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

Ans :

empty\_dict = {}

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

Ans:

If you have a dictionary with the key 'foo' and the value 42, accessing the value associated with the key 'foo' would simply give you 42. Here's how it looks:

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?

Ans:

The most significant distinction between a dictionary and a list in Python is how they store and retrieve data:

Organization:

Lists: Lists are ordered collections of items, where each item has a specific position or index. The order of items in a list is determined by the order in which they are added, and you access items in a list using their index.

Dictionaries: Dictionaries are unordered collections of key-value pairs, where each item (value) is associated with a unique key. Unlike lists, the order of items in a dictionary is not guaranteed, and you access items in a dictionary using their keys rather than indices.

Here's a summary of the distinction:

List:

Ordered collection of items.

Accessed by index.

Mutability: Lists are mutable (modifiable).

Syntax: Defined using square brackets [ ].

Dictionary:

Unordered collection of key-value pairs.

Accessed by keys.

Mutability: Dictionaries are mutable (modifiable).

Syntax: Defined using curly braces { } with key-value pairs separated by commas (key: value).

In essence, lists are suitable for storing ordered collections of items, while dictionaries are ideal for storing data where the association between keys and values is important and order is not a concern.

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

Ans:

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

Here's how it looks:

spam = {'bar': 100}

value = spam['foo'] # This line will raise a KeyError

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

Ans:

The expressions 'cat' in spam and 'cat' in spam.keys() check for the presence of the key 'cat' in the dictionary spam, but they differ in how they perform the check:

'cat' in spam:

This expression checks if the key 'cat' exists in the dictionary spam.

If 'cat' is a key in spam, the expression evaluates to True; otherwise, it evaluates to False.

Example:

spam = {'cat': 1, 'dog': 2}

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

'cat' in spam.keys():

This expression checks if the key 'cat' exists in the keys of the dictionary spam.

It first retrieves all the keys of spam using the keys() method and then checks if 'cat' is in that list of keys.

If 'cat' is a key in spam, the expression evaluates to True; otherwise, it evaluates to False.

Example:

spam = {'cat': 1, 'dog': 2}

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

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

Ans:

The expressions 'cat' in spam and 'cat' in spam.values() both check for the presence of the value 'cat' in the dictionary spam, but they differ in how they perform the check:

'cat' in spam:

This expression checks if the key 'cat' exists in the keys of the dictionary spam.

If 'cat' is a key in spam, the expression evaluates to True; otherwise, it evaluates to False.

Example:

spam = {'cat': 1, 'dog': 2}

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

'cat' in spam.values():

This expression checks if the value 'cat' exists in the values of the dictionary spam.

It first retrieves all the values of spam using the values() method and then checks if 'cat' is in that list of values.

If 'cat' is a value in spam, the expression evaluates to True; otherwise, it evaluates to False.

Example:

spam = {'cat': 1, 'dog': 2}

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

7. What is a shortcut for the following code?

if 'color' not in spam:

spam['color'] = 'black'

Ans:

You can use the setdefault() method as a shortcut for the given code. The setdefault() method checks if a key exists in a dictionary. If the key does not exist, it sets the key to the specified default value and returns that value. If the key already exists, it returns the current value associated with the key without modifying the dictionary.

Here's how you can use setdefault() as a shortcut for the given code:

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

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

Ans:

To "pretty print" dictionary values in Python, you can use the pprint module's pprint() function. The pprint() function stands for "pretty print" and provides a more readable and visually appealing representation of complex data structures like dictionaries.

Here's how you can use pprint() to pretty print dictionary values:

import pprint

my\_dict = {'name': 'John', 'age': 30, 'city': 'New York'}

pprint.pprint(my\_dict)