

Workshop 4: Handling Web Elements

หัวข้อหลัก:

1. **Basic Handling:**
 - **Dropdown**
 - **Checkbox**
 - **Radio Buttons**
2. **Advanced Handling:**
 - **Multi-Select Dropdown**
 - **Cookies**
 - **Drag-and-Drop**
3. **Error Handling:**
 - การจัดการข้อผิดพลาด (**Error Handling**)
 - การสร้าง **Screenshots** เมื่อเกิดข้อผิดพลาด

1. Basic Handling

1.1 Dropdown

- ตัวอย่างการจัดการ **Dropdown**:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select

service = Service('path_to_chromedriver')
driver = webdriver.Chrome(service=service)

driver.get("https://demo.seleniumeasy.com/basic-select-dropdown-demo.html")

# จัดการ Dropdown
dropdown = Select(driver.find_element("id", "select-demo"))
dropdown.select_by_visible_text("Friday")

# พิมพ์ค่าที่เลือก
print("Selected option:", dropdown.first_selected_option.text)

driver.quit()

from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select

# ระบุเส้นทาง ChromeDriver
service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')
driver = webdriver.Chrome(service=service)
```

```
# เปิดหน้า W3Schools Dropdown ตัวอย่าง
try:
    driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select")

    # สลับไปยัง iframe ที่มี Dropdown
    driver.switch_to.frame("iframeResult")

    # ค้นหา Dropdown และเลือกตัวเลือก
    dropdown = Select(driver.find_element("tag name", "select"))
    dropdown.select_by_visible_text("Opel") # เลือกตัวเลือก "Opel"

    # พิมพ์ค่าที่เลือก
    print("Selected option:", dropdown.first_selected_option.text)

except Exception as e:
    print(f"Error: {e}")

finally:
    # ปิดเบราว์เซอร์
    driver.quit()
```

1.2 Checkbox

- ตัวอย่างการจัดการ **Checkbox**:

#1.2 Checkbox

#ตัวอย่างการจัดการ **Checkbox**:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select, WebDriverWait
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC

try:
    # ระบุเส้นทางของ ChromeDriver
    service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')
    driver = webdriver.Chrome(service=service)

    # เปิดเว็บไซต์ใหม่ที่มี Dropdown
    driver.get("https://practice.expandtesting.com/dropdown?utm_source=chatgpt.com")
```

```
# รอให้ Dropdown ปรากฏ
dropdown_element = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "dropdown"))
)

# Handle Dropdown
dropdown = Select(dropdown_element)

# ตรวจสอบตัวเลือกทั้งหมดที่มีใน Dropdown
options = [option.text for option in dropdown.options]
print("Available options:", options)

# เลือกตัวเลือก (ปรับให้ตรงกับตัวเลือกใน Dropdown)
test_option = "Option 2" # ปรับตามตัวเลือกที่คุณต้องการ
dropdown.select_by_visible_text(test_option)

# ตรวจสอบว่าตัวเลือกที่เลือกถูกต้อง
selected_option = dropdown.first_selected_option.text
if selected_option == test_option:
    print(f"Test Passed: Selected option is '{selected_option}' as expected.")
else:
    print(f"Test Failed: Selected option is '{selected_option}', expected '{test_option}'.")

except Exception as e:
    print(f"Test Failed: An error occurred - {str(e)}")

finally:
    # ปิดเบราว์เซอร์
    driver.quit()
```

1.3 Radio Buttons

- ตัวอย่างการจัดการ **Radio Buttons:**

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By

service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')
driver = webdriver.Chrome(service=service)

try:
    driver.get("https://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_radio")

    # ค้นหา iframe ทั้งหมด
    iframes = driver.find_elements(By.TAG_NAME, "iframe")
    print(f"Found {len(iframes)} iframes.")

    for i, iframe in enumerate(iframes):
        driver.switch_to.frame(iframe)
        print(f"Switched to iframe {i}")

        # ค้นหา Nested iframe ภายใน iframe
        nested_iframes = driver.find_elements(By.TAG_NAME, "iframe")
        print(f"Nested iframes in iframe {i}: {len(nested_iframes)}")

        for j, nested_iframe in enumerate(nested_iframes):
            driver.switch_to.frame(nested_iframe)
            print(f"Switched to nested iframe {j} in iframe {i}")
            try:
                # ค้นหา Radio Button
                radio_button = driver.find_element(By.XPATH, "//*[@value='CSS']")
                radio_button.click()
                print("Radio button selected:", radio_button.is_selected())
                break
            except Exception as e:
                print(f"No radio button in nested iframe {j} of iframe {i}: {e}")
            finally:
                # กลับไปยัง iframe หลัก
                driver.switch_to.parent_frame()

        # กลับไปยัง context หลัก
        driver.switch_to.default_content()

except Exception as e:
    print(f"Error occurred: {e}")

finally:
    driver.quit()
    print("Browser closed.")
```

2. Advanced Handling

2.1 Multi-Select Dropdown

- ตัวอย่างการจัดการ **Multi-Select Dropdown**:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By

# ระบุเส้นทาง ChromeDriver
service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')
driver = webdriver.Chrome(service=service)

try:
    # เปิดเว็บไซต์ W3Schools Dropdown

    driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select_multiple")
    print("Website opened successfully.")

    # สลับไปยัง iframe
    driver.switch_to.frame("iframeResult")

    # ค้นหา Multi-Select Dropdown
    multi_select = Select(driver.find_element(By.TAG_NAME, "select"))

    # เลือกตัวเลือกใน Dropdown
    multi_select.select_by_visible_text("Volvo")
    multi_select.select_by_visible_text("Opel")

    # พิมพ์ค่าที่เลือกทั้งหมด
    selected_options = [option.text for option in multi_select.all_selected_options]
    print("Selected options:", selected_options)

except Exception as e:
    print(f"Error occurred: {e}")

finally:
    driver.quit()
    print("Browser closed.")
```

2.2 Cookies

- ตัวอย่างการจัดการ **Cookies**:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service

try:
    # ตั้งค่า ChromeDriver
    service = Service('C:\\Drivers\\Chrome\\chromedriver-
win64\\chromedriver.exe')
    driver = webdriver.Chrome(service=service)

    # เปิดเว็บไซต์
    driver.get("https://www.google.com")
    print("Website opened successfully.")

    # จัดการ Cookies
    driver.add_cookie({"name": "test_cookie", "value": "test_value"})
    print("Cookies:", driver.get_cookies())

    # ลบ Cookie
    driver.delete_cookie("test_cookie")
    print("Cookies after deletion:", driver.get_cookies())

except Exception as e:
    print(f"Error occurred: {e}")

finally:
    driver.quit()
    print("Browser closed.")
```

2.3 Drag-and-Drop

- ตัวอย่างการจัดการ Drag-and-Drop:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains

try:
    service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')
    driver = webdriver.Chrome(service=service)

    driver.get("https://the-internet.herokuapp.com/drag_and_drop")
    print("Website opened successfully.")

    source = driver.find_element(By.ID, "column-a")
    target = driver.find_element(By.ID, "column-b")

    actions = ActionChains(driver)
    actions.drag_and_drop(source, target).perform()
    print("Drag and Drop completed.")

except Exception as e:
    print(f"Error occurred: {e}")

finally:
    driver.quit()
    print("Browser closed.")
```

3. Error Handling & Screenshots (20 นาที)

3.1 การจัดการข้อผิดพลาด

- ตัวอย่าง Error Handling:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By

# กำหนดเส้นทาง ChromeDriver
service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')

try:
    # เริ่ม ChromeDriver
    driver = webdriver.Chrome(service=service)
    driver.get("https://www.google.com")

    # ค้นหา Search Bar
    search_bar = driver.find_element(By.NAME, "q")
    search_bar.send_keys("Error Handling Example")
    search_bar.submit()

    # แสดงข้อความเมื่อสำเร็จ
    print("Search completed successfully.")

except Exception as e:
    # แสดงข้อผิดพลาด
    print("An error occurred:", str(e))

finally:
    # ปิดเบราว์เซอร์
    try:
        driver.quit()
    except Exception as e:
        print("Error closing browser:", str(e))
```


3.2 การสร้าง Screenshots

- ตัวอย่างการบันทึก **Screenshot** เมื่อเกิดข้อผิดพลาด:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By

try:
    service = Service('C:\\Drivers\\Chrome\\chromedriver-win64\\chromedriver.exe')
    driver = webdriver.Chrome(service=service)

    driver.get("https://www.google.com")
    print("Website opened successfully.")

    # ค้นหา Search Bar และพิมพ์ข้อความ
    search_bar = driver.find_element(By.NAME, "q")
    search_bar.send_keys("Selenium Test")
    search_bar.submit()
    print("Search completed successfully.")

except Exception as e:
    print("An error occurred:", str(e))
    screenshot_path =
"C:\\Users\\chatp\\OneDrive\\Data\\OneDrive\\รูปภาพ\\ภาพหน้าจอ\\error_screenshot.png"
    driver.save_screenshot(screenshot_path)
    print(f"Screenshot saved at: {screenshot_path}")

finally:
    try:
        driver.quit()
    except Exception as e:
        print("Error while closing the browser:", str(e))
```