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

- An empty dictionary in Python is represented by a pair of curly braces with nothing inside: `{}`.

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

- The value of a dictionary with the key 'foo' and the value 42 would be accessed using dictionary indexing. Assuming the dictionary is assigned to the variable `spam`, you can access the value as `spam['foo']`, which would give you the value 42.

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

- The most significant distinction between a dictionary and a list is their structure and how they store data:
  - A dictionary is an **unordered collection of key-value pairs**. It uses unique keys to access corresponding values, allowing for efficient lookup. Keys in a dictionary must be unique, and they are typically immutable types like strings or numbers.
  - A list, on the other hand, is an **ordered collection of values**. It uses integer indices to access elements at specific positions in the list. Lists can contain duplicate values, and they can store mutable objects of any type.

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

- If you try to access `spam['foo']` when `spam` is `{'bar': 100}`, it will raise a **`KeyError`** because the key `'foo'` does not exist in the dictionary. The dictionary `spam` only contains the key `'bar'`, and trying to access a non-existent key will result in an error.

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

- The expression `'cat' in spam` **checks if the string `'cat'` exists as a key** in the dictionary `spam`. If `'cat'` is a key in `spam`, it will evaluate to `True`; otherwise, it will evaluate to `False`.
- The expression `'cat' in spam.keys()` also **checks if `'cat'` exists as a key in the dictionary `spam`.** It specifically checks the keys of the dictionary. The result will be the same as `'cat' in spam` since accessing `spam.keys()` returns a view object of the dictionary keys.

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

- The expression `'cat' in spam` checks if the string `'cat'` exists as a key in the dictionary `spam`. If `'cat'` is a key in `spam`, it will evaluate to `True`; otherwise, it will evaluate to `False`.
- The expression `'cat' in spam.values()` **checks if `'cat'` exists as a value** in the dictionary `spam`. It specifically checks the values of the dictionary. If `'cat'` is one of the values in `spam`, it will evaluate to `True`; otherwise, it will evaluate to `False`.

7. What is a shortcut for the following code?

```
if 'color' not in spam:
spam['color'] = 'black'
```

A shortcut for the given code can be achieved using the `setdefault()` method of dictionaries:

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

This method checks if the key `'color'` is already present in the dictionary `spam`. If it is present, it returns the corresponding value. If not, it adds the key-value pair `'color': 'black'` to the dictionary.

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

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

```
import pprint

dictionary = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
pprint.pprint(dictionary)
```

The `pprint()` function from the `pprint` module will display the dictionary in a more organized and visually appealing format. It can handle nested structures, making it convenient for printing complex dictionaries.