

1. My set and dict types would not implement the `__ror__` method because of the fact that the set and dictionary are not ordered types. While adding two classes for instance, the order that you call a certain method matters. When calling an or operation on two integers, the order of which integer is put first changes the resulting value. However, in the set and dictionary, this does not matter since the order of the dictionary and set changes regardless, and is completely unordered.
2. I would implement the `__ior__` method for both the set and the dictionary because it should be possible for you to combine two dictionaries or sets through a union operation. Also, in both cases, they deal with the same situations in that they eliminate any keys that are the same, and they only add the elements that are not the same, and the or operation already does not combine elements that are the same, which goes with what a set or dictionary deals with.