**Answer 1:-**

The contents of a dict can be written as a series of key:value pairs within braces { }

The "empty dict" is just an empty pair of curly braces {}.

**Answer 2:-**

if a dictionary has a key 'foo' with the value 42, you can access this value using the key. Here's an example:

my\_dict = {'foo': 42}

value = my\_dict['foo']

print(value)

The output will be:

42

So, the value of the key 'foo' in this case is 42.

**Answer 3:-**

The most significant difference between a list and dictionary:-

1. The list is an ordered collection of elements, while dictionary is unordered.
2. Elements in a list are accessed using indices, while elements in a dictionary are accessed using keys.
3. A list allows duplicate items, while a dictionary does not allow duplicate keys.
4. A list can store any data type, while the keys in a dictionary can be of any immutable data type, and values can be of any data type.
5. Lists perform faster for ordered operations like sorting, while dictionaries perform faster for lookup operations.

**Answer 4:-**

If you try to access spam['foo'] when spam is defined as ['bar': 100], you'll encounter a SyntaxError before even running the code.

**Accessing** spam['foo']

* If spam were a dictionary (e.g., {'bar': 100}), attempting to access spam['foo'] would raise a KeyError because the key 'foo' does not exist in the dictionary.
* If spam were a list (e.g., ['bar', 100]), attempting to access it with a string index (e.g., spam['foo']) would raise a TypeError, as list indices must be integers or slices.

**Answer 5:-**

both 'cat' in spam and 'cat' in spam.keys() check if the key 'cat' is present in the dictionary spam.

1. **'cat' in spam:**

* This checks directly if 'cat' is a key in the dictionary spam.
* It is a more concise and common way to check for the existence of a key in a dictionary.
* It performs efficiently because it directly looks up the key in the dictionary without creating an intermediate object.

1. **'cat' in spam.keys():**

* This checks if 'cat' is in the keys view of the dictionary spam.
* While it also verifies the presence of 'cat' as a key, it creates a dict\_keys object (a view of the dictionary's keys) and then performs the lookup.
* It is less efficient than 'cat' in spam because of the overhead of creating the keys view.

**Answer 6:-**

1. **'cat' in spam:**

* This checks if 'cat' is a key in the dictionary spam.
* It does not consider the values in the dictionary, only the keys.

1. **'cat' in spam.values():**

* This checks if 'cat' is a value in the dictionary spam.
* It ignores the keys and only looks at the values in the dictionary.

**Answer 7:-**

The shortcut for this code in Python is to use the setdefault method of dictionaries:

Spam.setdefault(‘colour', ‘black')

**Explanation**:

* The setdefault method checks if the key 'colour' is in the dictionary:
* If the key exists, it does nothing and returns the existing value.
* If the key does not exist, it adds the key with the value 'black'.

This one-liner is functionally equivalent to the longer if statement.

**Answer 8:-**

To "pretty print" dictionary values in Python, you can use the pprint module and its pprint function. This function formats dictionaries (and other nested data structures) to make them more readable**.**

import pprint

# Example dictionary

data = {

"name": "Alice",

"age": 25,

"skills": ["Python", "Machine Learning", "Data Analysis"],

"address": {

"city": "New York",

"zipcode": "10001"

}

}

Output:

{'address': {'city': 'New York', 'zipcode': '10001'},

'age': 25,

'name': 'Alice',

'skills': ['Python', 'Machine Learning', 'Data Analysis']}