

other dictionaries!

3.The dict.update method updates one dictionary with the items coming from the other dictionary, so that existing entries are replaced and new entries are added. What is the content of the dictionary "wardrobe" at the end of the following code?

1 / 1 point

```
1 wardrobe = {'shirt': ['red', 'blue', 'white'], 'jeans': ['blue', 'black']}
2 new_items = {'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
3 wardrobe.update(new_items)
```

- ☐ {'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
- ☒ {'shirt': ['red', 'blue', 'white'], 'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
- ☐ {'shirt': ['red', 'blue', 'white'], 'jeans': ['blue', 'black', 'white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
- ☐ {'shirt': ['red', 'blue', 'white'], 'jeans': ['blue', 'black'], 'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}

✓ Correct

Correct! The dict.update method updates the dictionary (wardrobe) with the items coming from the other dictionary (new_items), adding new entries and replacing existing entries.

4. What's a major advantage of using dictionaries over lists?

1 / 1 point

- ☐ Dictionaries are ordered sets
- ☐ Dictionaries can be accessed by the index number of the element
- ☐ Elements can be removed and inserted into dictionaries
- ☒ It's quicker and easier to find a specific element in a dictionary

✓ Correct

Right on! Because of their unordered nature and use of key value pairs, searching a dictionary takes the same

✓ Correct

Well done, you! You're now creating dictionaries out of other dictionaries!

3.The dict.update method updates one dictionary with the items coming from the other dictionary, so that existing entries are replaced and new entries are added. What is the content of the dictionary "wardrobe" at the end of the following code?

1 / 1 point

```
1 wardrobe = {'shirt': ['red', 'blue', 'white'], 'jeans': ['blue', 'black']}
2 new_items = {'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
3 wardrobe.update(new_items)
```

- ☐ {'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
- ☒ {'shirt': ['red', 'blue', 'white'], 'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
- ☐ {'shirt': ['red', 'blue', 'white'], 'jeans': ['blue', 'black', 'white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}
- ☐ {'shirt': ['red', 'blue', 'white'], 'jeans': ['blue', 'black'], 'jeans': ['white'], 'scarf': ['yellow'], 'socks': ['black', 'brown']}

✓ Correct

Correct! The dict.update method updates the dictionary (wardrobe) with the items coming from the other dictionary (new_items), adding new entries and replacing existing entries.

4. What's a major advantage of using dictionaries over lists?

1 / 1 point

- ☐ Dictionaries are ordered sets
- ☐ Dictionaries can be accessed by the index number of the element

4. What's a major advantage of using dictionaries over lists?

1 / 1 point

- ☐ Dictionaries are ordered sets
- ☐ Dictionaries can be accessed by the index number of the element
- ☐ Elements can be removed and inserted into dictionaries
- ☒ It's quicker and easier to find a specific element in a dictionary

✓ Correct

Right on! Because of their unordered nature and use of key value pairs, searching a dictionary takes the same amount of time no matter how many elements it contains

5. The add_prices function returns the total price of all of the groceries in the dictionary. Fill in the blanks to complete this function.

1 / 1 point

```
1 def add_prices(basket):
2     # Initialize the variable that will be used for the calculation
3     total = 0
4     # Iterate through the dictionary items
5     for key, value in basket.items():
6         # Add each price to the total calculation
7         # Hint: how do you access the values of
8         # dictionary items?
9         total += value
10    # Limit the return value to 2 decimal places
11    return round(total, 2)
12
13 groceries = {"bananas": 1.56, "apples": 2.50, "oranges": 0.99, "bread": 4.59,
14             "coffee": 6.99, "milk": 3.39, "eggs": 2.98, "cheese": 5.44}
15
16 print(add_prices(groceries)) # Should print 28.44
17
```

Run

Reset

✓ Correct

Nicely done! Dictionaries are a helpful way to store information, and access it easily when it's needed.