

## Work3:Locating Web Elements

### 3.1 การใช้ Selectors (ID, Name, XPath, CSS)

ทำความเข้าใจ Selectors:

- เป็นเทคนิคในการระบุ Web Elements บนหน้าเว็บ.
- ใช้ใน Selenium เพื่อโต้ตอบกับ Web Elements.

ประเภทของ Selectors:

- ID: เร็วและเฉพาะเจาะจงที่สุด.
- Name: คล้าย ID แต่ไม่ค่อยเฉพาะเจาะจง.
- Class Name: ใช้สำหรับการเลือกกลุ่มของ Elements.
- XPath: ทรงพลัง ใช้ค้นหา Element แบบเชิงโครงสร้าง.
- CSS Selector: เร็วและยืดหยุ่นในกรณีที่ XPath ซับซ้อน.

กิจกรรม:

- ใช้ Selectors เพื่อค้นหา Element เช่น ID, Name, Class, XPath, และ CSS
- สร้างสคริปต์ค้นหา Element บนหน้า Google

Key Learning:ใช้ Name เมื่อ ID ไม่มี.

- `find_element("name", "q")`

##ค้นหา Search Bar บน Google:

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

```
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time # สำหรับเพิ่มเวลาหยุดชั่วคราว

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

try:
    # เปิดหน้า Google
    driver.get("https://www.google.com")
    print("Opened Google homepage.")

    # เพิ่มเวลารอหน้าเว็บโหลด (หากจำเป็น)
    time.sleep(3)

    # ใช้ Explicit Wait เพื่อรอให้ Search Bar ปรากฏ
    wait = WebDriverWait(driver, 10)
    search_bar = wait.until(EC.presence_of_element_located((By.NAME, "q")))
    print("Search bar is ready.")

    # กรอกข้อความใน Search Bar
    search_query = "Automated Testing"
    search_bar.send_keys(search_query)
    print(f"Searching for: {search_query}")
```

# กด Enter เพื่อค้นหา

```
search_bar.submit()
```

# รอให้หน้าแสดงผลการค้นหา

```
wait.until(EC.presence_of_element_located((By.ID, "search")))
```

```
print("Search completed. Results displayed.")
```

except Exception as e:

```
print(f"Error occurred: {e}")
```

finally:

# ปิดเบราว์เซอร์

```
driver.quit()
```

```
print("Browser closed.")
```

### 3.2 การใช้ Class Name

- ตัวอย่าง: ค้นหา Element ด้วย Class Name.

```
driver.get("https://www.wikipedia.org")
element = driver.find_element("class name", "central-featured")
print("Element Text:", element.text)
```

#### Key Learning:

- เหมาะสำหรับค้นหากลุ่ม Elements.

#### #3.2 การใช้ Class Name

#ตัวอย่าง: ค้นหา Element ด้วย Class Name.

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

# ระบุเส้นทาง ChromeDriver

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

```
driver = webdriver.Chrome(service=service) # สร้าง WebDriver
```

# เปิดเว็บไซต์ Wikipedia

```
driver.get("https://www.wikipedia.org")
```

```
# ค้นหา Element ด้วย Class Name
```

```
element = driver.find_element("class name", "central-featured")
```

```
print("Element Text:", element.text)
```

```
# ปิดเบราว์เซอร์
```

```
driver.quit()
```

### 3.3 การใช้ XPath

- ตัวอย่าง: ค้นหา Search Bar ด้วย XPath.

```
search_bar_xpath = driver.find_element("xpath", "//input[@name='q']")
search_bar_xpath.send_keys("XPath Example")
search_bar_xpath.submit()
```

#ตัวอย่าง: ค้นหา Search Bar ด้วย XPath.

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

# ระบุเส้นทางของ ChromeDriver

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

# เปิดหน้า Wikipedia

```
driver.get("https://www.wikipedia.org")
```

try:

```
# ใช้ Explicit Wait เพื่อรอให้ Search Bar พร้อมใช้งาน
wait = WebDriverWait(driver, 10)
search_bar_xpath = wait.until(EC.presence_of_element_located((By.XPATH,
"//input[@id='searchInput']")))
```

```
# กรอกข้อความใน Search Bar และค้นหา
search_query = "XPath Example"
search_bar_xpath.send_keys(search_query)
search_bar_xpath.submit()
print(f"Searching for: {search_query}")
```

```
# รอให้ผลลัพธ์การค้นหาปรากฏ
result_heading = wait.until(EC.presence_of_element_located((By.ID,
"firstHeading"))))
print(f"Search result heading: {result_heading.text}")
```

except Exception as e:

```
print(f"Error: {e}")
```

finally:

# ปิดเบราว์เซอร์

driver.quit()

Key Learning:

- ใช้ XPath สำหรับโครงสร้างที่ซับซ้อน.
- สามารถกำหนดเส้นทางไปยัง Element ที่เฉพาะเจาะจง.



### 3.4 การใช้ CSS Selector

- ตัวอย่าง: ค้นหา Search Bar ด้วย CSS Selector.

```
search_bar_css = driver.find_element("css selector", "input[name='q']")
search_bar_css.send_keys("CSS Selector Example")
search_bar_css.submit()
```

#### Key Learning:

- CSS Selector เร็วกว่า XPath.
- ใช้เมื่อต้องการค้นหา Elements ที่ซับซ้อน.

### #3.3 การใช้ XPath

#ตัวอย่าง: ค้นหา Search Bar ด้วย XPath.

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

# ระบุเส้นทางของ ChromeDriver

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

# เปิดหน้า Wikipedia

```
driver.get("https://www.wikipedia.org")
```

try:

# ใช้ Explicit Wait เพื่อรอให้ Search Bar พร้อมใช้งาน

wait = WebDriverWait(driver, 10)

search\_bar\_xpath = wait.until(EC.presence\_of\_element\_located((By.XPATH, "  
"//input[@id='searchInput']"))))

# กรอกข้อความใน Search Bar และค้นหา

search\_query = "XPath Example"

search\_bar\_xpath.send\_keys(search\_query)

search\_bar\_xpath.submit()

print(f"Searching for: {search\_query}")

# รอให้ผลลัพธ์การค้นหาปรากฏ

result\_heading = wait.until(EC.presence\_of\_element\_located((By.ID, "  
"firstHeading")))

print(f"Search result heading: {result\_heading.text}")

except Exception as e:

print(f"Error: {e}")

finally:

# ปิดเบราว์เซอร์

driver.quit()

### 3.5 การประยุกต์: ค้นหา Element บน Wikipedia

#### 3.5.1 การใช้ ID

```
driver.get("https://www.wikipedia.org")
search_bar = driver.find_element("id", "searchInput")
search_bar.send_keys("Automated Testing")
search_bar.submit()
print("New Page Title:", driver.title)
driver.quit()
```

#### Key Learning:

- ID เป็น Selectors ที่เร็วและเชื่อถือได้.

### #3.5 การประยุกต์: ค้นหา Element บน Wikipedia

#### #3.5.1 การใช้ ID

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

```
# ตั้งค่า ChromeDriver
```

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

```
ระบุ path ของ ChromeDriver
```

```
driver = webdriver.Chrome(service=service)
```

try:

# เปิดหน้า Wikipedia

driver.get("https://www.wikipedia.org")

# ค้นหา Search Bar ด้วย ID

search\_bar = driver.find\_element(By.ID, "searchInput")

search\_bar.send\_keys("Automated Testing") # กรอกข้อความลงใน Search Bar

search\_bar.submit() # กด Enter เพื่อค้นหา

except Exception as e:

print(f"Error: {e}")

finally:

# ปิดเบราว์เซอร์

driver.quit()

### 3.5.2 การรวม Selectors

- ตัวอย่าง: ใช้ XPath และ CSS Selector ร่วมกัน

```
search_bar_xpath = driver.find_element("xpath", "//input[@id='searchInput']")
search_bar_xpath.send_keys("XPath Combined Example")

search_button_css = driver.find_element("css selector", "button.pure-button")
search_button_css.click()
```

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

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

try:
    # เปิดหน้า Wikipedia
    driver.get("https://www.wikipedia.org")
    print("Opened Wikipedia homepage.")

    # ค้นหา Search Bar ด้วย ID
    search_bar = driver.find_element(By.ID, "searchInput")

    # กรอกคำค้นหาและส่งฟอร์ม
    search_query = "Automated Testing"
    search_bar.send_keys(search_query)
    search_bar.submit()
    print(f"Searched for: {search_query}")

    # แสดงชื่อของหน้าใหม่ (Title)
```

```
print("New Page Title:", driver.title)
```

```
except Exception as e:
```

```
    print(f"Error: {e}")
```

```
finally:
```

```
    # ปิดเบราว์เซอร์
```

```
    driver.quit()
```

```
    print("Browser closed.")
```