Python的内置函数enumerate将返回一个迭代器。该迭代器每次将返回其下表index以及对应的值。而For循环只返回可迭代对象的元素。在既需要其index又需要其元素的时候的情况，例如迭代列表，既需要其index又需要其元素的时候，可以优雅的实现。

例如：打印列表values = [“a”, “b”, “c”]的下标以及元素

方法1：for循环，手动定义下标，手动更新

>>> index = 0 # 需要先定义一个下标，然后在for循环中，每次迭代手动+1

>>> for value in values:

... print(index, value)

... index += 1

...

0 a

1 b

2 c

方法2：for循环，轮询下标，每次手动获取元素

>>> for index in range(len(values)):

... value = values[index] # 手动获取元素， 与方法1不同的是，不需要手动更新index但需要手动获取对应下标的元素

... print(index, value)

...

0 a

1 b

2 c

方法3：使用enumerate函数  
>>> for count, value in enumerate(values): # enumerate(iterable)返回迭代器，每次取出下标以及对应的元素

... print(count, value)

...

0 a

1 b

2 c