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

Answer

An empty dictionary in Python is represented by a pair of curly braces {}
my_dict = {}

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

Answer

The value of a dictionary with the key 'foo' and the value 42 would be 42. In a dictionary, values are associated with specific keys, and you can access the value by referencing its corresponding key.

```
my_dict = {'foo': 42}
value = my_dict['foo']
print(value)  # Output: 42
```

In this example, the dictionary my_dict has a key-value pair where the key is 'foo' and the value is 42. By accessing the value using the key my_dict['foo'], you retrieve the value 42.

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

Answer

List:

- A list is an ordered collection of elements that can be of any data type. Elements in a list are stored in a specific order and accessed using an index. Lists are defined with square brackets [].
- Elements in a list are accessed using an index, which represents the position of the element in the list. The index starts from 0 for the first element, and negative indices can be used to access elements from the end of the list.

Dictionary:

- A dictionary is an unordered collection of key-value pairs. Each key-value pair is associated with a unique key, allowing for efficient lookup of values. Dictionaries are defined with curly braces {}.
- Elements in a dictionary are accessed using keys. Instead of numeric indices, keys are used to retrieve corresponding values. Keys can be of various data types such as strings, numbers, or tuples

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

Answer

If you 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.
Here's an example of how it would raise a KeyError:

```
spam = {'bar': 100}
value = spam['foo']  # Raises 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()?

Answer

'cat' in spam checks for the existence of the key directly within the dictionary. It evaluates to True if the key is present and False otherwise.

'cat' in spam.keys() first retrieves the keys of the dictionary using the keys() method, which returns a view object containing all the keys of the dictionary. Then, it checks if the key 'cat' exists within the obtained keys. Like 'cat' in spam, it evaluates to True if the key is present 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()?

Answer

'cat' in spam checks if the value 'cat' exists as a key in the dictionary spam. It evaluates to True if the value is a key in the dictionary and False otherwise.

'cat' in spam.values() checks if the value 'cat' exists within the values of the dictionary spam. It evaluates to True if the value is present in any of the dictionary's values and False otherwise.

7. What is a shortcut for the following code?

```
if ‘color’ not in spam:
    spam[‘color’] = ‘black’
```

Answer

A shortcut for the given code can be achieved using the setdefault() method available for dictionaries in Python. The setdefault() method allows you to set a default value for a key if the key is not already present in the dictionary.

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

In this code, spam.setdefault('color', 'black') checks if the key 'color' is present in the spam dictionary. If the key is not present, it sets the key to 'black' as the default value.

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

Answer

To "pretty print" dictionary values in Python, you can use the pprint module, which provides the pprint() function. The pprint() function allows for nicely formatted output of complex data structures, including dictionaries.

```
import pprint
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
pprint.pprint(my_dict)
```