Assignment – 5

1. **Empty dictionary code:** `{}`

2. **Value of dictionary with key 'foo' and value 42:**

```
my_dict = {'foo': 42}  # Key: 'foo', Value: 42
value = my_dict['foo']  # Accessing the value with the key 'foo'
print(value)  # Output: 42
```

3. **Main difference between dictionary and list:**

- **Dictionaries:** Unordered collections of key-value pairs. Keys must be unique and immutable (e.g., strings, numbers, tuples). Values can be of any data type. Used to store data associated with specific keys.
- **Lists:** Ordered collections of elements. Elements can be accessed by their position (index). Used to store a sequence of items.

4. **Accessing a non-existent key:**

```
spam = {'bar': 100}
value = spam['foo']  # This will raise a KeyError since 'foo' is not a key
in spam
```

5. **'cat' in spam vs. 'cat' in spam.keys():**

- `'cat' in spam`: Checks if a key named 'cat' exists in the dictionary `spam`. Returns `True` if it exists, `False` otherwise.
- `'cat' in spam.keys()`: Checks if a key named 'cat' exists in the dictionary `spam`'s **key set**. Returns `True` if it exists, `False` otherwise. The `keys()` method returns a view of all the keys in the dictionary.

6. **'cat' in spam vs. 'cat' in spam.values():**

- `'cat' in spam`: Same as question 5 (checks for key existence).
- `'cat' in spam.values()`: Checks if a value named 'cat' exists in the dictionary `spam`'s **value collection**. Returns `True` if it exists, `False` otherwise. The `values()` method returns a view of all the values in the dictionary.

7. **Shortcut for adding a default value:**

```
spam.setdefault('color', 'black')  # This does the same as the if-else block
```

8. **Pretty printing dictionary values:**

- Module: `pprint`
- Function: `pprint.pprint(my_dict)`

This will print the dictionary with indentation and line breaks for better readability.