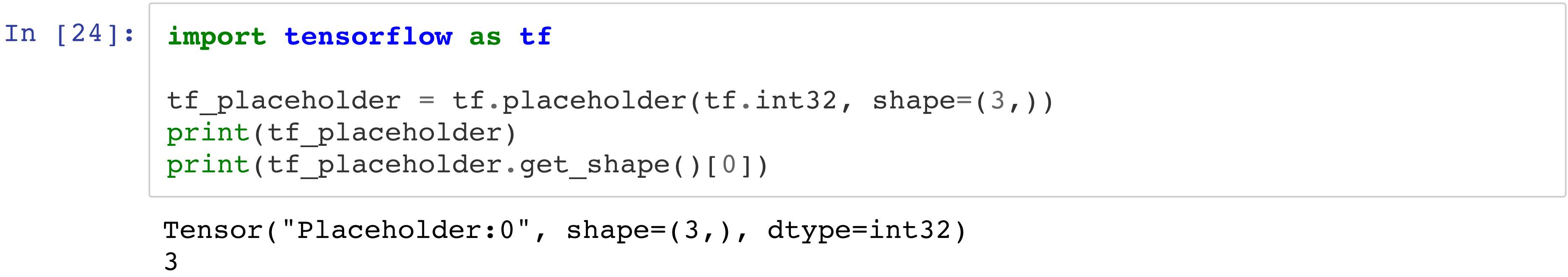
```
import numpy as np

none_list = [None for _ in range(3)]
nan_arr = [np.NaN for _ in range(3)]
print("Pythonic:")
print(len(none_list))
print(none_list)
print("NumPy:")
print(len(nan_arr))
print(nan_arr)
```

```
Pythonic:
3
[None, None, None]
NumPy:
3
[nan, nan, nan]
```

In [23]:

```
import numpy as np

none_list = [None for _ in range(3)]
nan_arr = [np.NaN for _ in range(3)]
lucky_numbers = [7, 24, 34]
for i in range(3):
    none_list[i] = lucky_numbers[i]
    nan_arr[i] = lucky_numbers[i]
print(none_list)
print(nan_arr)
```

```
[7, 24, 34]
[7, 24, 34]
```



In [24]:

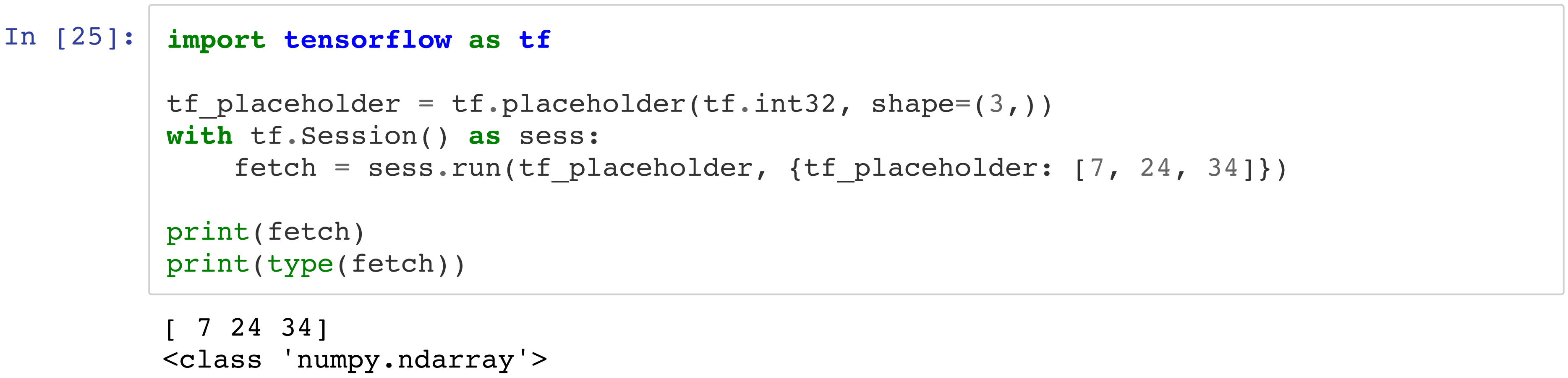
```
import tensorflow as tf

tf_placeholder = tf.placeholder(tf.int32, shape=(3,))
print(tf_placeholder)
print(tf_placeholder.get_shape()[0])
```

```
Tensor("Placeholder:0", shape=(3,), dtype=int32)
3
```

將資料以 dict 餵入 placeholder

語法為：

```
import tensorflow as tf

my_ph = tf.placeholder(...)
my_op = ...
feed_dict = {
  my_ph: ...
}
with tf.Session() as sess:
  fetch = sess.run(my_op, feed_dict)
```



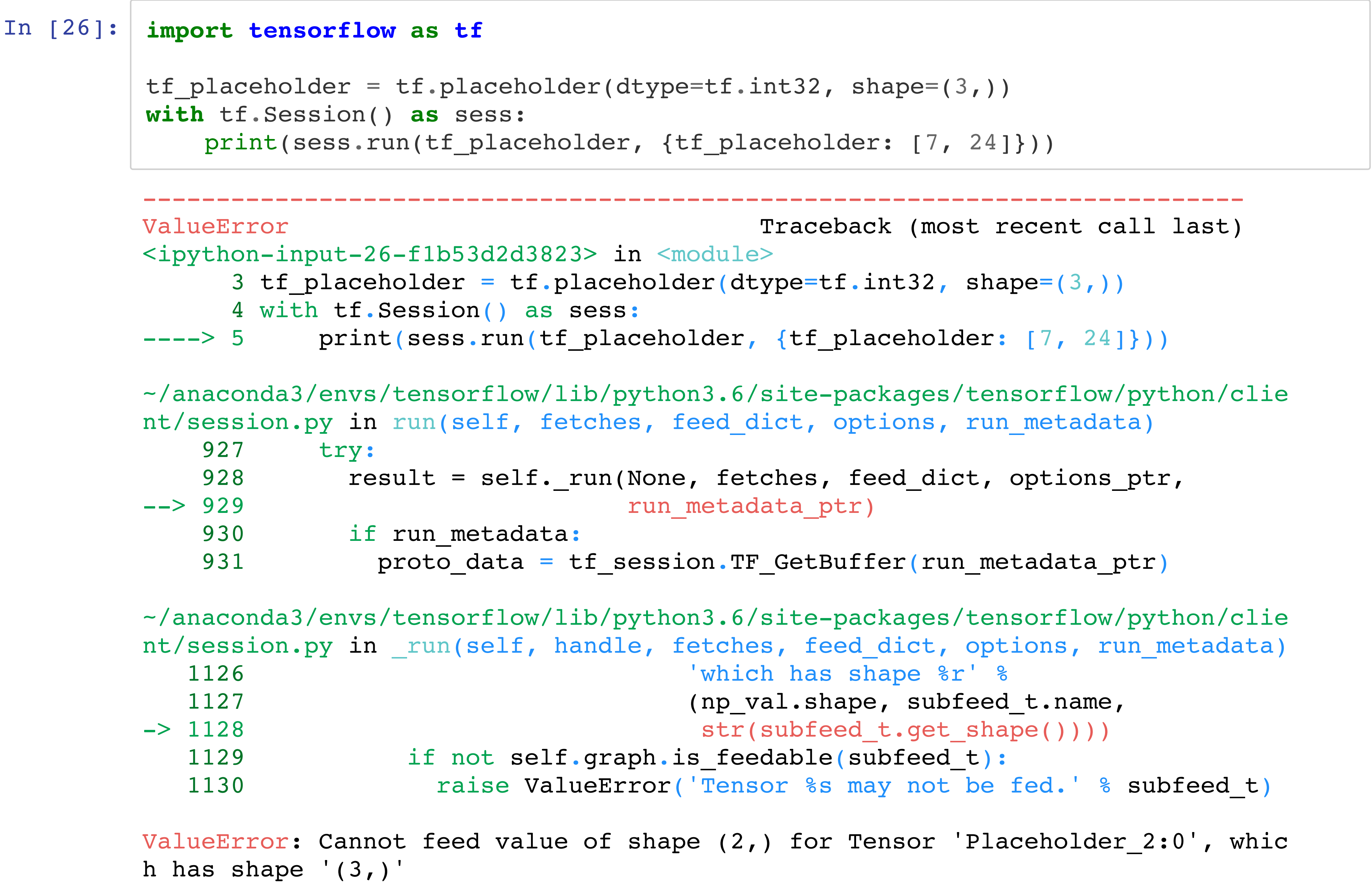
In [25]:

```
import tensorflow as tf

tf_placeholder = tf.placeholder(tf.int32, shape=(3,))
with tf.Session() as sess:
    fetch = sess.run(tf_placeholder, {tf_placeholder: [7, 24, 34]})

print(fetch)
print(type(fetch))
```

```
[ 7 24 34]
<class 'numpy.ndarray'>
```

Placeholders 也很嚴謹

不同的外觀不能餵入



```
In [26]: import tensorflow as tf

         tf_placeholder = tf.placeholder(dtype=tf.int32, shape=(3,))
         with tf.Session() as sess:
             print(sess.run(tf_placeholder, {tf_placeholder: [7, 24]}))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-26-f1b53d2d3823> in <module>
      3 tf_placeholder = tf.placeholder(dtype=tf.int32, shape=(3,))
      4 with tf.Session() as sess:
----> 5     print(sess.run(tf_placeholder, {tf_placeholder: [7, 24]}))

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in run(self, fetches, feed_dict, options, run_metadata)
    927     try:
    928       result = self._run(None, fetches, feed_dict, options_ptr,
--> 929                          run_metadata_ptr)
    930       if run_metadata:
    931         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _run(self, handle, fetches, feed_dict, options, run_metadata)
   1126                              'which has shape %r' %
   1127                              (np_val.shape, subfeed_t.name,
-> 1128                               str(subfeed_t.get_shape())))
   1129           if not self.graph.is_feedable(subfeed_t):
   1130             raise ValueError('Tensor %s may not be fed.' % subfeed_t)

ValueError: Cannot feed value of shape (2,) for Tensor 'Placeholder_2:0', whic
h has shape '(3,)'
```

Placeholders 也很嚴謹

外觀相同、不同的資料類型則會做隱性轉換

```
In [27]:
import tensorflow as tf

tf_placeholder = tf.placeholder(dtype=tf.int32, shape=(3,))
with tf.Session() as sess:
    print(sess.run(tf_placeholder, {tf_placeholder: [7.0, 24.0, 34.0]}))
```

```
[ 7 24 34]
```

隨堂練習

以 TensorFlow 的 placeholder 張量將五組華氏溫度轉換為攝氏溫度。

$$C = \frac{(F - 32) \times 5}{9}$$

In [28]:

```
city_temps_f = [59, 30, 45, 28, 47] # Taipei, New York, London, Reykjavik, Tokyo
```

隨堂練習

以 TensorFlow 的 placeholder 張量計算五個球員的 BMI

$$BMI = \frac{weight_{kg}}{height_m^2}$$

In [30]:

```
# Shaq, Dirk, LeBron, MJ, Nash
player_heights = [216, 213, 203, 198, 191]
player_weights = [147, 111, 113, 98, 82]
```