Actions and Reducers 101 Lab

Now that we've dug into reducers some more, let's refactor our current reducer to see if we can make it more in line with some best practices.

Extracting the reducer

1. Create a new file in your store folder called reducers.js. Move your reducer function out of the store.js file and into here. Make sure you export your reducer function from reducers.js and import it into store.js.
2. Run and test to make sure it still works.
3. Take a look at your reducer function. Did you write it to be a series of if-else statements? Refactor it to be a switch case, switching on action.type.
4. Again, run and test to make sure it still works.

Refactoring the cases

5. Look at the case statement for SET_EMAIL. Do your best to make it a single return statement. This will make the file easier to read. (Hint: The spread operator on objects and arrays may help you to condense the code.)

Of course you and your partner feel free to discuss and use as many lines as you like so that it is optimally easy to maintain -- that's the most important thing. But it is good to stretch yourself. And if you look at how the best Redux developers are coding, they're usually a single return statement. So you might try to make yourself get accustomed to this pattern. You'll probably be seeing it a lot.

6. Run and test, making sure that the SET_EMAIL case still works.
7. Now do the same for SET_CELL and for SET_NAME_FIRST if you have those written.

Catching unusual situations in the reducer

Note that if the user provides any action type besides the ones we've coded for here, we are returning an undefined. This is bad. It would mean that if you misspelled an action type, the state would become undefined at run time and you'd destroy data. Let's fix that.

8. Add a default case to your switch. The default should just return the state that was passed in to the reducer.

Nice. That is fixed. Now let's handle an empty action.

9. Add an if statement to the top. If the action is falsey, return state unchanged.
10. Run and test. Make sure it all still works.

If so, you can be finished with this lab.

11. Bonus! If you have extra time, go ahead and implement the reducer case for street address. (Hint: look in the user.json file to see what data point will need to be changed.)