Test-Driven Development (TDD) is a software development approach where tests are written before the actual code. In the context of Selenium, which is a popular tool for automating web browsers, TDD can be implemented by following these steps:

1. Setup Your Testing Environment

Ensure you have a programming language installed (like Java, Python, etc.), a testing framework (like JUnit, TestNG for Java; unittest, pytest for Python), and the Selenium WebDriver library.

2. Define Test Cases

Identify the functionalities or scenarios you want to test on your web application.

3. Write the Test Case

Write a failing test case before writing the actual code. For example (using Python with pytest):

pythonCopy code

Explain

# test\_sample.py

from selenium import webdriver

def test\_title():

driver = webdriver.Chrome() # or Firefox(), etc.

driver.get('[https://www.example.com](https://www.example.com/)')

assert driver.title == 'Example Domain'

driver.quit()

4. Run the Test

Execute the test you just wrote. Since there's no implementation yet, it will fail.

5. Write Implementation Code

Write the actual code that fulfills the requirements specified in the failing test case.

pythonCopy code

# implementation

from selenium import webdriver

def get\_title():

driver = webdriver.Chrome()

driver.get('[https://www.example.com](https://www.example.com/)')

title = driver.title

driver.quit()

return title

6. Refactor

Refactor your code if necessary to improve its structure and maintainability while ensuring that the test continues to pass.

7. Repeat

Repeat this cycle for each new functionality or scenario.