**ДОМАШНЯЯ РАБОТА №4**

*На уроке мы углубляли наши познания в XPath. В качестве базы мы использовали тестовый сценарий на базе собраного по сайту https://habr.com/ru/post/262929/.*

*В качестве домашнего задания нужно обновить тестовый сценарий и сделать так, чтобы он проверял отображаемые варианты автоподстановки поиска. Необходимо проверить количество выданных вариантов автоподстановки, наличие ссылки на сайт world of tanks и иконки слева,а также список авто корректировок поиска.*

*Напомню условия задачи:*

|  |
| --- |
| *Задача:*   1. *Исправить тестовый сценарий таким образом, чтобы он проверял результаты отображаемые варианты автоподстановки поиска.* 2. *Необходимо проверить количество выданных вариантов автоподстановки (на скриншоте 10)* 3. *Необходимо проверить существование теста “World of Tanks”, иконки и текста “Online game” среди выданных вариантов автоподстановки* 4. *Необходимо проверить список авто корректировок поиска (на скриншоте их 5 штук, среди которых присутствует “world cup 2022”)* |
| *Ограничения:*   1. *Тест должен содержать один файл \*.feature.* 2. *Тест должен содержать 2 сценария (проверка в 2х браузерах)..* |
| *Подсказки, найденные на паре:*   1. *Смотрите скриншоты в чат* |
| *Что будет оцениваться:*   1. *Работа тестового сценария* 2. *Именование переменных* 3. *Наличие функций и их именование.* 4. *Структура проекта (насколько легко разобраться в том что и как работает)* |

Скрипт работает так:

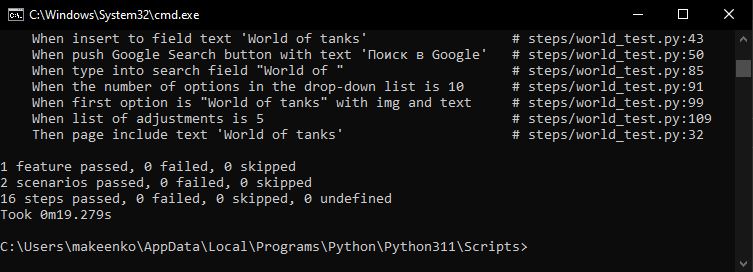
* Открывает Google в двух браузерах (Chrome, Edge);
* Находит поле ввода запроса, вводит запрос, кликает кнопку поиска;
* Очищает поле поискового запроса и вводит начальную часть запроса;
* Сравнивает количество опций выпадающего списка с ожидаемым количеством;
* Проверяет, что первый вариант выпадающего меню содержит изображение, ожидаемый текст заголовка и подсказки;
* Сравнивает количество опций корректировки запроса в выпадающем меню с ожидаемым количеством;
* Проверяет, что страница результатов поиска содержит текст поискового запроса.

Файл w.feature

|  |
| --- |
| Feature: Проверка результатов поиска "World of tanks" через Google в браузерах Chrome и Edge  Scenario: Поиск запроса "World of tanks" в Google через Chrome  Given on Chrome website 'https://www.google.com/'  When insert to field text 'World of tanks'  When push Google Search button with text 'Поиск в Google'  When type into search field "World of "  When the number of options in the drop-down list is 10  When first option is "World of tanks" with img and text  When list of adjustments is 5  Then page include text 'World of tanks'    Scenario: Поиск запроса "World of tanks" в Google через Edge  Given on Edge website 'https://www.google.com/'  When insert to field text 'World of tanks'  When push Google Search button with text 'Поиск в Google'  When type into search field "World of "  When the number of options in the drop-down list is 10  When first option is "World of tanks" with img and text  When list of adjustments is 5  Then page include text 'World of tanks' |

Файл world\_test.py

|  |
| --- |
| 1. **from** behave **import** **\*** 2. **from** selenium **import** webdriver 3. **from** selenium.webdriver.common.keys **import** Keys 4. **from** selenium.webdriver.common.by **import** By 5. **from** selenium.webdriver.support.ui **import** WebDriverWait 6. **from** selenium.webdriver.support **import** expected\_conditions as EC 7. **from** selenium.webdriver.chrome.options **import** Options 8. **import** time 10. #=================== GENERAL FUNCTIONS ==================# 12. #Откроем главную страницу в Chrome. Передадим в качестве аргумента адрес страницы. 13. @given("on Chrome website '{url}'") 14. **def** step(context, url): 15. #Измените строку, для выполнения теста в другом браузере 16. chrome\_options **=** Options() 18. context.browser **=** webdriver.Chrome() 19. context.browser.maximize\_window() 20. context.browser.get(url)  23. #Откроем главную страницу в Edge. Передадим в качестве аргумента адрес страницы. 24. @given("on Edge website '{url}'") 25. **def** step(context, url): 26. #Измените строку, для выполнения теста в другом браузере 27. context.browser **=** webdriver.Edge() 28. context.browser.maximize\_window() 29. context.browser.get(url)  32. #Проверим, что мы на странице с результатами поиска, есть некоторый искомый текст 33. @then("page include text '{text}'") 34. **def** step(context, text): 35. WebDriverWait(context.browser, 120).until( 36. EC.presence\_of\_element\_located((By.XPATH, '//\*[contains(text(), "%s")]' **%** text)) 37. ) 38. **assert** context.browser.find\_element(By.XPATH,'//\*[contains(text(), "%s")]' **%** text)  41. #====================== GOOGLE ===========================# 43. #Теперь введём запрос в Google 44. @when("insert to field text '{text}'") 45. **def** step(context, text): 46. context.browser.implicitly\_wait(4) 47. context.browser.find\_element(By.CSS\_SELECTOR, '[name="q"]').send\_keys(text)  50. #Теперь нажмем на кнопку "Поиск в Google" 51. @when("push Google Search button with text '{text}'") 52. **def** step(context, text): 53. context.browser.find\_element(By.CSS\_SELECTOR, '[name="btnK"]').click()  56. # =======================  WORLD OF =========================# 58. # Вводим текст в поисковую строку, чтобы получить выпадающий список 59. @when ('type into search field "{text}"') 60. **def** step(context, text): 61. context.browser.find\_element(By.CSS\_SELECTOR, '[name="q"]').clear() 62. context.browser.find\_element(By.CSS\_SELECTOR, '[name="q"]').send\_keys(text)  65. # Ищем количество элементов выпадающего списка строки поиска 66. @when ('the number of options in the drop-down list is 10') 67. **def** step(context): 68. OptionsInList **=** context.browser.find\_elements(By.XPATH,'//div[@class="lnnVSe" and @role="option" and @aria-description]') 70. **assert** int(len(OptionsInList)) **==** 10  73. # Проверяем наличие в первом пункте выпадающего меню текста, картинки и подсказки 74. @when ('first option is "World of tanks" with img and text') 75. **def** step(context): 76. HeadTextOnFirst **=** context.browser.find\_element(By.XPATH,'(//span[contains(text(), "world of tanks")])[1]').is\_displayed() 77. ImageOnfirst **=** context.browser.find\_element(By.XPATH,'(//div[contains(@class, "sbic")])[1]').is\_displayed() 78. SmallTextOnFirst **=**  context.browser.find\_element(By.XPATH, "(//\*[contains(text(), 'World of — Серия видеоигр')])[1]").is\_displayed() 80. **assert** HeadTextOnFirst **==** True **and** ImageOnfirst **==** True **and** SmallTextOnFirst **==** True  83. # Проверяем количество корректировочных пунктов выпадающего меню 84. @when('list of adjustments is 5') 85. **def** step(context): 86. ListOfAdjustments **=** context.browser.find\_elements(By.XPATH,'//div[@class="sbic sb43"]') 88. **assert** int(len(ListOfAdjustments)) **==** 5 |



Судя по отчёту в консоли, оба сценария и 16 шагов прошли успешно.