```
In [14]:  df_unmelted = df_melted.pivot(index = ['Class', 'Name'],
                                        columns='Subject',
                                        values='Grade')
          df_unmelted
```

```
Out[14]:  Subject       Chinese   Math
          Class  Name
          1      A           80     80
          2      B           90     75
```

恢复索引并重命名列索引后，通过 equals()函数可以验证得到的 df_unmelted 结果与最初的 df 完全一致。

```
In [15]:  df_unmelted = df_unmelted.reset_index()
          df_unmelted = df_unmelted.rename_axis(columns={'Subject':''})
          df_unmelted.equals(df)
```

```
Out[15]:  True
```

注解

equals()函数的作用是检验两个 Series 或两个 DataFrame 是否一致。两个 Series 一致，当且仅当：（1）Series 的 dtype 属性一致；（2）Series 的 values 个数和值一致；（3）索引的元素值一致。两个 DataFrame 一致，当且仅当：（1）DataFrame 每列的 dtype 一致；（2）DataFrame 的 values 维度和值一致；（3）行索引和列索引的元素值一致。

在 melt()函数中，列索引上的待转换元素对应列表示的含义相同，例如 Chinese 列和 Math 列下的数都表示学生某一次考试的成绩。如果列索引上的元素分别为语文期中成绩、数学期中成绩、语文期末成绩和数学期末成绩，想要把由语文和数学构成的 Subject 转换到列或行索引上，将期中和期末的信息保留在列索引上，这种需求对于 melt()是难以实现的。

```
In [16]:  df = pd.DataFrame({'Class':[1,2],'Name':['A', 'B'],
                             'Mid_Chinese':[80, 85], 'Mid_Math':[70, 80],
                             'Final_Chinese':[90, 75], 'Final_Math':[95, 85]})
```

此时，我们可以选择 wide_to_long()函数来完成：

```
In [17]:  pd.wide_to_long(df, stubnames=['Mid', 'Final'],
                          i=['Class', 'Name'], j='Subject',
                          sep='_', suffix='.+')
```

```
Out[17]:                      Mid   Final
          Class Name Subject
          1     A    Chinese   80      90
                     Math      70      95
          2     B    Chinese   85      75
                     Math      80      85
```

wide_to_long()函数的参数较多且参数名设计不友好，其中 i 等价于 melt()中的 id_vars 或 pivot()中的 index，j 表示列索引待转换元素列表的含义，此处由于需要把学科进行转移，故取 j 为“Subject”。列索引元素现在由两部分构成并用下划线分割，sep 表示分割的字符串参数，sep 之前的元素仍然保留在列索引上，此处为期中或期末成绩，sep 之后的元素（Subject 的元素）可以用 suffix 正则参数进行捕获，“.+”表示匹配至少一个除换行符以外的字符，有关正则表达式的用法