

What is Page Object Model in Selenium?

Page Object Model, also known as POM is a design pattern in Selenium that creates an object repository for storing all web elements. It is useful in reducing code duplication and improves test case maintenance.

In Page Object Model, consider each web page of an application as a class file. Each class file will contain only corresponding web page elements. Using these elements, testers can perform operations on the website under test.

Advantages of Page Object Model

Helps with easy maintenance: POM is useful when there is a change in a UI element or there is a change in an action. An example would be if a drop down menu is changed to a radio button.

In this case, POM helps to identify the page or screen to be modified. As every screen will have different java files, this identification is necessary to make the required changes in the right files. This makes test cases easy to maintain and reduces errors.

Helps with reusing code: As already discussed, all screens are independent. By using POM, one can use the test code for any one screen, and reuse it in another test case. There is no need to rewrite code, thus saving time and effort.

Readability and Reliability of scripts: When all screens have independent java files, one can easily identify actions that will be performed on a particular screen by navigating through the java file. If a change must be made to a certain section of code, it can be efficiently done without affecting other files.

Data driven framework:

This framework is used when the application contains lot of forms and lot of test data to be passed.

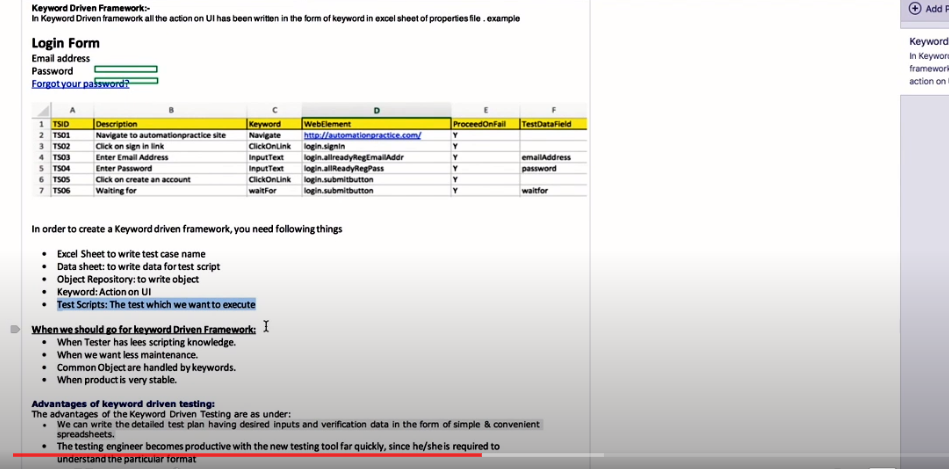
Keyword driven framework:

When to use:

when testers have little coding knowledge.

When we want less maintenance.

When product is very stable.



Hybrid framework:

Mixture of keyword driven, page object, data driven etc. If the application has mixture of static pages (keyword), dynamic pages (page object) and forms we can use this framework.

Framework design patterns: https://blog.testproject.io/2020/06/29/design-patterns-in-test-automation/