

Selenium으로 DOM 요소를 선택하는 방법

DOM 내부에 있는 여러 개의 요소 중 처음 찾아지는 요소를 추출

메서드명	설명
<code>find_element_by_id(id)</code>	id 속성으로 요소를 하나 추출
<code>find_element_by_name(name)</code>	name 속성으로 요소를 하나 추출
<code>find_element_by_css_selector(query)</code>	CSS 선택자로 요소를 하나 추출
<code>find_element_by_xpath(query)</code>	XPath를 지정해 요소를 하나 추출
<code>find_element_by_tag_name(name)</code>	태그 이름이 name에 해당하는 요소를 하나 추출
<code>find_element_by_link_text(text)</code>	링크 텍스트로 요소를 추출
<code>find_element_by_partial_link_text(text)</code>	링크의 자식 요소에 포함돼 있는 텍스트로 요소를 하나 추출
<code>find_element_by_class_name(name)</code>	클래스 이름이 name에 해당하는 요소를 하나 추출

DOM 내부에 있는 모든 요소 추출

메서드명	설명
<code>find_elements_by_css_selector(query)</code>	CSS 선택자로 요소를 여러개 추출
<code>find_elements_by_xpath(query)</code>	XPath를 지정해 요소를 여러개 추출
<code>find_elements_by_tag_name(name)</code>	태그 이름이 name에 해당하는 요소를 여러개 추출
<code>find_elements_by_class_name</code>	클래스 이름이 name에 해당하는 요소를 여러 개 추출
<code>find_elements_by_partial_link_text(text)</code>	링크의 자식 요소에 포함돼 있는 텍스트로 요소를 여러 개 추출

찾지 못하면 NoSuchElementException 예외 발생.

DOM 요소에 적용할 수 있는 메서드와 속성

메서드명	설명
<code>clear()</code>	글자를 입력할 수 있는 요소의 글자를 지움
<code>click()</code>	요소를 클릭
<code>get_attribute(name)</code>	요소의 속성 중 name에 해당하는 속성의 값을 추출
<code>is_displayed()</code>	요소가 화면에 출력되는지 확인
<code>is_enabled()</code>	요소가 활성화돼 있는지 확인
<code>is_selected()</code>	체크박스 등의 요소가 선택된 상태인지 확인
<code>screenshot(filename)</code>	스크린샷을 찍음
<code>send_keys(value)</code>	키를 입력
<code>submit()</code>	입력 양식을 전송
<code>value_of_css_property(name)</code>	name에 해당하는 CSS 속성의 값을 추출
<code>id</code>	요소의 id 속성
<code>location</code>	요소의 위치
<code>parent</code>	부모 요소
<code>rect</code>	크기와 위치 정보를 가진 딕셔너리 자료형을 반환
<code>screenshot_as_base64</code>	스크린샷을 Base64로 추출

size	요소의 크기
tag_name	태그 이름
text	요소 내부의 글자

특수 키를 사용하려면 `from selenium.Webdriver.common.keys import keys` 모듈 import 필요.

Selenium 드라이버 조작 (PhantomJS 전용 드라이버의 메서드와 속성)

메서드명	설명
<code>add_cookie(cookie_dict)</code>	쿠키 값을 딕셔너리 형식으로 지정
<code>back()/forward()</code>	이전 페이지 또는 다음 페이지로 이동
<code>close()</code>	브라우저를 닫음
<code>current_url</code>	현재 URL을 추출
<code>delete_all_cookies()</code>	모든 쿠키를 제거
<code>delete_cookie(name)</code>	특정 쿠키를 제거
<code>execute(command, params)</code>	브라우저 고유의 명령어를 실행
<code>execute_async_script(script, *args)</code>	비동기 처리하는 자바스크립트를 실행
<code>execute_script(script, *args)</code>	동기 처리하는 자바스크립트를 실행
<code>get(url)</code>	웹 페이지를 읽어 들임
<code>get_cookie(name)</code>	특정 쿠키 값을 추출
<code>get_log(type)</code>	로그를 추출(browser/driver/client/server)
<code>get_screenshot_as_base64()</code>	base64 형식으로 스크린샷을 추출
<code>get_screenshot_as_file(filename)</code>	스크린샷을 파일로 저장
<code>get_window_position(windowHandle='current')</code>	브라우저의 위치를 추출
<code>get_window_size(windowHandle='current')</code>	브라우저의 크기를 추출
<code>implicitly_wait(sec)</code>	최대 대기 시간을 초 단위로 지정해서 처리가 끝날 때 까지 대기
<code>quit()</code>	드라이버를 종료시켜 브라우저를 닫음
<code>save_screenshot(filename)</code>	스크린샷을 저장
<code>set_page_load_timeout(time_to_wait)</code>	페이지를 읽는 타임아웃 시간을 지정
<code>set_script_timeout(time_to_wait)</code>	스크립트의 타임아웃 시간을 지정
<code>set_window_position(x, y, windowHandle='current')</code>	브라우저의 위치를 지정

- 참고문헌 : 파이썬을 이용한 머신러닝, 딥러닝 실전 개발 입문
- Selenium with 파이썬 : [URL \(http://selenium-python.readthedocs.io/index.html\)](http://selenium-python.readthedocs.io/index.html)
- SeleniumHQ Documentation : [URL \(http://docs.seleniumhq.org/docs/\)](http://docs.seleniumhq.org/docs/)