Selenium Summary **-By Keshav Potdar**

Level	Category	Command	Description
WebElement	Locator	driver.find_element(**By.ID**, '<id>')	Find WebElement by ID
WebElement	Locator	driver.find_element(**By.NAME**, '<name>')	Find WebElement by Name
WebElement	Locator	driver.find_element(**By.CLASS_NAME**, '<cls>')	Find WebElement by Class
WebElement	Locator	driver.find_element(**By.TAG_NAME**, '<tag>')	Find WebElement by Tag
WebElement	Locator	driver.find_element(**By.LINK_TEXT**, '<text>')	Find WebElement by Link Text
WebElement	Locator	driver.find_element(**By.PARTIAL_LINK_TEXT**, '<txt>')	Find WebElement by Partial Link
WebElement	Locator	driver.find_element(**By.XPATH**, '<//div>')	Find WebElement by XPath
WebElement	Locator	driver.find_element(**By.CSS_SELECTOR**, '<div.cls>')	Find WebElement by CSS
WebElement	Locator	driver.find_elements**(By.XPATH**, '<//div>')	Find multiple WebElements
WebElement	Locator-XPath	//div[@id=**'**<value>**']**	Find WebElement by attribute
WebElement	Locator-XPath	//div**[contains(@class, '**<btn>**')]**	XPath contains() example
WebElement	Locator-XPath	//div**[contains(text(), '**<msg>**')]**	XPath contains() example
WebElement	Locator-XPath	//div**[text()='**<Submit>**']**	XPath by text() example
WebElement	Locator-XPath	//div**[starts-with(@id, '**<You are>**') and @type='**msg**']**	XPath starts-with() example
WebElement	Action	element**.click()**	Click element
WebElement	Action	element**.send_keys('**<keshav>**')**	Enter text
WebElement	Action	element**.clear()**	Clear input
WebElement	Action	element**.submit()**	Submit form
WebElement	Property	element**.text**	Get element text
WebElement	Property	element**.get_attribute('**<val>**')**	Get attribute value
WebElement	Property	element**.get_dom_attribute('**<id>**')**	Get DOM attribute
WebElement	Property	element**.get_property('**<checked>**')**	Get property
WebElement	Property	element**.tag_name**	Get tag name
WebElement	Property	element**.size**	Get size
WebElement	Property	element**.location** element**.size** element**.rect**	1. Returns only position (x, y). Eg. {'x': 50, 'y': 120} 2. Returns only dimensions (height, width). Eg. {'height': 100, 'width': 200} 3. Returns both position, dimensions. Eg. {'height': 100, 'width': 200, 'x': 50, 'y': 120}
WebElement	Property	element**.screenshot('**<elem.png>**')**	Screenshot element
WebElement	State	element**.is_displayed()**	Check if visible
WebElement	State	element**.is_enabled()**	Check if enabled
WebElement	State	element**.is_selected()**	Check if selected
WebElement	ActionChains	**ActionChains(driver).move_to_element(**<WebEle>**).click().perform()**	Hover + click
WebElement	ActionChains	**ActionChains(driver).double_click(**<WebEle>**).perform()**	Double click
WebElement	ActionChains	**ActionChains(driver).context_click(**<WebEle>**).perform()**	Right click
WebElement	ActionChains	**ActionChains(driver).drag_and_drop(**<SrcWebEle>**,** <DstWebEle>**).perform()**	Drag & drop

WebElement	Select	from selenium.webdriver.support.ui import Select
		Select(<WebEle>**).de/select_by_index(**2**)**
			De/Select by index

WebElement	Select	**Select(**<WebEle>**).de/select_by_value('**<India>**')**	De/Select by value
WebElement	Select	**Select(**<WebEle>**).de/select_by_visible_text('**<India>**')**	De/Select by text
WebElement	Select	**Select(**<WebEle>**).deselect_all()**	Deselect all options
WebElement	Select	**Select(**<WebEle>**).options**	List all options
WebElement	Select	**Select(**<WebEle>**).all_selected_options**	List selected options
Exception	Exception	NoSuchElementException	Occurs when element not found. Fix: Use explicit waits or correct locator.
Exception	Exception	TimeoutException	Wait timed out. Fix: Increase wait time, validate locator.
Exception	Exception	ElementNotInteractableException	Element not interactable. Fix: Ensure visible & enabled, use waits.
Exception	Exception	StaleElementReferenceException	Element is no longer attached to DOM. Fix: Re-find element after refresh.
Exception	Exception	NoSuchWindowException	Window not found. Fix: Switch to correct window handle.
Exception	Exception	NoSuchFrameException	Frame not found. Fix: Verify frame locator & switch correctly.
Exception	Exception	ElementClickInterceptedException	Click blocked by another element. Fix: Scroll into view, wait until clickable.
Exception	Exception	InvalidSelectorException	Locator syntax invalid. Fix: Correct XPath/CSS syntax.
Exception	Exception	WebDriverException	Generic driver failure. Fix: Ensure driver/browser compatibility.
Exception	Exception	SessionNotCreatedException	Driver session failed. Fix: Update WebDriver & browser version.

Selenium Summary -By Keshav Potdar

Level	Category	Command	Description
Driver	Browser	driver.**get('https://google.com')**	Open a URL
Driver	Browser	driver.**current_url**	Get current URL
Driver	Browser	driver.**title**	Get page title
Driver	Browser	driver.**page_source**	Get page source HTML
Driver	Browser	driver.**close()**	Close current tab/window
Driver	Browser	driver.**quit()**	Quit browser & end session
Driver	Browser	driver.**refresh()** driver.**forward()** driver.**back()**	Refresh page Navigate forward Navigate back
Driver	Window	driver.**current_window_handle**	Get current window handle
Driver	Window	driver.**window_handles**	Get all window handles
Driver	Window	driver.**switch_to.window(handle)**	Switch to given window/tab
Driver	Window	driver.**switch_to.new_window('tab')** driver.**switch_to.new_window('window')**	Open new tab Open new window
Driver	Window	driver.**fullscreen_window()**	Fullscreen mode
Driver	Window	driver.**maximize_window()**	Maximize window
Driver	Window	driver.**minimize_window()**	Minimize window
Driver	Window	driver.**set_window_size(**1024,768**)**	Set custom size
Driver	Window	driver.**get_window_size()**	Get window size
Driver	Window	driver.**get_window_position()**	Get window position
Driver	Window	driver.**set_window_position(**200,100**)**	Move window position
Driver	Alert	**alert** = driver.**switch_to.alert**	Switch to active alert
Driver	Alert	**alert.accept()**	Accept alert
Driver	Alert	**alert.dismiss()**	Dismiss alert
Driver	Alert	**alert.send_keys('**Keshav**')**	Enter text in prompt
Driver	Alert	**alert.text**	Get alert text
Driver	Frame	driver.**switch_to.frame('frame_name')**	Switch by name
Driver	Frame	driver.**switch_to.frame(0)**	Switch by index
Driver	Frame	driver.**switch_to.frame(**<WebEle>**)**	Switch by WebElement
Driver	Frame	driver.**switch_to.parent_frame()**	Switch to parent
Driver	Frame	driver.**switch_to.default_content()**	Exit to main content
Driver	JavaScript	driver.**execute_script('return document.title;')**	Execute JS sync
Driver	JavaScript	driver.**execute_async_script('callback();')**	Execute JS async
Driver	Screenshot	driver.**save_screenshot('page.png')**	Full page screenshot
Driver	Cookies	driver.**get_cookies()**	Get all cookies
Driver	Cookies	driver.**get_cookie('**name**')**	Get single cookie
Driver	Cookies	driver.**add_cookie(**{'name':'k','value':'v'}**)**	Add cookie
Driver	Cookies	driver.**delete_cookie('**<sess_id>**')**	Delete one cookie
Driver	Cookies	driver.**delete_all_cookies()**	Delete all cookies
Driver	Logging	driver.**get_log("browser")** driver.**get_log("performance")**	browser : Fetches JavaScript console logs (e.g., warnings, errors) performance: Captures browser performance logs (network, rendering, events)
Driver	Tool	**driver.execute_cdp_cmd(**"Page.setDownloadBehavior", {"behavior": "allow", "downloadPath": "/<path>"})	File downloads with DevTools (no prompt)
Driver	Tool	**driver.set_network_conditions(** **offline=**False**,** **latency=**100**,** # milliseconds **download_throughput=**1024**,** # bytes/second **upload_throughput=**1024 # bytes/second **)**	Test slow/unstable networks. set_network_conditions() is wrapper on execute_cdp_cmd() with limited operations.
Driver	Tool	driver**.install_addon("<**/path/to/extension.xpi>**", temporary=<**True>**)**	Install browser add-on (Firefox only)

Driver	Capabilities	from selenium.webdriver.chrome.options import Options
		Options().add_argument('--headless')
			Set Chrome option. Below are few more useful arguments like --headless:
			--**incognito** --**ignore-certificate-errors**
			--**start-maximized** --**disable-popup-blocking**
			--**window-size=**1920,1080 --**mute-audio**
			--**disable-notifications** --**disable-extensions**
			--**disable-infobars** (useul for automation)
			--**kiosk** (full screen mode)

Driver	Capabilities	**options.set_capability("acceptInsecureCerts",** True**)** options.set_capability**("browserName", "**chrome**")** options.set_capability**("browserVersion", "**stable**")** options.set_capability(**"platformName", "**ANY**")** options.set_capability**("acceptSslCerts",** True**)** options.set_capability(**"unhandledPromptBehavior", "**dismiss**")** options.set_capability**("strictFileInteractability",** True**)**	# Accept SSL cert errors # Browser type # Specific version (or "latest"/"stable") # OS platform ("WINDOWS", "LINUX", "MAC", "ANY") # Accept SSL certificates # Auto-dismiss JS alerts # Strict file handling
Driver	Capabilities	**webdriver.Chrome(options=options)**	Launch browser with above passed options
Driver	Timeouts	driver**.set_page_load_timeout(**<sec>**)**	Page load timeout, not applicable for WebEle finding. Applies for driver.get(url), .back() .forward()
Driver	Timeouts	driver**.set_script_timeout(**<sec>**)**	Wait time for JavaScript execution when used driver.execute_async_script(). Not applicable to execute_script() which is synchronous.
Driver	Wait	driver**.implicitly_wait(**<sec>**)**	Set implicit wait. Applicable for all findElement() only
Driver	Wait	from selenium.webdriver.support.ui import WebDriverWait from selenium.webdriver.support import expected_conditions as EC **WebDriverWait(** **driver,** **timeout=**<sec>**,** **poll_frequency=**<max_attempt>**,** **ignored_exceptions=[**NoSuchElementException, ElementNotInteractableException**]** **).until(EC.element_to_be_clickable((By.ID,'**<WebEle>**')))**	Set Explicit / Fluent wait EC.**presence_of_element_located** EC.**visibility_of_element_located** EC.**element_located_to_be_selected** EC.**text_to_be_present_in_element** EC.**frame_to_be_available_and_switch_to_it** EC.**alert_is_present** EC.**invisibility_of_element_located** EC.**number_of_windows_to_be**