QUERY : selenium의 HTML 요소 선택 (Selectors: ID, Class, CSS, XPath) 용도와 예제를

###### **ChatGPt**

Selenium에서는 HTML 요소를 선택하기 위해 여러 가지 방법(ID, Class, CSS Selector, XPath 등)을 제공하며며. 각 선택자의 특징과 사용 예제를 정리

## **1. ID 선택자**

* **설명**: HTML 요소의 id 속성을 사용하여 요소를 찾음
* **장점**: id는 고유하기 때문에 가장 빠르고 정확한 방식
* **메서드**: find\_element(By.ID, "id값")

### **✅ 예제**

from selenium import webdriver

from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get("https://example.com")

# ID로 요소 선택

element = driver.find\_element(By.ID, "login-button")

element.click()

## **2. Class 선택자**

* **설명**: HTML 요소의 class 속성을 사용하여 요소를 찾음
* **주의**: 같은 클래스명을 가진 여러 요소가 있을 수 있음
* **메서드**:
  + 단일 요소: find\_element(By.CLASS\_NAME, "class값")
  + 여러 요소: find\_elements(By.CLASS\_NAME, "class값")

### **✅ 예제**

# 단일 요소 선택

element = driver.find\_element(By.CLASS\_NAME, "btn-primary")

element.click()

# 여러 요소 선택

elements = driver.find\_elements(By.CLASS\_NAME, "menu-item")

for e in elements:

print(e.text)

## **3. CSS 선택자**

* **설명**: CSS 선택자를 사용하여 요소를 찾음 (e.g., .class, #id, tagname)
* **장점**: 복잡한 구조에서도 간결한 코드 가능
* **메서드**:
  + 단일 요소: find\_element(By.CSS\_SELECTOR, "css 선택자")
  + 여러 요소: find\_elements(By.CSS\_SELECTOR, "css 선택자")

### **✅ 예제**

# ID 선택: '#id명'

element = driver.find\_element(By.CSS\_SELECTOR, "#login-button")

# 클래스 선택: '.class명'

element = driver.find\_element(By.CSS\_SELECTOR, ".btn-primary")

# 태그+클래스 조합: 'tagname.class명'

element = driver.find\_element(By.CSS\_SELECTOR, "button.btn-primary")

# 하위 요소 선택: '부모 > 자식'

child\_element = driver.find\_element(By.CSS\_SELECTOR, "div.container > p")

## **4. XPath 선택자**

* **설명**: XML 경로 표현식을 사용하여 요소를 찾음
* **장점**: HTML 구조가 복잡해도 원하는 요소를 정확히 찾을 수 있음
* **메서드**:
  + 단일 요소: find\_element(By.XPATH, "XPath 경로")
  + 여러 요소: find\_elements(By.XPATH, "XPath 경로")

### **✅ 예제**

# 절대 XPath (전체 경로)

element = driver.find\_element(By.XPATH, "/html/body/div/button")

# 상대 XPath (추천)

element = driver.find\_element(By.XPATH, "//button[@id='login-button']")

# 특정 텍스트를 포함하는 요소 선택

element = driver.find\_element(By.XPATH, "//button[contains(text(), '로그인')]")

# 부모 요소에서 자식 요소 찾기

child = driver.find\_element(By.XPATH, "//div[@class='container']/p")

## **5. 각 선택자의 비교**

| **선택자** | **메서드** | **장점** | **단점** |
| --- | --- | --- | --- |
| **ID** | find\_element(By.ID, "id값") | 빠름, 고유 ID 활용 가능 | ID가 없거나 동적으로 변경될 경우 사용 어려움 |
| **Class** | find\_element(By.CLASS\_NAME, "class값") | 동일한 스타일의 요소 선택 가능 | 같은 클래스명이 많으면 특정 요소 찾기 어려움 |
| **CSS Selector** | find\_element(By.CSS\_SELECTOR, "css 선택자") | 간결하고 직관적 | 복잡한 선택자는 가독성이 떨어질 수 있음 |
| **XPath** | find\_element(By.XPATH, "XPath 경로") | 강력한 탐색 기능 제공 | 속도가 느릴 수 있고, 긴 XPath는 유지보수가 어려움 |

## **6. 추천 사용 방법**

* **ID가 있다면 ID 선택자 사용** → 가장 빠름
* **여러 요소를 찾을 경우 CSS Selector 사용** → 간결하고 유지보수 쉬움
* **특정 텍스트를 포함한 요소 찾기나 복잡한 구조 탐색은 XPath 사용**
* **클래스는 같은 이름이 많을 수 있으므로 주의해서 사용**

**✅ 실무에서는 ID, CSS Selector, 그리고 가끔 XPath를 혼합하여 사용**하는 것이 일반적입니다.