You can sort a dictionary in Python based on its keys or values. Here's how you can do it:

1. \*\*Sorting by Keys\*\*:

```python

my\_dict = {'b': 3, 'a': 1, 'c': 2}

sorted\_dict = dict(sorted(my\_dict.items()))

print(sorted\_dict)

```

This code snippet sorts the dictionary `my\_dict` based on its keys and assigns the result to `sorted\_dict`.

2. \*\*Sorting by Values\*\*:

```python

my\_dict = {'b': 3, 'a': 1, 'c': 2}

sorted\_dict = dict(sorted(my\_dict.items(), key=lambda item: item[1]))

print(sorted\_dict)

```

This code snippet sorts the dictionary `my\_dict` based on its values. The `key` parameter in the `sorted()` function specifies a function to be called on each item for sorting. In this case, we use a lambda function that returns the second element of each tuple (which is the value in the dictionary).

In both cases, the `sorted()` function returns a list of tuples, which is then converted back into a dictionary using the `dict()` constructor.

Remember, dictionaries in Python are inherently unordered collections, so the order of elements in a dictionary doesn't have any meaning. However, if you need a sorted representation of a dictionary, you can use the techniques mentioned above.