**Name: Abhas**

**Email:** [**abhas2495@gmail.com**](mailto:abhas2495@gmail.com)

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

**Answers:**

In Python, an empty dictionary is represented by a pair of curly braces ({}). Here's an example of how you can create an empty dictionary in Python:

my\_dict = {}

**For example:**

my\_dict = {}

my\_dict['name'] = 'John'

my\_dict['age'] = 25

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

**Answers:**

If you have a dictionary with the key 'foo' and the value 42, the value associated with the key 'foo' is simply 42. In Python, you can access the value of a dictionary using the key within square brackets.

**Here's an example:**

my\_dict = {'foo': 42}

value = my\_dict['foo']

print(value)

**Output: 42**

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

**Answers:**

**Structure:**

List: A list is an ordered collection of elements that can be of any data type. It maintains the order of elements based on their position in the list.

Dictionary: A dictionary is an unordered collection of key-value pairs. It stores elements as key-value pairs, where each key is unique and associated with a value.

**Accessing Elements:**

List: Elements in a list are accessed by their index, which represents their position in the list. The index starts from 0, and you can use square brackets ([]) with the index to retrieve an element.

Dictionary: Elements in a dictionary are accessed by their keys. Instead of using an index, you use the specific key associated with the value. You can retrieve the value using square brackets ([]) with the key.

**Mutability:**

List: Lists are mutable, meaning you can modify individual elements. You can change, add, or remove elements from a list.

Dictionary: Dictionaries are also mutable. You can modify the values associated with existing keys, add new key-value pairs, or remove existing key-value pairs.

**Order:**

List: Lists preserve the order of elements based on their position in the list.

Dictionary: Dictionaries do not guarantee any specific order for the key-value pairs. The order of elements may not necessarily match the order of insertion.

**Key-Value Structure:**

List: Lists do not have a key-value structure. Elements are accessed solely based on their position/index in the list.

Dictionary: Dictionaries have a key-value structure, where elements are accessed and identified by unique keys.

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

**Code:**

spam = {'bar': 100}

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

**Output:**

**KeyError: 'foo'**

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

**Answers:**

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

This expression checks if the key 'cat' exists directly in the dictionary spam. It returns a Boolean value (True or False) indicating whether the key is present as a direct key in the dictionary.

**Code:**

spam = {'cat': 42, 'dog': 56}

print('cat' in spam) **# Output: True**

print('elephant' in spam) **# Output: False**

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

This expression checks if the key 'cat' exists in the list of keys of the dictionary spam. It returns a Boolean value (True or False) indicating whether the key is present among the dictionary's keys.

**Code:**

spam = {'cat': 42, 'dog': 56}

print('cat' in spam.keys()) **# Output: True**

print('elephant' in spam.keys()) **# Output: False**

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

**Answers:**

**'cat' in spam:**

This expression checks if the key 'cat' exists directly in the dictionary spam. It returns a Boolean value (True or False) indicating whether the key is present as a direct key in the dictionary.

**Code:**

spam = {'cat': 42, 'dog': 56}

print('cat' in spam) **# Output: True**

print('elephant' in spam) **# Output: False**

'cat' in spam.values():

This expression checks if the value 'cat' exists in the dictionary spam as one of the values. It searches through the values of the dictionary and returns a Boolean value (True or False) indicating whether the value is present in any of the dictionary's values.

**Code:**

spam = {'name': 'cat', 'age': 3}

print('cat' in spam.values()) **# Output: True**

print('dog' in spam.values()) **# Output: False**

7. What is a shortcut for the following code?

if 'color' not in spam:

spam['color'] = 'black'

**Answers:**

A shortcut for the code you provided, which adds the key 'color' with the value 'black' to the dictionary spam if the key is not already present, is to use the **dict.setdefault()** method.

Here's the equivalent code using dict.setdefault():

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

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

**Answers:**

import pprint

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

pprint.pprint(my\_dict)

**Output:**

**{'age': 25, 'city': 'New York', 'name': 'John'}**