MODIFYING DICTIONARIES

Referencing a *new* key and assigning it a value will **add a new key-value pair**, while referencing an *existing* key will **overwrite the existing pair**

```
item_details = {'skis': [249.99, 10, 'in stock'],
                'snowboard': [219.99, 0, 'sold out'],
                'goggles': [99.99, 0, 'sold out'],
                'boots': [79.99, 7, 'in stock']}
```

Notice that 'bindings' is not in the dictionary keys

```
item_details['bindings'] = [149.99, 4, 'in stock']
item_details
```

```
{'skis': [249.99, 10, 'in stock'],
 'snowboard': [219.99, 0, 'sold out'],
 'goggles': [99.99, 0, 'sold out'],
 'boots': [79.99, 7, 'in stock'],
 'bindings': [149.99, 4, 'in stock']}
```

Therefore, assigning a value to a key of 'bindings' adds the key-value pair to the dictionary

```
item_details['bindings'] = [139.99, 0, 'out of stock']

item_details
```

```
{'skis': [249.99, 10, 'in stock'],
 'snowboard': [219.99, 0, 'sold out'],
 'goggles': [99.99, 0, 'sold out'],
 'boots': [79.99, 7, 'in stock'],
 'bindings': [139.99, 0, 'out of stock']}
```

Now that 'bindings' is a key in the dictionary, assigning a value to a key of 'bindings' overwrites the value in the dictionary for that key

MODIFYING DICTIONARIES

Use the **del** keyword to delete key-value pairs

```
item_details
```

```
{'skis': [249.99, 10, 'in stock'],
 'snowboard': [219.99, 0, 'sold out']
 'goggles': [99.99, 0, 'sold out'],
 'boots': [79.99, 7, 'in stock'],
 'bindings': [149.99, 4, 'in stock']}
```

```
del item_details['boots']

item_details
```

```
{'skis': [249.99, 10, 'in stock'],
 'snowboard': [219.99, 0, 'sold out'],
 'goggles': [99.99, 0, 'sold out'],
 'bindings': [139.99, 0, 'out of stock']}
```

This has deleted the key-value pair with a key of 'boots' from the dictionary