Daily Coding Problem #173

Problem

This problem was asked by Stripe.

Write a function to flatten a nested dictionary. Namespace the keys with a period.

For example, given the following dictionary:

```
{
    "key": 3,
    "foo": {
        "a": 5,
        "bar": {
            "baz": 8
        }
    }
}
```

it should become:

```
{
    "key": 3,
    "foo.a": 5,
    "foo.bar.baz": 8
}
```

You can assume keys do not contain dots in them, i.e. no clobbering will occur.

Solution

One solution is to iterate over every key-value in the dict, check whether it's a dictionary, and if it, recursively flatten it and add its values prefixed with its current key.

```
def flatten(d, separator='.'):
    new_dict = {}
    for key, val in d.items():
        if isinstance(val, dict):
            flattened_subdict = flatten(val)
            for flatkey, flatval in flattened_subdict.items():
                new_dict[key + separator + flatkey] = flatval
        else:
            new_dict[key] = val

    return new_dict
```

This takes O(n) time since we have to look at every key and value in the original dict.