Inverting a Dictionary

Switching Keys and Values

Dictionaries have keys that are unique and each key has a value associated with it. For example, here is a dictionary mapping fruit to their colours:

```
fruit_to_colour = {'watermelon': 'green', 'pomegranate': 'red',
'peach': 'orange', 'cherry': 'red', 'pear': 'green',
'banana': 'yellow', 'plum': 'purple', 'orange': 'orange'}
```

To invert the dictionary, that is, switch the mapping to be colours to fruit, here is one approach:

```
>>> colour_to_fruit = {}
>>> for fruit in fruit_to_colour:
    colour = fruit_to_colour[fruit]
    colour_to_fruit[colour] = fruit


>>> colour_to_fruit
{'orange': 'orange', 'purple': 'plum', 'green': 'pear', 'yellow': 'banana', 'red': 'pomegranate'}
```

The resulting dictionary is missing some fruit. This happens since colours, which are keys, are unique so later assignments using the same colour replace earlier entries. A way to remedy this is to map colours to a list of fruit.

Mapping A Key To A List

For the example above, we need to consider two cases when adding a colour and a fruit to the dictionary:

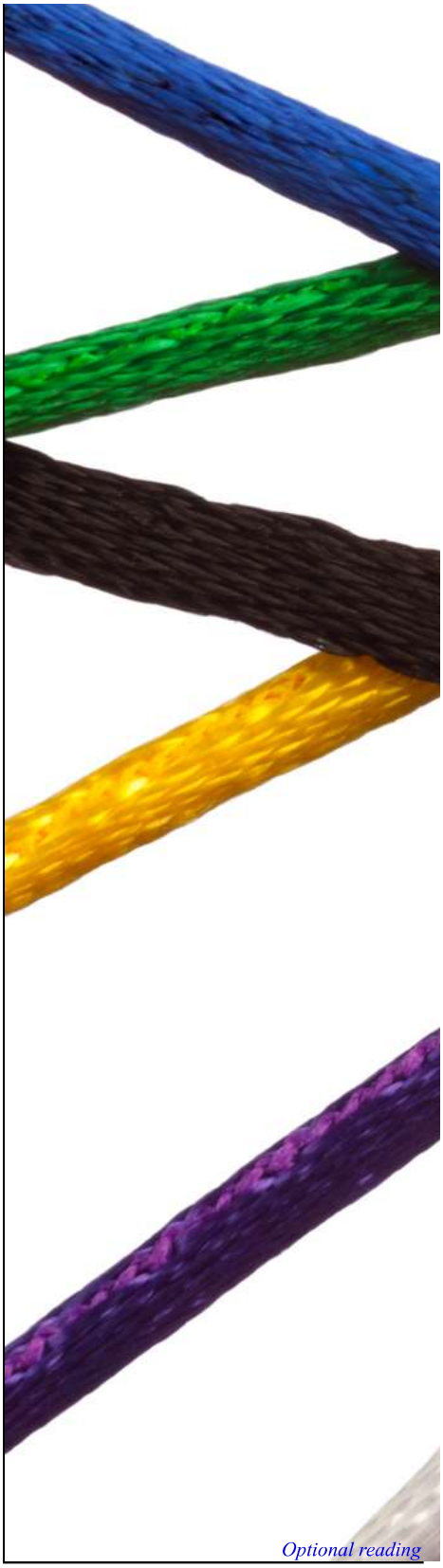
1. If the colour is not a key in the dictionary, add it with its value being a single element a list consisting of the fruit.
2. If the colour is already a key, append the fruit to the list of fruit associated with that key.

```
>>> colour_to_fruit = {}
>>> for fruit in fruit_to_colour:
    # What colour is the fruit?
    colour = fruit_to_colour[fruit]
    if not (colour in colour_to_fruit):
        colour_to_fruit[colour] = [fruit]
    else:
        colour_to_fruit[colour].append(fruit)


>>> colour_to_fruit
{'orange': ['peach', 'orange'], 'purple': ['plum'], 'green': ['watermelon', 'pear'], 'yellow': ['banana'], 'red': ['cherry', 'pomegranate']}
```


Jennifer Campbell • Paul Gries
University of Toronto


Pr
Sec


Optional reading