

And generate parametrized
test cases, dynamically!

Component: Application Wrappers

The standard application wrapper is the Page Object Model, where pages are described in a class structure. The conceit is that these classes “know about themselves and their affordances”.

The goal is to abstract out the specifics of the interactions with these pages from the test cases to supporting code, making it easier to maintain the interface to the application under test.

And generate parametrized test cases, dynamically!

```
$ pytest tests/test_params.py --collect-only
===== test session starts =====
collected 8 items
<Module 'test_params.py'>
  <Class 'SomeTests'>
    <Instance '()'>
      <Function 'test_add_shirt[blue_s]'>
      <Function 'test_add_shirt[blue_m]'>
      <Function 'test_add_shirt[blue_l]'>
      <Function 'test_add_shirt[blue_xl]'>
      <Function 'test_add_shirt[red_s]'>
      <Function 'test_add_shirt[red_m]'>
      <Function 'test_add_shirt[red_l]'>
      <Function 'test_add_shirt[red_xl]'>
```