```
In [93]: clear
```

```
In [94]: df1 = DataFrame({'key':['b', 'b', 'a', 'c', 'a', 'a', 'b'], 'data1':range(7)})

In [95]: df1 = DataFrame({'key':['a', 'b', 'd'], 'data2':range(3)})

In [96]: df1
Out[96]:
   data2 key
0      0   a
```

```
1      1   b
2      2   d

In [97]: df1 = DataFrame({'key':['b', 'b', 'a', 'c', 'a', 'a', 'b'], 'data1':range(7)})

In [98]: df2 = DataFrame({'key':['a', 'b', 'd'], 'data2':range(3)})

In [99]: df1
Out[99]:
   data1 key
0      0   b
1      1   b
2      2   a
3      3   c
4      4   a
5      5   a
6      6   b

In [100]: df2
Out[100]:
   data2 key
0      0   a
1      1   b
2      2   d

In [101]: import pandas as pd

In [102]: pd.merge(df1, df2)
Out[102]:
   data1 key  data2
0      0   b      1
1      1   b      1
2      6   b      1
3      2   a      0
4      4   a      0
5      5   a      0

In [103]: df3 = DataFrame({'lkey': ['b','b','a','c','a','a','b'], 'data1':range(7)})

In [104]: df4 = DataFrame({'rkey':['a','b','d'],})

In [105]: df4 = DataFrame({'rkey':['a','b','d'],
     ...: 'data2':range(7)})
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-105-03b5adc5f720> in <module>()
      1 df4 = DataFrame({'rkey':['a','b','d'],
----> 2 'data2':range(7)})

~/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in __init__(self, data, index, columns,
dtype, copy)
    328                                 dtype=dtype, copy=copy)
    329         elif isinstance(data, dict):
--> 330             mgr = self._init_dict(data, index, columns, dtype=dtype)
    331         elif isinstance(data, ma.MaskedArray):
    332             import numpy.ma.mrecords as mrecords

~/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in _init_dict(self, data, index, columns
, dtype)
    459             arrays = [data[k] for k in keys]
    460
--> 461         return _arrays_to_mgr(arrays, data_names, index, columns, dtype=dtype)
    462
    463     def _init_ndarray(self, values, index, columns, dtype=None, copy=False):

~/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in _arrays_to_mgr(arrays, arr_names, ind
ex, columns, dtype)
   6161     # figure out the index, if necessary
```

```
   6162     if index is None:
-> 6163         index = extract_index(arrays)
   6164     else:
   6165         index = _ensure_index(index)

~/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in extract_index(data)
   6209             lengths = list(set(raw_lengths))
   6210             if len(lengths) > 1:
-> 6211                 raise ValueError('arrays must all be same length')
   6212
   6213             if have_dicts:

ValueError: arrays must all be same length

In [106]: df4 = DataFrame({'rkey':['a','b','d'],
     ...: 'data2':range(3
     ...: )})

In [107]: pd.merge(df3, df4, left_on='lkey', right_on='rkey')
Out[107]:
   data1 lkey  data2 rkey
0      0    b      1    b
1      1    b      1    b
2      6    b      1    b
3      2    a      0    a
4      4    a      0    a
5      5    a      0    a

In [108]: df3
Out[108]:
   data1 lkey
0      0    b
1      1    b
2      2    a
3      3    c
4      4    a
5      5    a
6      6    b

In [109]: df4
Out[109]:
   data2 rkey
0      0    a
1      1    b
2      2    d

In [110]: pd.merge(df1, df2, how='outer')
Out[110]:
   data1 key  data2
0    0.0   b    1.0
1    1.0   b    1.0
2    6.0   b    1.0
3    2.0   a    0.0
4    4.0   a    0.0
5    5.0   a    0.0
6    3.0   c    NaN
7    NaN   d    2.0

In [111]: pd.merge?

In [112]: pd.merge?

In [113]: df1 = DataFrame({'key': ['b', 'b', 'a', 'c', 'a', 'b'], 'data1': range(6)})

In [114]: df2 = DataFrame({'key': ['a', 'b', 'a', 'b', 'd'], 'data2': range(5)})

In [115]: df1
Out[115]:
```

```
   data1 key
0      0   b
1      1   b
2      2   a
3      3   c
4      4   a
5      5   b

In [116]: df2
Out[116]:
   data2 key
0      0   a
1      1   b
2      2   a
3      3   b
4      4   d

In [117]: pd.merge(df1, df2, on = 'key', how='left')
Out[117]:
    data1 key  data2
0       0   b    1.0
1       0   b    3.0
2       1   b    1.0
3       1   b    3.0
4       2   a    0.0
5       2   a    2.0
6       3   c    NaN
7       4   a    0.0
8       4   a    2.0
9       5   b    1.0
10      5   b    3.0

In [118]: pd.merge(df1, df2)
Out[118]:
   data1 key  data2
0      0   b      1
1      0   b      3
2      1   b      1
3      1   b      3
4      5   b      1
5      5   b      3
6      2   a      0
7      2   a      2
8      4   a      0
9      4   a      2

In [119]: left = DataFrame({'key1': ['foo', 'foo', 'bar'],
     ...: 'key2': ['one', 'two', 'one'],
     ...: 'lval': [1, 2, 3]})

In [120]: left = DataFrame({'key1': ['foo', 'foo', 'bar',
     ...: 'bar'],
     ...: 'key2': ['one', 'one', 'one', 'two'],
     ...:
     ...: 'rval': [4, 5, 6, 7]})

In [121]: left = DataFrame({'key1': ['foo', 'foo', 'bar'],
     ...: 'key2': ['one', 'two', 'one'],
     ...: 'lval': [1, 2, 3]})

In [122]: left = DataFrame({'key1': ['foo', 'foo', 'bar', 'bar'],
     ...: 'key2': ['one', 'one', 'one', 'two'],
     ...: 'rval': [4, 5, 6, 7]})

In [123]: left = DataFrame({'key1': ['foo', 'foo', 'bar'],
     ...: 'key2': ['one', 'two', 'one'],
     ...: 'lval': [1, 2, 3]})
```

```
In [124]: right = DataFrame({'key1': ['foo', 'foo', 'bar', 'bar'],
     ...: 'key2': ['one', 'one', 'one', 'two'],
     ...: 'rval': [4, 5, 6, 7]})

In [125]: pd.merge(left, right, on = ['key1', 'key2'], how = 'outer' )
Out[125]:
  key1 key2  lval  rval
0  foo  one   1.0   4.0
1  foo  one   1.0   5.0
2  foo  two   2.0   NaN
3  bar  one   3.0   6.0
4  bar  two   NaN   7.0

In [126]: left
Out[126]:
  key1 key2  lval
0  foo  one     1
1  foo  two     2
2  bar  one     3

In [127]: right
Out[127]:
  key1 key2  rval
0  foo  one     4
1  foo  one     5
2  bar  one     6
3  bar  two     7

In [128]: pd.merge(left, right, on = 'key1')
Out[128]:
  key1 key2_x  lval key2_y  rval
0  foo    one     1    one     4
1  foo    one     1    one     5
2  foo    two     2    one     4
3  foo    two     2    one     5
4  bar    one     3    one     6
5  bar    one     3    two     7

In [129]: left1 = DataFrame({'key': ['a', 'b', 'a', 'a', 'b', 'c'], 'value': range(6)})

In [130]: right1 = DataFrame({'group_val': [3.5, 7]}, index = ['a', 'b'])

In [131]: left1
Out[131]:
  key  value
0   a      0
1   b      1
2   a      2
3   a      3
4   b      4
5   c      5

In [132]: right1
Out[132]:
   group_val
a        3.5
b        7.0

In [133]: pd.merge(left1, right1, left_on = 'key', right_index = True)
Out[133]:
  key  value  group_val
0   a      0        3.5
2   a      2        3.5
3   a      3        3.5
1   b      1        7.0
4   b      4        7.0

In [134]: pd.merge(left1, right1, left_on = 'key')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-134-d7977d76bd55> in <module>()
----> 1 pd.merge(left1, right1, left_on = 'key')

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in merge(left, right, how, on, l
eft_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
     55                          right_index=right_index, sort=sort, suffixes=suffixes,
     56                          copy=copy, indicator=indicator,
---> 57                          validate=validate)
     58     return op.get_result()
     59

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in __init__(self, left, right, h
ow, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, copy, indicator, validate)
    558             warnings.warn(msg, UserWarning)
    559
--> 560         self._validate_specification()
    561
    562         # note this function has side effects

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in _validate_specification(self)
    975                                     'of levels in the index of "left"')
    976                 self.left_on = [None] * n
--> 977         if len(self.right_on) != len(self.left_on):
    978             raise ValueError("len(right_on) must equal len(left_on)")
    979

TypeError: object of type 'NoneType' has no len()

In [135]: pd.merge(left1, right1, left_on = 'key')
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-135-d7977d76bd55> in <module>()
----> 1 pd.merge(left1, right1, left_on = 'key')

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in merge(left, right, how, on, l
eft_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
     55                          right_index=right_index, sort=sort, suffixes=suffixes,
     56                          copy=copy, indicator=indicator,
---> 57                          validate=validate)
     58     return op.get_result()
     59

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in __init__(self, left, right, h
ow, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, copy, indicator, validate)
    558             warnings.warn(msg, UserWarning)
    559
--> 560         self._validate_specification()
    561
    562         # note this function has side effects

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in _validate_specification(self)
    975                                     'of levels in the index of "left"')
    976                 self.left_on = [None] * n
--> 977         if len(self.right_on) != len(self.left_on):
    978             raise ValueError("len(right_on) must equal len(left_on)")
    979

TypeError: object of type 'NoneType' has no len()

In [136]: pd.merge(left1, right1, left_on = 'key', right_index=0)
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-136-94dc3c1597c3> in <module>()
----> 1 pd.merge(left1, right1, left_on = 'key', right_index=0)

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in merge(left, right, how, on, l
```

```
eft_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
     55                          right_index=right_index, sort=sort, suffixes=suffixes,
     56                          copy=copy, indicator=indicator,
---> 57                          validate=validate)
     58     return op.get_result()
     59

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in __init__(self, left, right, h
ow, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, copy, indicator, validate)
    548             raise ValueError(
    549                 'right_index parameter must be of type bool, not '
--> 550                 '{right_index}'.format(right_index=type(right_index)))
    551
    552         # warn user when merging between different levels

ValueError: right_index parameter must be of type bool, not <class 'int'>

In [137]: pd.merge(left1, right1, left_on = 'key', right_index=False)
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-137-84fb37ca666d> in <module>()
----> 1 pd.merge(left1, right1, left_on = 'key', right_index=False)

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in merge(left, right, how, on, l
eft_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
     55                          right_index=right_index, sort=sort, suffixes=suffixes,
     56                          copy=copy, indicator=indicator,
---> 57                          validate=validate)
     58     return op.get_result()
     59

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in __init__(self, left, right, h
ow, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, copy, indicator, validate)
    558             warnings.warn(msg, UserWarning)
    559
--> 560         self._validate_specification()
    561
    562         # note this function has side effects

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in _validate_specification(self)
    975                                     'of levels in the index of "left"')
    976                 self.left_on = [None] * n
--> 977         if len(self.right_on) != len(self.left_on):
    978             raise ValueError("len(right_on) must equal len(left_on)")
    979

TypeError: object of type 'NoneType' has no len()

In [138]: pd.merge(left1, right1, left_on = 'key', right_index=True)
Out[138]:
  key  value  group_val
0   a      0        3.5
2   a      2        3.5
3   a      3        3.5
1   b      1        7.0
4   b      4        7.0

In [139]: pd.merge(left1, right1, left_on = 'key', right_index = True, how = 'outer')
Out[139]:
  key  value  group_val
0   a      0        3.5
2   a      2        3.5
3   a      3        3.5
1   b      1        7.0
4   b      4        7.0
5   c      5        NaN

In [140]: pd.merge(left1, right1, left_on = 'key', right_index = True, how = 'righter')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-140-e2dba0cdfb5a> in <module>()
----> 1 pd.merge(left1, right1, left_on = 'key', right_index = True, how = 'righter')

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in merge(left, right, how, on, l
eft_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
     56                          copy=copy, indicator=indicator,
     57                          validate=validate)
---> 58     return op.get_result()
     59
     60

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in get_result(self)
    580                 self.left, self.right)
    581
--> 582         join_index, left_indexer, right_indexer = self._get_join_info()
    583
    584         ldata, rdata = self.left._data, self.right._data

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in _get_join_info(self)
    746         else:
    747             (left_indexer,
--> 748              right_indexer) = self._get_join_indexers()
    749
    750             if self.right_index:

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in _get_join_indexers(self)
    725                                       self.right_join_keys,
    726                                       sort=self.sort,
--> 727                                       how=self.how)
    728
    729     def _get_join_info(self):

~/anaconda3/lib/python3.6/site-packages/pandas/core/reshape/merge.py in _get_join_indexers(left_keys,
 right_keys, sort, how, **kwargs)
   1062     if how == 'left':
   1063         kwargs['sort'] = sort
-> 1064     join_func = _join_functions[how]
   1065
   1066     return join_func(lkey, rkey, count, **kwargs)

KeyError: 'righter'

In [141]: pd.merge(left1, right1, left_on = 'key', right_index = True, how = 'right')
Out[141]:
  key  value  group_val
0   a      0        3.5
2   a      2        3.5
3   a      3        3.5
1   b      1        7.0
4   b      4        7.0

In [142]: pd.merge(left1, right1, left_on = 'key', right_index = True, how = 'left')
Out[142]:
  key  value  group_val
0   a      0        3.5
1   b      1        7.0
2   a      2        3.5
3   a      3        3.5
4   b      4        7.0
5   c      5        NaN

In [143]: left2 = DataFrame([[1., 2.], [3., 4.], [5., 6.]], index=['a', 'c', 'e'],
     ...: columns = ['Ohio', "Nevada"])

In [144]: right2 = DataFrame([[7., 8.], [9., 10.], [11., 12.], [13, 14]],
     ...: index = ['b', 'c', 'd', 'e'], columns = ['Missouri', 'Alabama'])
```

```
In [145]: pd.merge(left2, right2, how = 'outer', left_index=True, right_index=True)
Out[145]:
   Ohio  Nevada  Missouri  Alabama
a   1.0     2.0       NaN      NaN
b   NaN     NaN       7.0      8.0
c   3.0     4.0       9.0     10.0
d   NaN     NaN      11.0     12.0
e   5.0     6.0      13.0     14.0

In [146]: left2
Out[146]:
   Ohio  Nevada
a   1.0     2.0
c   3.0     4.0
e   5.0     6.0

In [147]: right2
Out[147]:
   Missouri  Alabama
b       7.0      8.0
c       9.0     10.0
d      11.0     12.0
e      13.0     14.0

In [148]: left2.join(right2, how = 'outer')
Out[148]:
   Ohio  Nevada  Missouri  Alabama
a   1.0     2.0       NaN      NaN
b   NaN     NaN       7.0      8.0
c   3.0     4.0       9.0     10.0
d   NaN     NaN      11.0     12.0
e   5.0     6.0      13.0     14.0

In [149]: arr = np.arange(12).reshape((3,4))
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-149-ef9635e953f1> in <module>()
----> 1 arr = np.arange(12).reshape((3,4))

NameError: name 'np' is not defined

In [150]: import numpy as np

In [151]: arr = np.arange(12).reshape((3,4))

In [152]: arr
Out[152]:
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])

In [153]: np.concatenate([arr, arr], axis = 1)
Out[153]:
array([[ 0,  1,  2,  3,  0,  1,  2,  3],
       [ 4,  5,  6,  7,  4,  5,  6,  7],
       [ 8,  9, 10, 11,  8,  9, 10, 11]])

In [154]: from pandas import Series

In [155]: s1 = Series([0,1], index = ['a', 'b'])

In [156]: s2 = Series([2, 3, 4], index = ['c', 'd', 'e'])

In [157]: s3 = Series([5,6], index = ['f', 'g'])

In [158]: pd.concat([s1, s2, s3])
Out[158]:
```

```
a    0
b    1
c    2
d    3
e    4
f    5
g    6
dtype: int64

In [159]: pd.concat([s1, s2, s3], axis = 1)
Out[159]:
     0    1    2
a  0.0  NaN  NaN
b  1.0  NaN  NaN
c  NaN  2.0  NaN
d  NaN  3.0  NaN
e  NaN  4.0  NaN
f  NaN  NaN  5.0
g  NaN  NaN  6.0

In [160]: s4 = pd.concat([s1*5, s3])

In [161]: s4
Out[161]:
a    0
b    5
f    5
g    6
dtype: int64

In [162]: s1
Out[162]:
a    0
b    1
dtype: int64

In [163]: pd.concat([s1, s2, s3], axis=1, keys=['one', 'two', 'three'])
Out[163]:
   one  two  three
a  0.0  NaN    NaN
b  1.0  NaN    NaN
c  NaN  2.0    NaN
d  NaN  3.0    NaN
e  NaN  4.0    NaN
f  NaN  NaN    5.0
g  NaN  NaN    6.0

In [164]: df1 = DataFrame(np.arange(6).reshape(3,2), index = ['a', 'b', 'c'], columns = ['one', 'two'
     ...: ])

In [165]: df2 = DataFrame(5+np.arange(4).reshape(2,2), index = ['a', 'c'], columns = ['three', 'four'
     ...: ])

In [166]: df1
Out[166]:
   one  two
a    0    1
b    2    3
c    4    5

In [167]: df2
Out[167]:
   three  four
a      5     6
c      7     8

In [168]: pd.concat([df1,df2], axis = 1, keys = ['level1', 'level2'])
Out[168]:
```

```
  level1     level2
     one two  three four
a      0   1    5.0  6.0
b      2   3    NaN  NaN
c      4   5    7.0  8.0

In [169]: pd.concat([df1,df2], axis = 1)
Out[169]:
   one  two  three  four
a    0    1    5.0   6.0
b    2    3    NaN   NaN
c    4    5    7.0   8.0

In [170]: a = Series([np.nan, 2.5, np.nan, 3.5, 4.5, np.nan],
     ...: index=['f', 'e', 'd', 'c', 'b', 'a'])

In [171]: b = Series(np.arange(len(a), dtype=np.float64),index=['f', 'e', 'd', 'c', 'b', 'a'])

In [172]: a
Out[172]:
f    NaN
e    2.5
d    NaN
c    3.5
b    4.5
a    NaN
dtype: float64

In [173]: b
Out[173]:
f    0.0
e    1.0
d    2.0
c    3.0
b    4.0
a    5.0
dtype: float64

In [174]: np.where(pd.isnull(a), b, a)
Out[174]: array([0. , 2.5, 2. , 3.5, 4.5, 5. ])

In [175]: b[:-2]
Out[175]:
f    0.0
e    1.0
d    2.0
c    3.0
dtype: float64

In [176]: b[:-2].combine_first(a[2:])
Out[176]:
a    NaN
b    4.5
c    3.0
d    2.0
e    1.0
f    0.0
dtype: float64

In [177]: a[2:]
Out[177]:
d    NaN
c    3.5
b    4.5
a    NaN
dtype: float64

In [178]: df1 = DataFrame({'a': [1., np.nan, 5., np.nan], ....: 'b': [np.nan, 2., np.nan, 6.], ....:
```

```
     ...: 'c': range(2, 18, 4)})
  File "<ipython-input-178-0bfbf9c72a4a>", line 1
    df1 = DataFrame({'a': [1., np.nan, 5., np.nan], ....: 'b': [np.nan, 2., np.nan, 6.], ....: 'c': r
ange(2, 18, 4)})
                                                          ^
SyntaxError: invalid syntax


In [179]: df1 = DataFrame({'a': [1., np.nan, 5., np.nan],
     ...: 'b': [np.nan, 2., np.nan, 6.],
     ...: 'c': range(2, 18, 4)})

In [180]: df2 = DataFrame({'a': [5., 4., np.nan, 3., 7.],
     ...: 'b': [np.nan, 3., 4., 6., 8.]})

In [181]: df1.combine_first(df2)
Out[181]:
     a    b     c
0  1.0  NaN   2.0
1  4.0  2.0   6.0
2  5.0  4.0  10.0
3  3.0  6.0  14.0
4  7.0  8.0   NaN

In [182]: df1
Out[182]:
     a    b   c
0  1.0  NaN   2
1  NaN  2.0   6
2  5.0  NaN  10
3  NaN  6.0  14

In [183]: df2
Out[183]:
     a    b
0  5.0  NaN
1  4.0  3.0
2  NaN  4.0
3  3.0  6.0
4  7.0  8.0

In [184]: data = DataFrame(np.arange(6).reshape((2,3)), index = pd.Index(['Ohio', 'Colorado'], name =
     ...:  'state'), columns = pd.Index(['one', 'two', 'three'], name = 'number'))

In [185]: data
Out[185]:
number    one  two  three
state
Ohio        0    1      2
Colorado    3    4      5

In [186]: result = data.stack()

In [187]: regult
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-187-a903ac7fb30c> in <module>()
----> 1 regult

NameError: name 'regult' is not defined

In [188]: result
Out[188]:
state     number
Ohio      one       0
          two       1
          three     2
Colorado  one       3
```

```
          two       4
          three     5
dtype: int64

In [189]: result.shape
Out[189]: (6,)

In [190]: result.unstack()
Out[190]:
number    one  two  three
state
Ohio        0    1      2
Colorado    3    4      5

In [191]: result.unstack()
Out[191]:
number    one  two  three
state
Ohio        0    1      2
Colorado    3    4      5

In [192]: result.unstack(1)
Out[192]:
number    one  two  three
state
Ohio        0    1      2
Colorado    3    4      5

In [193]: result.unstack(0)
Out[193]:
state   Ohio  Colorado
number
one        0         3
two        1         4
three      2         5

In [194]: result.unstack?
Signature: result.unstack(level=-1, fill_value=None)
Docstring:
Unstack, a.k.a. pivot, Series with MultiIndex to produce DataFrame.
The level involved will automatically get sorted.

Parameters
----------
level : int, string, or list of these, default last level
    Level(s) to unstack, can pass level name
fill_value : replace NaN with this value if the unstack produces
    missing values

    .. versionadded: 0.18.0

Examples
--------
>>> s = pd.Series([1, 2, 3, 4],
...     index=pd.MultiIndex.from_product([['one', 'two'], ['a', 'b']]))
>>> s
one  a    1
     b    2
two  a    3
     b    4
dtype: int64

>>> s.unstack(level=-1)
     a  b
one  1  2
two  3  4

>>> s.unstack(level=0)
```

```
   one  two
a    1    3
b    2    4

Returns
-------
unstacked : DataFrame
File:      ~/anaconda3/lib/python3.6/site-packages/pandas/core/series.py
Type:      method

In [195]: s1 = Series([0, 1, 2, 3], index=['a', 'b', 'c', 'd'])

In [196]: s2 = Series([4, 5, 6], index=['c', 'd', 'e'])

In [197]: data2 = pd.concat([s1, s2], keys=['one', 'two'])

In [198]: data2.unstack()
Out[198]:
       a    b    c    d    e
one  0.0  1.0  2.0  3.0  NaN
two  NaN  NaN  4.0  5.0  6.0

In [199]: 是
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-199-b0d23dcafa4c> in <module>()
----> 1 是

NameError: name '是' is not defined

In [200]: s1
Out[200]:
a    0
b    1
c    2
d    3
dtype: int64

In [201]: s2
Out[201]:
c    4
d    5
e    6
dtype: int64

In [202]: data2
Out[202]:
one  a    0
     b    1
     c    2
     d    3
two  c    4
     d    5
     e    6
dtype: int64

In [203]: ldata[:10]
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-203-e7468bd30900> in <module>()
----> 1 ldata[:10]

NameError: name 'ldata' is not defined

In [204]:
```