Additional Exercises

- These exercises are designed for people who get ahead of the rest of the course.
- For practising during breaks if you want to maximise the learning opportunities on the course
- For practising at home to reinforce your understanding of WebDriver and provide more opportunities to tackle automation problems
- There are deliberately a lot of these exercises, do not expect to do them all during the course

Exercise: Create a Fail to Logon Test

- Randomly generate an invalid password for a known user
- Test for blank password
- Refactor the test to move repeated code into local methods

Exercise: Refactor Login Tests

- Create a 'TracksUser' object
- Add a username and password to the TracksUser
- Add a login method to the TracksUser object, which navigates to the page and logs in

Exercise: Basic Synchronisation with Expected Conditions

- Add additional waits to the page to ensure that:
  - The title is correct
  - The button is clickable
  - The url is correct
- Add additional waits after login to ensure that
  - A Logout link is present
  - The Form to add a New Action is available

Exercise: Refactor waits into Page Objects

- Login as a 'normal' user and check that they don't have access to the Admin functionality
- Login as an 'Admin' user and check thaty they do have access to the Admin functionality
- Create a domain object for Admin User as well as a 'Normal' User
- Refactor the test code to use Page Objects and domain objects

Exercise: Refactor Fail to Login Test for JUnit

- Add some Logging to your test which outputs the random username and password you used to fail to login
- If you have randomly generated the password then have that test run multiple times to increase the possibility of the random test exposing a bug

Exercise: Create a Logout Method

- Write a logout test that clicks on the logout link and logouts as a side-effect of this navigation
- Write a logout method for the User object
- Write a test that uses the User object to login and out

Exercise Drop Down Menu Navigation

- Wait for the top menu to be clickable before you click on it
- Create an Enum or Constants for the Menu items to avoid having literal strings in the test

Exercise: Actions Drop Down Menu

- Experiment with different action sequences
  - e.g.
    - moveToElement(topNav).click()
    - click(topNav)
- Build and save the action in a variable, and then perform it.

Exercise: Refactor the Navigation Menu

- Automate “View \ Done” and wait until the correct page has loaded
- Automate the menu items without sub menus e.g. “Home”, “Starred”, “Projects”
- Automate clicking on Search Menu
- Can you make the Navigation Menu API consistent for top menus without sub menus, menus with sub menus, help, and search?

Exercise: Convert TracksDashboardHomePage

- Add any additional waits that you consider appropriate to the page objects
- Add any additional page objects based on the navigation menus you automated
- Add waits for the additional pages
- Randomly navigate around the site using the menu items and wait for the appropriate page to load

Exercise: Create a test to add a next action

- Automate clicking on the pop up calendar [Done] button
- Automate the pop up calendar
  - Wait for it to appear
  - Click on the Month, Year and Day
  - Click Done
  - Check the date in the field is what you selected on the popup calendar

Exercise: Create a method to count the number of next actions

- Refactor the Add Next Action form into a page object
- Create a “Next Action” domain object
- Randomly generate valid data for the “Next Action” domain object
- Use the domain object and page object in the test
- Create an “addNextAction” method on the TracksUser class, and use the TracksUser class in a test

Exercise: Write code to wait for action count to increase

- Write the expected condition as a local method
- Write the expected condition as inline code
- Write the expected condition as a class

Exercise: Cookies

- Write a test that checks the correct cookies are created when you login
- Write a test that check the correct cookies are deleted when you logout
- Write a test that checks the session is invalidated and you are logged out when the session cookie values are amended after logging in