Solutions

1. In Python, an empty dictionary can be created using curly braces {} or by using the built-in dict() function without any arguments. Here's an example of both methods:

```
my_dict = {}
```

2. If a dictionary has a key 'foo' with the value 42, then the value of the dictionary with the key 'foo' is 42.

3. A list is an ordered collection of elements, where each element is identified by its position or index in the list. Lists can contain elements of any type, including other lists or dictionaries. Elements in a list can be added, removed, or modified using their index, and lists are mutable, meaning that they can be changed after they are created.

   A dictionary, on the other hand, is an unordered collection of key-value pairs. Each element in a dictionary consists of a key and its associated value, which can be of any type. Keys in a dictionary must be unique and immutable (e.g., strings, numbers, or tuples), while values can be mutable or immutable. Unlike lists, elements in a dictionary are not accessed by their position, but by their keys. Dictionaries are also mutable, meaning that their values can be changed after they are created.

4. If you try to access spam['foo'] and spam is { 'bar': 100 }, you will get a KeyError.

5. The main difference between cat in spam and cat in spam.keys() is the data type of the right-hand operand. The first expression checks if the key exists in the dictionary, while the second expression first gets a list of keys from the dictionary and then checks if the key exists in the list.

6. The main difference between cat in spam and cat in spam.values() is the data type of the right-hand operand. The first expression checks if the

value exists as a key in the dictionary, while the second expression checks if the value exists in the values of the dictionary.

7. spam.setdefault('color', 'black')

8. In Python, you can "pretty print" dictionary values using the pprint module and the pprint() function.

   The pprint module provides a way to pretty-print arbitrary Python data structures in a form that can be used as input to the Python interpreter. The pprint() function in this module prints the data structure in a readable format by sorting the keys, wrapping long lines, and indenting the output

   Example:

```
import pprint

# Define a dictionary
my_dict = {'name': 'John', 'age': 30, 'city': 'New York'}

# Use pprint to pretty-print the dictionary
pprint.pprint(my_dict)
```