

Component: Test Fixtures


```
@pytest.yield_fixture(scope='session')
def driver(request):
    from selenium import webdriver
    driver = webdriver.Chrome()
    driver.implicitly_wait(10)
    yield driver
    driver.quit()
```



```
def test_something(self, driver):  
    page = driver.get('https://cnn.com')
```


Component: Test Data

The specific data used in your test cases will typically follow a migration route starting inside your test cases, and moving into a central “repository” in your framework.

Method-local test data:

```
def test_add_xl-blue-shirt(self):  
    shirt = {'color': 'blue', 'size': 'xl'}  
    # navigate to the product page for xl blue shirts  
    # verify that you can find xl blue shirt  
    # add xl blue shirt to cart  
    # verify that xl blue shirt is in cart
```

Module-local test data:

```
shirt = {'color': 'blue', 'size': 'xl'}  
def test_add_shirt(self, shirt):  
    # navigate to the product page for `shirt`  
    # verify that you can find `shirt`  
    # add `shirt` to cart  
    # verify that `shirt` is in cart
```


Component: Test Fixtures

Let's create a fixture that initiates the Chrome driver to be used in our tests.

tests/conftest.py

```
@pytest.yield_fixture(scope='session')
def driver(request):
    from selenium import webdriver
    driver = webdriver.Chrome()
    driver.implicitly_wait(10)
    yield driver
    driver.quit()
```

To hook this fixture up to a particular test method, just add the fixture name as a parameter. For example:

```
def test_something(self, driver):
    page = driver.get('https://cnn.com')
```