Part1

capital={"Korea":"Seoul","China":"Beijing","Vietnam":"Saigon","Japan":"Tokyo"}  
def Return\_Key(dic):  
 Dkey=list()  
 for i in dic:  
 Dkey.append(dic[i])  
 return Dkey  
print(Return\_Key(capital))

>>> ['Seoul', 'Beijing', 'Saigon', 'Tokyo']

This example I use for statement traverse dictionary key to get the value then store to list Dkey, which is a state variable and make it as a return value.

Part2:

capital={"Korea":"Seoul","China":"Beijing","Vietnam":"Saigon","Japan":"Tokyo"}  
def invert\_dict(d):  
 inverse = dict()  
 for key in d:  
 val = d[key]  
 if val not in inverse:  
 inverse[val] = [key]  
 else:  
 inverse[val].append(key)  
 for key in inverse:  
 inverse[key]=inverse[key][0]  
 return inverse  
  
print(capital)  
print(invert\_dict(capital))

>>> {'Korea': 'Seoul', 'China': 'Beijing', 'Vietnam': 'Saigon', 'Japan': 'Tokyo'}

{'Seoul': 'Korea', 'Beijing': 'China', 'Saigon': 'Vietnam', 'Tokyo': 'Japan'}

The highlight is modified additional code to turn each of the list items into separate keys in the inverted dictionary.

The dictionary is useful in my example make capital city match countries. Because some special country has multiple capital, the value will be a list.

For example:

{“South Africa”:[“Cape Town”,”Bloemfontein”,”Pretoria”],”Bolivia”:[”La Paz”,”Sucre”]}

Inverted dictionary is also useful in this example, we can easily to track a country from their capital in inverted dictionary, and some of the different city will match the same country.