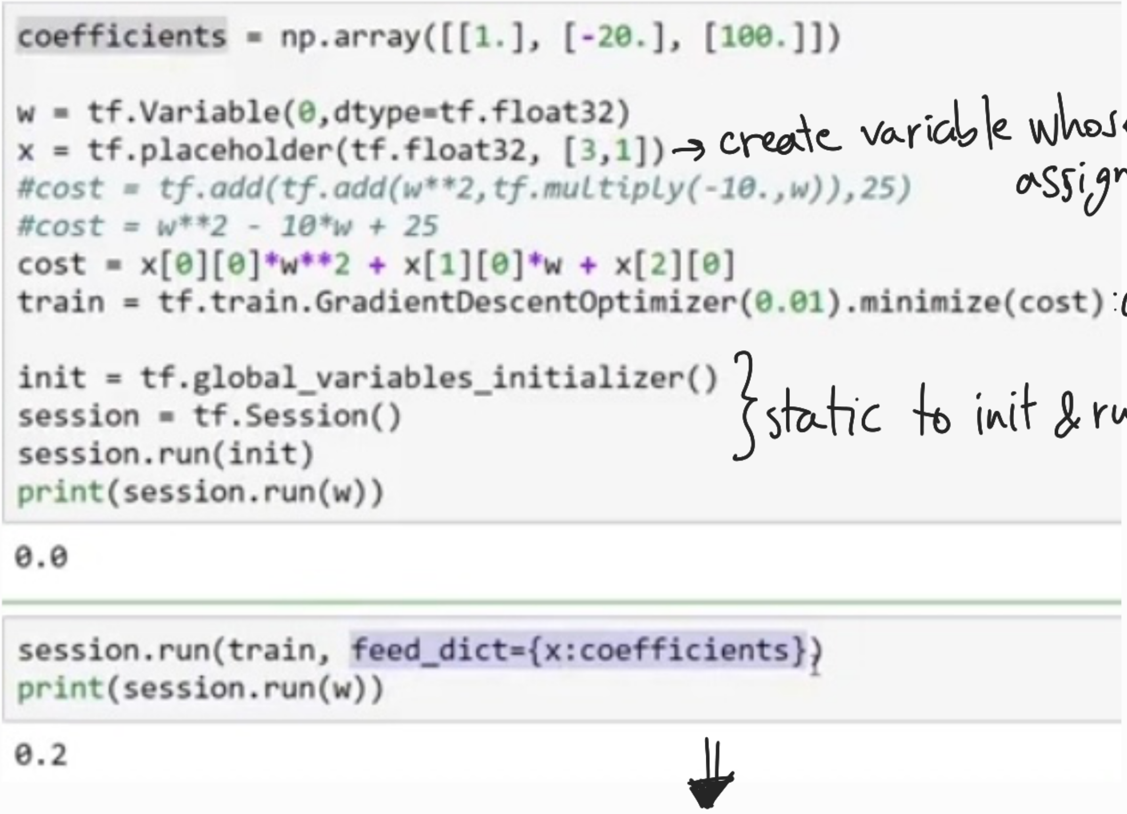


```
coefficients = np.array([[1.], [-20.], [100.]])

w = tf.Variable(0,dtype=tf.float32)
x = tf.placeholder(tf.float32, [3,1])
#cost = tf.add(tf.add(w**2,tf.multiply(-10.,w)),25)
#cost = w**2 - 10*w + 25
cost = x[0][0]*w**2 + x[1][0]*w + x[2][0]
train = tf.train.GradientDescentOptimizer(0.01).minimize(cost)

init = tf.global_variables_initializer()
session = tf.Session()
session.run(init)
print(session.run(w))
```

```
0.0
```

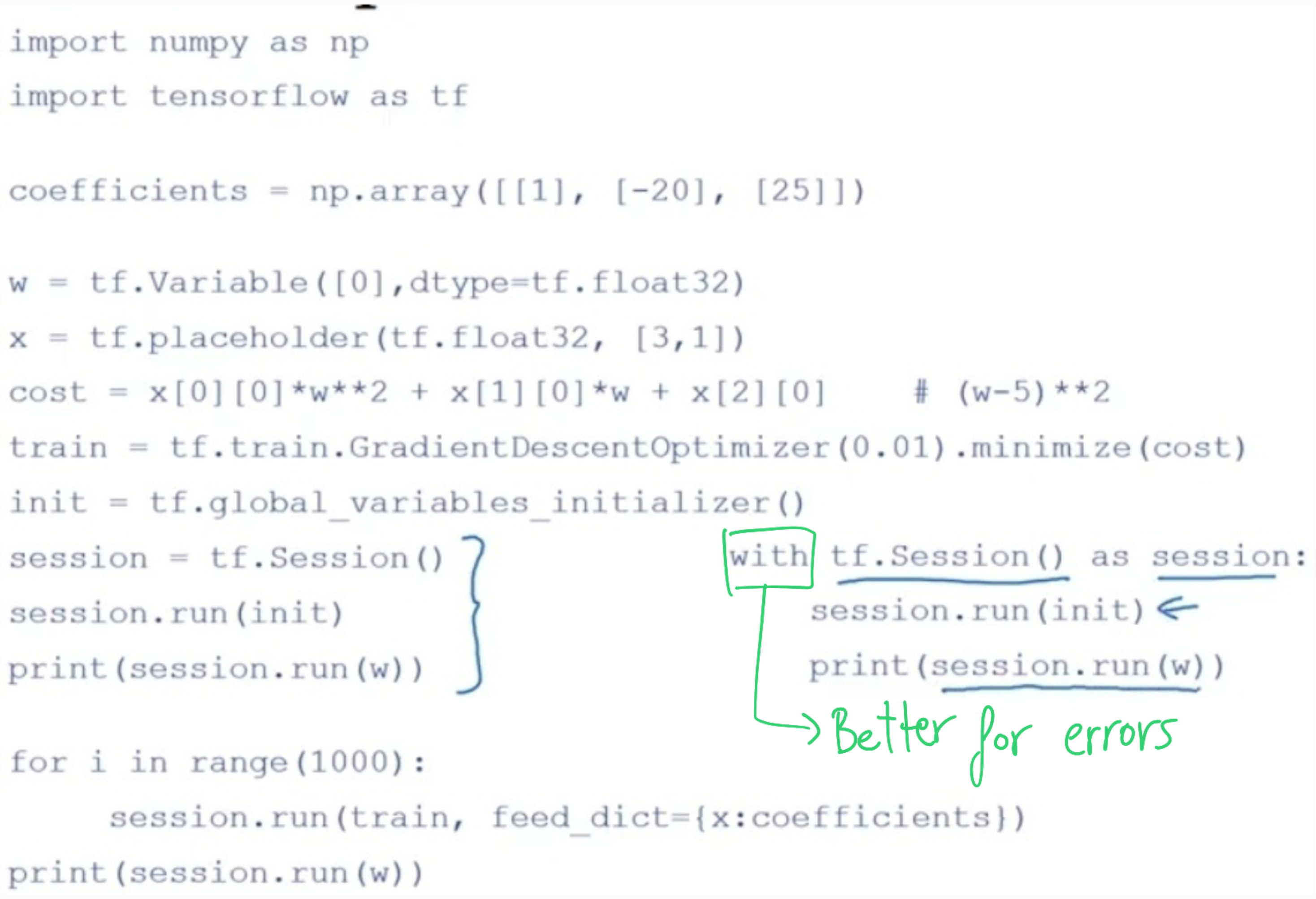
```
session.run(train, feed_dict={x:coefficients})
print(session.run(w))
```

```
0.2
```

→ create variable whose value will be assigned later

algorithm

} static to init & run



```
import numpy as np
import tensorflow as tf

coefficients = np.array([[1], [-20], [25]])

w = tf.Variable([0],dtype=tf.float32)
x = tf.placeholder(tf.float32, [3,1])
cost = x[0][0]*w**2 + x[1][0]*w + x[2][0]     # (w-5)**2
train = tf.train.GradientDescentOptimizer(0.01).minimize(cost)
init = tf.global_variables_initializer()
session = tf.Session()
session.run(init)
print(session.run(w))

for i in range(1000):
    session.run(train, feed_dict={x:coefficients})
print(session.run(w))
```

```
with tf.Session() as session:
    session.run(init)
    print(session.run(w))
```

Better for errors