Here are the answers to your questions:

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

- An empty dictionary is created with a pair of curly braces:

```python

empty\_dict = {}

```

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

- If a dictionary has a key `'foo'` with a value `42`, accessing `dictionary['foo']` will return `42`.

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

- The most significant distinction is that dictionaries are key-value pairs where each key is unique and maps to a value, while lists are ordered collections of items indexed by integers. Dictionaries use keys for access, whereas lists use numeric indices.

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

- If you try to access `spam['foo']` and `'foo'` is not a key in `spam`, a `KeyError` will be raised because the key `'foo'` does not exist in the dictionary.

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

- Both expressions check if `'cat'` is a key in the dictionary. `'cat' in spam` is more straightforward and achieves the same result as `'cat' in spam.keys()`. The latter explicitly checks if `'cat'` is in the dictionary's keys, but both are functionally equivalent.

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

- `'cat' in spam` checks if `'cat'` is a key in the dictionary. `'cat' in spam.values()` checks if `'cat'` is a value in the dictionary. These expressions serve different purposes.

7. \*\*What is a shortcut for the following code?\*\*

```python

if 'color' not in spam:

spam['color'] = 'black'

```

- You can use the `setdefault` method to achieve the same result:

```python

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

```

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

- To "pretty print" dictionary values, you can use the `pprint` module's `pprint` function:

```python

import pprint

pprint.pprint(dictionary)

```