tf.train.import_meta_graph

```
import_meta_graph(
    meta_graph_or_file,
    clear_devices=False,
    import_scope=None,
    **kwargs
)
```

Defined in `tensorflow/python/training/saver.py`.

See the guides: Exporting and Importing a MetaGraph, Variables > Exporting and Importing Meta Graphs

Recreates a Graph saved in a `MetaGraphDef` proto.

This function takes a `MetaGraphDef` protocol buffer as input. If the argument is a file containing a `MetaGraphDef` protocol buffer , it constructs a protocol buffer from the file content. The function then adds all the nodes from the `graph_def` field to the current graph, recreates all the collections, and returns a saver constructed from the `saver_def` field.

In combination with `export_meta_graph()`, this function can be used to

- Serialize a graph along with other Python objects such as `QueueRunner`, `Variable` into a `MetaGraphDef`.
- Restart training from a saved graph and checkpoints.
- Run inference from a saved graph and checkpoints.

```
...
# Create a saver.
saver = tf.train.Saver(...variables...)
# Remember the training_op we want to run by adding it to a collection.
tf.add_to_collection('train_op', train_op)
sess = tf.Session()
for step in xrange(1000000):
    sess.run(train_op)
    if step % 1000 == 0:
        # Saves checkpoint, which by default also exports a meta_graph
        # named 'my-model-global_step.meta'.
        saver.save(sess, 'my-model', global_step=step)
```

Later we can continue training from this saved `meta_graph` without building the model from scratch.

```
with tf.Session() as sess:
  new_saver = tf.train.import_meta_graph('my-save-dir/my-model-10000.meta')
  new_saver.restore(sess, 'my-save-dir/my-model-10000')
  # tf.get_collection() returns a list. In this example we only want the
  # first one.
  train_op = tf.get_collection('train_op')[0]
  for step in xrange(1000000):
    sess.run(train_op)
```

NOTE: Restarting training from saved `meta_graph` only works if the device assignments have not changed.

Args:

- `meta_graph_or_file`: `MetaGraphDef` protocol buffer or filename (including the path) containing a `MetaGraphDef`.
- `clear_devices`: Whether or not to clear the device field for an `Operation` or `Tensor` during import.
- `import_scope`: Optional `string`. Name scope to add. Only used when initializing from protocol buffer.
- `**kwargs`: Optional keyed arguments.

Returns:

A saver constructed from `saver_def` in `MetaGraphDef` or None.

A None value is returned if no variables exist in the `MetaGraphDef` (i.e., there are no variables to restore).
