Answer:

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

Answer: An empty dictionary in Python is represented by curly braces `{}`. The code for an empty dictionary looks like this:

empty\_dict = {}

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

Answer: If the key `'foo'` has the value `42` in a dictionary, it would look like this:

my\_dict = {'foo': 42}

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

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

1. Lists are ordered collections of items that are accessed by their index (starting from 0).

2. Dictionaries are unordered collections of key-value pairs, and elements are accessed using their unique keys instead of indexes.

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

Answer: To access `spam['foo']` and the key `'foo'` does not exist in the dictionary `spam`, Python will raise a `KeyError`. In this case, since `'foo'` is not a key in `spam`, to encounter 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()`?

Answer: Both expressions check for the existence of the key `'cat'` in the dictionary `spam`. However, the difference lies in how the expressions work:

1. `'cat' in spam` checks if the key `'cat'` exists in the dictionary `spam`. It returns `True` if the key is present and `False` otherwise.
2. `'cat' in spam.keys()` explicitly accesses the dictionary's keys and checks if `'cat'` exists among those keys. It also returns `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: expressions can check for the existence of the value `'cat'` in the dictionary `spam`. The difference lies in how the expressions work:

1. 'cat' in spam` checks if the value `'cat'` exists as a key in the dictionary `spam`. It returns `True` if there is a key with the value `'cat'` and `False` otherwise.
2. `'cat' in spam.values()` explicitly accesses the dictionary's values and checks if `'cat'` exists among those values. It returns `True` if the value is present and `False` otherwise.

7. What is a shortcut for the following code?

Answer:

if 'color' not in spam:

spam['color'] = 'black'

The shortcut for the above code is to use the `setdefault()` method:

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

The `setdefault()` method checks if the key `'color'` exists in the dictionary. If the key is present, it returns the corresponding value. If the key is not present, it adds the key-value pair to the dictionary with the default value provided (in this case, `'black'`).

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

Answer: To "pretty print" dictionary values in a well-formatted and readable way, you can use the `pretty print` module and its `prettyprint()` function.

import prettyprint

Then, use the `prettyprint()` function to print the dictionary:

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

prettyprint.prettyprint(my\_dict)

The `prettyprint()` function will display the dictionary in a more organized and human-readable format. This is especially useful when dealing with complex dictionaries with nested structures.