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

empty\_dict = {}

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

The value of a dictionary value with the key 'foo' and the value 42 is simply 42. In a dictionary, values are associated with keys, allowing you to retrieve the corresponding value by specifying the key. Here's an example of how to access the value:

my\_dict = {'foo': 42}

value = my\_dict['foo']

print(value) # Output: 42

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

The most significant distinction between a dictionary and a list is the way they store and access data:

1. Structure:
   * List: A list is an ordered collection of items enclosed in square brackets ([]). It maintains the order of the elements and allows duplicate values. The elements in a list are accessed by their index, starting from 0.
   * Dictionary: A dictionary is an unordered collection of key-value pairs enclosed in curly braces ({}). Each element in a dictionary consists of a key and its associated value. The keys are unique within a dictionary and used to access the corresponding values. The order of elements in a dictionary is not guaranteed.
2. Accessing Elements:
   * List: Elements in a list are accessed by their index. You can retrieve an element from a list using square brackets and specifying the index position. For example, my\_list[0] retrieves the first element of the list.
   * Dictionary: Elements in a dictionary are accessed using keys. You can retrieve a value from a dictionary by specifying the corresponding key within square brackets. For example, my\_dict['key'] retrieves the value associated with the key 'key'.
3. What happens if you try to access spam['foo'] if spam is {'bar': 100}?

If we try to access spam['foo'] where spam is {'bar': 100}, it will raise a KeyError because the key 'foo' does not exist in the dictionary spam.

In Python, when you try to access a dictionary using a key that does not exist in the dictionary, a KeyError exception is raised. The KeyError indicates that the key you are trying to access is not found in the dictionary.

1. 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() differ in the way they check for the presence of a key in the dictionary spam.

1. 'cat' in spam:
   * This expression checks if the key 'cat' exists directly in the dictionary spam.
   * It returns True if the key is present in spam and False otherwise.
   * It does not involve any additional method calls.
2. 'cat' in spam.keys():
   * This expression checks if the key 'cat' exists in the keys of the dictionary spam.
   * spam.keys() returns a view object containing the keys of spam.
   * It then checks if 'cat' is present in that view object.
   * It also returns True if the key is present in spam and False otherwise.
3. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.values()?

In terms of functionality, both expressions can be used to determine if a specific value exists in the dictionary. However, there are some differences to consider:

* Performance: Similar to the previous case, 'cat' in spam is generally more efficient because it directly checks the dictionary for the presence of the value without creating an additional view object (spam.values()). Accessing the values directly is typically faster than creating a view object and then checking for membership.
* Use of spam.values(): Using spam.values() can be useful if you specifically need a separate iterable of the dictionary values, but if your main goal is to check for the presence of a value, using 'cat' in spam is simpler and more efficient.

In most cases, 'cat' in spam would be the preferred and more straightforward way to check for the presence of a value in a dictionary.

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. It allows you to set a default value for a key if the key is not already present in the dictionary. Here's how you can use setdefault() as a shortcut:

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

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

To "pretty print" dictionary values in a formatted and readable manner, you can use the pprint module and its pprint() function in Python. The pprint module provides a convenient way to display complex data structures, such as dictionaries, in an organized and visually appealing format.

import pprint

my\_dict = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}

pprint.pprint(my\_dict)