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

In Python, an empty dictionary is represented by a pair of curly braces with no elements inside. Here's an example of code that creates an empty dictionary:

empty\_dict = {}

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

=> 42

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

The most significant distinction between a dictionary and a list is their underlying data structure and how they store and access data.

1. Structure:

- Dictionary: A dictionary is an unordered collection of key-value pairs. Each key within a dictionary is unique, and it is used to access its corresponding value. Keys in a dictionary are typically strings or immutable objects.

- List: A list is an ordered collection of elements. The elements in a list are stored in a specific order and accessed by their position, or index, starting from zero.

2. Accessing Elements:

- Dictionary: Elements in a dictionary are accessed by their keys. You can retrieve the value associated with a specific key in constant time, regardless of the size of the dictionary.

- List: Elements in a list are accessed by their index. The index allows you to retrieve elements based on their position in the list. Accessing elements by index takes linear time, so the larger the list, the longer it takes to access elements from the end.

3. Mutability:

- Dictionary: Dictionaries are mutable, which means you can modify, add, or remove key-value pairs once the dictionary is created.

- List: Lists are also mutable, allowing you to modify, add, or remove elements from the list.

4. Order:

- Dictionary: Dictionaries do not preserve the order of elements. The key-value pairs are stored in an arbitrary order.

- List: Lists maintain the order of elements. The position of each element in the list is fixed unless explicitly modified.

5. Use Cases:

- Dictionary: Dictionaries are suitable for storing and retrieving data when you have a unique key associated with each value, such as mapping names to ages or words to their definitions.

- List: Lists are useful for storing collections of items where the order matters, such as a sequence of tasks or a set of numerical values.

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

If you try to access spam['foo'] and spam is {'bar': 100}, you will encounter a KeyError 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()?

The expressions 'cat' in spam and 'cat' in spam.keys() both check for the presence of the key 'cat' in the dictionary spam, but there is a slight difference in their behavior:

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

'cat' in spam.keys(): This expression checks if the key 'cat' exists among the keys of the dictionary spam. The spam.keys() returns a view object containing all the keys of the dictionary. It returns True if the key is found among the keys and False otherwise.

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

The expressions 'cat' in spam and 'cat' in spam.values() have different meanings and behavior:

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

'cat' in spam.values(): This expression checks if the value 'cat' exists among the values of the dictionary spam. It returns True if the value is found among the values and False otherwise.

7. What is a shortcut for the following code?

if 'color' not in spam:

spam['color'] = 'black'

A shortcut for the given code is to use the `dict.setdefault()` method. This method provides a concise way to set a default value for a key in a dictionary if the key is not already present. Here's how you can use it to achieve the same result:

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

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

To "pretty print" dictionary values in a more human-readable format, you can make use of the `pprint` module and its `pprint()` function in Python. The `pprint` module provides a convenient way to format and print complex data structures, such as dictionaries, with indentation and line breaks.

Here's an example of how you can use `pprint` to pretty print a dictionary:

import pprint

my\_dict = {'name': 'John', 'age': 30, 'occupation': 'Engineer'}

pprint.pprint(my\_dict)

Output:

{'age': 30,

'name': 'John',

'occupation': 'Engineer'}

In this code, the `pprint.pprint()` function is called with the dictionary `my\_dict` as the argument. It formats the dictionary with indentation and sorts the keys alphabetically. The result is a more visually appealing representation of the dictionary.

Using `pprint` is especially useful when dealing with larger or nested dictionaries, as it helps improve readability and comprehension of the structure and content of the dictionary.