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

ANS - In Python, you can create an empty dictionary using the following methods:

Use of { } symbol. We can create an empty dictionary object by giving no elements in curly brackets in the assignment statement. Here’s an example:

emptyDict = {}

print(emptyDict)

Output:

{}

Use of dict() built-in function. Empty dictionary is also created by dict() built-in function without any arguments. Here’s an example:

emptyDict = dict()

print(emptyDict)

Output:

{}

Initialize a dictionary using a dictionary comprehension. A dictionary comprehension is a concise way to create a dictionary in Python, using a single line of code. In this case, the comprehension is empty because there are no key-value pairs to iterate over. Therefore, an empty list is passed to comprehension. Here’s an example:

emptyDict = {key: value for key, value in []}

print(emptyDict)

Output:

{}

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

ANS - In Python, you can access the value of a dictionary value with the key ‘foo’ and the value 42 using the following code:

myDict = {1:"egg", "Answer":42, 8:14, "foo":42}

print(myDict['foo'])

Output:

42

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

ANS- The most significant distinction between a dictionary and a list in Python is that a list is an ordered collection of elements whereas a dictionary is an unordered collection of key-value pairs.

Lists are linear in nature and can contain duplicate values. They are mutable, meaning that they can be changed after creation. Lists are created by placing elements in [ ] separated by commas.

Dictionaries are hashed structures of key-value pairs. Each key-value pair in a dictionary is separated by a colon (:) whereas each key is separated by a comma. The keys of the dictionary can be of any data type. Dictionaries are unordered and mutable but they cannot contain duplicate keys.

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

ANS- If you try to access spam['foo'] when spam is {'bar': 100} in Python, you will get a KeyError. This is because the key 'foo' does not exist 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()?

In Python, there is no difference between the expressions 'cat' in spam and 'cat' in spam.keys(). The in operator checks whether a value exists as a key in the dictionary. 'cat' in spam checks whether there is a 'cat' key in the dictionary, while 'cat' in spam.values() checks whether there is a value 'cat' for one of the keys in spam.

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

ANS - In Python, there is no difference between the expressions 'cat' in spam and 'cat' in spam.keys(). The in operator checks whether a value exists as a key in the dictionary. 'cat' in spam checks whether there is a 'cat' key in the dictionary, while 'cat' in spam.values() checks whether there is a value 'cat' for one of the keys in spam.

7. What is a shortcut for the following code?

if 'color' not in spam:

spam['color'] = 'black'

ANS - The shortcut for the following code:

if 'color' not in spam:

spam['color'] = 'black'

is: -

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

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

ANS - In Python, you can “pretty print” dictionary values using the pprint module and the pprint() function. The pprint() function prints data structures like dictionaries in a more human-readable format that can also be used as input to the interpreter.

Here’s an example:

import pprint

myDict = {'name': 'John', 'age': 25, 'gender': 'male'}

pprint.pprint(myDict)

Output:

{'age': 25, 'gender': 'male', 'name': 'John'}