An important detour into parametrization

So...

So let's create the following data model:

data/shirts.py

```
shirts = {
        'blue_s': {'id': 'blue_s', 'color': 'blue', 'size': 'small'},
        'blue_m': {'id': 'blue_m', 'color': 'blue', 'size': 'medium'},
        'blue_l': {'id': 'blue_l', 'color': 'blue', 'size': 'large'},
        'blue_xl': {'id': 'blue_xl', 'color': 'blue', 'size': 'extra-large'},
        'red_s': {'id': 'red_s', 'color': 'red', 'size': 'small'},
        'red_m': {'id': 'red_m', 'color': 'red', 'size': 'medium'},
        'red_l': {'id': 'red_l', 'color': 'red', 'size': 'large'},
        'red_xl': {'id': 'red_xl', 'color': 'red', 'size': 'extra-large'}
}
```

An important detour into parametrization

Parametrization is a test design superpower. If you have series of tests that vary only in the specific test data, for example what should happen with a blue XL t-shirt vs. a red XL t-shirt, then you should parametrize.

Parametrization works roughly like this:

1. Identify a scenario where you are dealing with essential the same test but with different data or action parameters.
2. Generalize the test steps.
3. Build a data model that can be iterated over.

With pytest, for a test method that is parametrized, during test collection a new test method for each parameter is dynamically created.