1. What does an empty dictionary&#39;s code look like?

Ans an empty dictionary can be defined using curly braces {} or by using the dict() constructor without any arguments. Here's how it looks:

empty\_dict = {}

empty\_dict = dict()

Both of these approaches create an empty dictionary with no key-value pairs.

1. What is the value of a dictionary value with the key &#39;foo&#39; and the value 42?

Ans the value of a dictionary is accessed using the corresponding key. If a dictionary has a key 'foo' with a value of 42, you can access that value by referencing the key. Here's an example:

my\_dict = {'foo': 42}

value = my\_dict['foo']

print(value)

In this case, value will be assigned the value 42, which is the value associated with the key 'foo' in the dictionary my\_dict.

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

AnsThe most significant distinction between a dictionary and the way they store and access data:

Storage of Data:

Dictionary: A dictionary stores data in key-value pairs, where each value is associated with a unique key. The keys in a dictionary are used to access and retrieve the corresponding values. Dictionary keys must be immutable objects (e.g., strings, numbers, tuples) since they are used as identifiers.

List: A list stores data in an ordered sequence. It is an ordered collection of objects where each object has an index value. Lists are mutable, which means you can modify, add, or remove elements.

Accessing Data:

Dictionary: Data in a dictionary is accessed by providing the associated key. You can use the key as an index to retrieve the corresponding value. Dictionary lookup is fast and efficient for large collections of data since it uses hash-based indexing.

List: Data in a list is accessed by using an index position. Elements in a list are ordered and indexed starting from zero. You can access elements by their index values, allowing for positional retrieval.

1. **What happens if you try to access spam[&#39;foo&#39;] if spam is {&#39;bar&#39;: 100}?**

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

A KeyError is raised when you try to access a key that does not exist in a dictionary. In this case, the key 'foo' does not exist in the spam dictionary. Here's an example:

spam = {'bar': 100}

value = spam['foo'] # Raises KeyError: 'foo'

Python will raise a KeyError with a message indicating that the key 'foo' does not exist in the dictionary. It's important to ensure that the key you are accessing actually exists in the dictionary to avoid this error.

**5. If a dictionary is stored in spam, what is the difference between the expressions &#39;cat&#39; in spam and**

**&#39;cat&#39; in spam.keys()?**

Ans there is no functional difference between the expressions 'cat' in spam and 'cat' in spam.keys() when spam is a dictionary. Both expressions check for the presence of the key 'cat' in the dictionary spam.

The in operator is used to test membership, and it can be used directly on the dictionary spam to check if a key exists. It will return True if the key is present and False otherwise.

Here's an example that demonstrates the equivalence of the two expressions:

spam = {'cat': 42, 'dog': 21, 'bird': 7}

print('cat' in spam)

print('cat' in spam.keys())

Both print statements will output True because the key 'cat' exists in the spam dictionary. Therefore, whether you use 'cat' in spam or 'cat' in spam.keys(), you will obtain the same result.

**6. If a dictionary is stored in spam, what is the difference between the expressions &#39;cat&#39; in spam and**

**&#39;cat&#39; in spam.values()?**

Ans the expressions 'cat' in spam and 'cat' in spam.values() have different meanings and outcomes when spam is a dictionary.

'cat' in spam: This expression checks whether the string 'cat' exists as a key in the dictionary spam. It returns True if the key 'cat' is present in the dictionary, and False otherwise.

'cat' in spam.values(): This expression checks whether the string 'cat' exists as a value in the dictionary spam. It returns True if the value 'cat' is found within any of the dictionary's values, and False otherwise.

Here's an example to illustrate the difference:

spam = {'animal1': 'cat', 'animal2': 'dog', 'animal3': 'bird'}

print('cat' in spam) # Output: False

print('cat' in spam.values())

In this example, 'cat' in spam returns False because 'cat' is not a key in the spam dictionary. However, 'cat' in spam.values() returns True because 'cat' is one of the values present in the dictionary.

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

**if &#39;color&#39; not in spam:**

**spam[&#39;color&#39;] = &#39;black&#39;**

Ans use the dict.setdefault() method as a shortcut to achieve the same result as the provided code. The setdefault() method allows you to set a default value for a key in a dictionary only if the key does not already exist. Here's the equivalent shortcut code:

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

This code checks if the key 'color' exists in the spam dictionary. If the key is not present, it sets the key-value pair 'color': 'black' in the dictionary. If the key already exists, it does nothing and keeps the existing value associated with that key.

Using setdefault() eliminates the need for an explicit if statement and reduces the code length while achieving the same functionality.

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

AnsTo "pretty print" dictionary values, you can use the pprint module and its pprint() function. The pprint module provides a way to format and display data structures in a more readable and organized manner. Here's an example of how to use it:

import pprint

my\_dict = {'name': 'John', 'age': 25, 'city': 'New York'}

pprint.pprint(my\_dict)

The pprint.pprint() function takes the dictionary as input and prints it in a formatted way, with each key-value pair on a separate line and indentation for better readability. It also handles nested structures and automatically adjusts the layout to improve readability.