POM

- Page Object Model, also known as POM,
- It is a design pattern that creates an object repository for storing all web elements.
- For each web page in the application, there should be a corresponding Page Class.
- These page classes contain all the web elements of that specific web page and action methods which perform operations on those WebElements.
- The name of these methods should be given as per the task they are performing,
- We are going to use the encapsulation concept in which data members should be private and methods (member functions) as public.
- We are going to create two packages
    - Page Layer - It contains web elements(Object Repository) and action methods
    - Test Layer - Test cases

Advantages -

- **Reusability** - reusing code - We can reuse the page class if required in different test cases which means we don't need to write code again.
- **Maintainability** - Test case and page class are different from each other which means we can easily update the code if any new web element is added or existing one updated.
- **Readability** - Page code is separated from test code which helps to improve code readability.

Disadvantages -

POM initializes data members before performing action on it. Sometimes an application may contain some web elements which are hidden or lazily loaded so we can find those webelement but are not able to perform actions on those web elements. So to overcome this problem we have extended form of POM called Page Factory.

Two ways -
Simple POM
Initialize all web elements at the same time.

Page Factory in Selenium

- Page Factory is a class provided by Selenium WebDriver to support Page Object Design patterns.
- In Page Factory, we are going to use
    - @FindBy annotation.
    - initElements method
- **@FindBy:** An annotation used in Page Factory to locate and declare web elements using different locators.
    - @FindBy(locator="value")
      WebElement element;
    - Below are locators that can be used:
    - Id, name, className, xpath, TagName, LinkText, PartialLinkText
- **initElements():** initElements is a static method in Page Factory class. Using the initElements method, one can initialize all the web elements located by @FindBy annotation.
- **lazy initialization:** AjaxElement Locator Factory is a lazy load concept in Page Factory.
    - This is used to identify web elements only when they are used in any operation or activity.
    - The timeout of a web element can be assigned to the object class with the help of the AjaxElementLocatorFactory.

Why Page Object Model?

Starting an UI Automation in Selenium WebDriver is NOT a tough task. You just need to find elements, perform operations on it.

Page Object Model	**Page Factory**
Finding web elements using **By**	Finding web elements using **@FindBy**
POM does not provide lazy initialization	Page Factory does provide lazy initialization
Page Object Model is a design pattern	PageFactory is a class that provides the implementation of the Page Object Model design pattern
In POM, one needs to initialize every page object individually	In PageFactory, all page objects are initialized by using the **initElements()** method