Assignment No-5

1. What does an empty dictionary's code look like?

Solution: An empty dictionary in Python looks like this:

{}

2. What is the value of a dictionary value with the key 'foo' and the value 42?

Solution: The value of a dictionary with the key 'foo' and the value 42 is 42.

3. What is the most significant distinction between a dictionary and a list?

Solution: The most significant distinction between a dictionary and a list is the way they store and access data.

A dictionary is an unordered collection of key-value pairs. It uses a key to uniquely identify each value in the collection. In other words, it maps keys to values. The keys in a dictionary are unique, and each key is associated with a value. Dictionaries provide fast access to values based on their keys and are typically used when you need to lookup values by their keys.

On the other hand, a list is an ordered collection of elements. It does not use keys to associate values; instead, it uses positions or indices to store and retrieve elements. Lists are linear data structures, meaning the elements are stored in a sequence and are accessible by their index values. Lists are commonly used when you need to work with a collection of items in a specific order.

The critical distinction is that a dictionary uses keys to access values, while a list uses indices. Dictionaries are suitable for lookup operations based on unique keys, whereas lists are helpful when dealing with elements in a specific order.

4. What happens if you try to access spam['foo'] if spam is {'bar': 100}?

Solution: If you try to access “spam['foo']” when “spam” is “{'bar': 100}”, you will get a “KeyError” because “'foo'” is not a key in the dictionary “spam”.

5. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.keys()?

Solution: In Python, if a dictionary is stored in a variable called `spam`, the expression “'cat' in spam” checks if the key `'cat'` exists in the dictionary “spam” and returns `True` or `False` accordingly.

On the other hand, the expression `'cat' in spam.keys()` checks if the string `'cat'` exists as a key in the dictionary `spam` and returns `True` or `False` accordingly.

In essence, both expressions are similar and check if the key `'cat'` exists in the dictionary `spam`. However, the second expression explicitly uses the `.keys()` method to obtain a list of all the keys in the dictionary `spam` before checking the presence of the key `'cat'`.

6. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.values()?

Solution: The expression 'cat' in spam checks if the key 'cat' exists in the dictionary spam. If the key 'cat' exists, it will return True.

On the other hand, the expression 'cat' in spam.values() checks if the value 'cat' exists in the dictionary spam. If the value 'cat' exists, it will return True.

The difference lies in which part of the dictionary is being checked: 'cat' in spam checks if 'cat' is a key, while 'cat' in spam.values() checks if 'cat' is a value in the dictionary.

7. What is a shortcut for the following code?

if 'color' not in spam:

spam['color'] = 'black'

Solution: spam.setdefault('color', 'black')

8. How do you "pretty print" dictionary values using which module and function?

Solution: We can implement "pretty print" dictionary values using the `pprint` module and its `pprint()` function. Here's an example:

import pprint

my_dict = {"key1": "value1", "key2": "value2", "key3": "value3"}

pprint.pprint(my_dict)

The `pprint()` function in the `pprint` module formats the dictionary values in a cleaner and more readable way by adding line breaks and indentation.