**1.What does an empty dictionary code look like?**

An empty dictionary in Python is denoted using {} or the dict() constructor:

empty\_dict = {}

# or

empty\_dict = dict()

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

The value of a dictionary with key 'foo' and value 42 in Python would be as follows:

my\_dict = {'foo': 42}

You can access the value of the key 'foo' in the dictionary my\_dict using square brackets []:

value = my\_dict['foo']

print(value)

Output:

42

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

The most significant distinction between a dictionary and a list in Python is their usage and structure.

A list is an ordered collection of items, where each item is identified by its index in the list, starting from 0. Lists can contain items of any type, including other lists.

my\_list = [1, 2, 3, 'apple', 'banana']

A dictionary, on the other hand, is an unordered collection of key-value pairs, where each key is used to access its associated value. Dictionaries can contain items of any type, including other dictionaries.

my\_dict = {'a': 1, 'b': 2, 'c': 'apple'}

So, the main difference between dictionaries and lists is that lists use indexes to access items, while dictionaries use keys to access values. This makes dictionaries useful when you need to access items by some sort of key, rather than by their position in a collection.

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

If you try to access spam['foo'] where spam = {'bar': 100}, a KeyError will be raised in Python, because the key 'foo' is not present 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()?**

The expression 'cat' in spam checks if the string 'cat' is one of the keys in the dictionary spam. If 'cat' is a key in the dictionary, the expression evaluates to True, otherwise it evaluates to False.

The expression 'cat' in spam.keys() does the same thing, but the .keys() method returns a view of all the keys in the dictionary, so you are checking if 'cat' is in the list of keys, rather than in the dictionary itself.

So, the difference between the two expressions is that the first one checks if 'cat' is a key in the dictionary directly, while the second one checks if 'cat' is a key in the list of keys returned by the .keys() method. The result of both expressions will be the same, but the second expression is less efficient, as it requires creating a view of all the keys in the dictionary before checking for membership.

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

The expression 'cat' in spam checks if the string 'cat' is one of the keys in the dictionary spam. If 'cat' is a key in the dictionary, the expression evaluates to True, otherwise it evaluates to False.

The expression 'cat' in spam.values() checks if the string 'cat' is one of the values in the dictionary spam. If 'cat' is a value in the dictionary, the expression evaluates to True, otherwise it evaluates to False.

So, the difference between the two expressions is that the first one checks if 'cat' is a key in the dictionary, while the second one checks if 'cat' is a value in the dictionary. The result of both expressions will be different based on the content of the dictionary.

**7. What is a shortcut for the following code?**

**if 'color' not in spam:**

**spam['color'] = 'black'**

The following code is a shorthand equivalent of the code you provided:

spam['color'] = spam.get('color', 'black')

The .get() method retrieves the value of the specified key in the dictionary. If the key is not found, it returns the default value specified as the second argument (in this case, 'black').

The code spam['color'] = spam.get('color', 'black') first calls spam.get('color', 'black') to retrieve the value of the key 'color' in the dictionary. If the key 'color' is present, its value is returned. If the key is not present, the default value 'black' is returned. Then, the returned value is assigned to the key 'color' in the dictionary. This way, the code sets the value of the key 'color' to 'black' only if it's not already present in the dictionary.

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

You can use the pprint function from the pprint module in Python to "pretty print" dictionary values. The pprint function takes a Python object as an argument and returns a string that is a nicely formatted representation of the object. For dictionaries, it will print the keys and values in an easy-to-read format, with each key-value pair on a separate line and proper indentation.