1. What does an empty dictionary's code look like?
Ans: my_dict = {}

2. What is the value of a dictionary value with the key 'foo' and the value 42?
Ans: 42 would be the value of a dictionary entry with the key "foo" and the value. Each key-value pair in a Python dictionary is separated by a colon; the key is used to look up the corresponding value. Therefore, you would use the syntax my_dict['foo'] to get the value linked to the key "foo," which would return the value 42.

3. What is the most significant distinction between a dictionary and a list?
Ans: The primary difference between a list and a dictionary in Python is that a list is an ordered collection of values, whereas a dictionary is an unordered collection of key-value pairs. While entries in a list are retrieved by their index, keys are utilised to look up equivalent values in dictionaries. Lists are surrounded in square brackets [] and elements are separated by commas, whereas dictionaries are encased in curly braces and each key-value pair is separated by a colon.

4. What happens if you try to access spam['foo'] if spam is {'bar': 100}?
Ans: Python will throw a KeyError if you attempt to read spam['foo'] when spam is set to "bar": 100 because the key 'foo' does not exist in the dictionary spam. Each key in a Python dictionary is distinct and is used to look up the appropriate item. When attempting to access a value using a key that isn't present in the dictionary, a KeyError will appear.

5. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.keys()?
Ans: If a dictionary is kept in the spam variable in Python, the phrases "cat" in spam and "cat" in spam will work.keys() is the same thing. Both formulas determine whether the word 'cat' is a key in the dictionary spam. Both expressions will return True if the key is present; otherwise, they will both return False.

Python dictionary's keys() method produces a view object containing the dictionary's keys, like 'cat' in spam.'Cat' is specifically verified as one of the keys by keys(). Python's dictionary implementation, however, checks to see if the key is in the dictionary, therefore "cat" in spam will likewise yield True if the spam key 'cat' exists.

6. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.values()?
Ans: While the expression 'cat' in spam checks to see if the key 'cat' is present in the dictionary spam.values() checks each value in the dictionary spam to see if the value 'cat' is present. As a result, the first expression verifies the key, whereas the second verifies the value.

7. What is a shortcut for the following code?

```
if 'color' not in spam:

spam['color'] = 'black'
```

Ans: A shortcut for the above code is to use the setdefault() method of the dictionary. So the equivalent code using setdefault() method would be:

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

This code checks if the key 'color' is present in the dictionary spam. If it is not present, it adds the key-value pair 'color':'black' to the dictionary. If the key is already present, it does not modify the dictionary.

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

Ans: You can "pretty print" dictionary values using the `pprint` module and its `pprint()` function.

First, you need to import the `pprint` module:

```
import pprint
```

Then, you can use the `pprint()` function to print a dictionary in a readable format:

```
my_dict = {'name': 'John', 'age': 30, 'city': 'New York'}
pprint.pprint(my_dict)
```

This will print the dictionary with each key-value pair on a separate line and indented for readability:

```
{'age': 30,
 'city': 'New York',
 'name': 'John'}
```