Selenium Cheat Sheet with Python

Setup

Install Selenium	`pip install selenium`

WebDriver Setup

Chrome	`webdriver.Chrome(executable_path="path/to/chromedriver")`
Firefox	`webdriver.Firefox(executable_path="path/to/geckodriver")`

Microsoft Edge	`webdriver.Safari(executable_path="path/to/safaridriver")`
Safari (Only Mac)	`webdriver.Edge(executable_path="path/to/msedgedriver")`

PyTest

Install Pytest	`pip install pytest`

Basic Test	`import pytest`
	`def test_example():`
	`  assert 1 + 1 == 2`

Run Tests	`pytest test_file.py`

Run Specific Test	`pytest -k "test_name"`
Run Tests in a Directory	`pytest tests/`

Markers	`@pytest.mark.smoke`
	`def test_smoke_case():`
	`  assert True`

Run Specific Marker	`pytest -m smoke`

Unittest

Import Unittest	`import unittest`

Create a Test Case	`class TestExample(unittest.TestCase):`
	`  def test_addition(self):`
	`  self.assertEqual(1 + 1, 2)`

Run Tests	`python -m unittest test_file.py`

Setup and Teardown	`class TestExample(unittest.TestCase):`
	`  def setUp(self):`
	`  # Setup code pass`
	
	`  def tearDown(self):`
	`  # Teardown code pass`

Run All Tests in a Directory	`python -m unittest discover tests/`

Assert Methods	`assertEqual(a, b)`
	`assertNotEqual(a, b)`
	`assertTrue(x)`
	`assertFalse(x)`
	`assertIs(a, b)`
	`assertIsNot(a, b)`

Locators

By ID	`driver.find_element(By.ID, "id_value")`

By Name	`driver.find_element(By.NAME, "name_value")`
By Class Name	`driver.find_element(By.CLASS_NAME, "class_value")`
By Tag Name	`driver.find_element(By.TAG_NAME, "tag_value")`
By Link Text	`driver.find_element(By.LINK_TEXT, "link_text_value")`
By Partial Link Text	`driver.find_element(By.PARTIAL_LINK_TEXT, "partial_link")`
By CSS Selector	`driver.find_element(By.CSS_SELECTOR, "css_selector")`
By XPath	`driver.find_element(By.XPATH, "xpath_value")`

Interactions

Click an Element	`button = driver.find_element(By.ID, "button_id")` `button.click()`
Input Text	`element = driver.find_element(By.ID, "input_id")` `element.send_keys("Your text")`
Clear Text	`element.clear()`
Submit a Form	`form = driver.find_element(By.ID, "form_id")` `form.submit()`

Waits

Implicit Wait	`driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));`

Explicit Wait	WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10)); WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("elementId")));

Handling Alerts

Switch to Alert	alert = driver.switch_to.alert
Accept Alert	alert.accept()
Dismiss Alert	alert.dismiss();
Get Alert Text	alert_text = alert.text print(alert_text)

Handling Frames

Switch to a Frame	`driver.switch_to.frame("frame_name_or_id")`
Switch to a Frame by Name or ID	`driver.switch_to.frame("frame_name_or_id")`
Switch to a Frame by Index	`driver.switch_to.frame(0)`
Switch to a Frame using a WebElement	`frame_element = driver.find_element(By.XPATH, "//iframe[@id='frame_id']")` `driver.switch_to.frame(frame_element)`
Switch to Parent Frame	`driver.switch_to.parent_frame()`

Switch to Default Content	`driver.switch_to.default_content()`

Handling Dropdowns

Select by Visible Text	`dropdown = Select(driver.find_element(By.ID, "dropdown_id"))` `dropdown.select_by_visible_text("Option Text")`
Select by Index	`dropdown.select_by_index(2)`
Select by Value	`dropdown.select_by_value("option_value")`

Browser Navigation

Navigate to URL	`driver.get("https://example.com")`

Back	`driver.back()`
Forward	`driver.forward()`
Refresh Page	`driver.refresh()`
Close Current Window	`driver.close()`
Quit Entire Session	`driver.quit()`

Browser Window Management

Get the Current	`current_window = driver.current_window_handle`

Window Handle	
Get All Window Handles	`all_windows = driver.window_handles`
Switch to a Specific Window	`driver.switch_to.window(window_handle)`
Switch to the Newly Created Window	`driver.switch_to.window(driver.window_handles[-1])`
Fullscreen Window	`driver.fullscreen_window()`

Minimize Window	`driver.minimize_window()`
Maximize Window	`driver.maximize_window()`
Get Window Size	`size = driver.get_window_size()`
Set Window Position	`driver.set_window_position(100, 200)`

Cookies

Add Cookie	`cookie = {"name": "test_cookie", "value": "cookie_value"} driver.add_cookie(cookie)`

Get All Cookies	`cookies = driver.get_cookies()`
Delete Cookie	`driver.delete_cookie("test_cookie")`
Delete All Cookie	`driver.delete_all_cookies()`

Taking Screenshot

Full Page	`driver.save_screenshot("full_page_screenshot.png")`
WebElement Screenshot	`element = driver.find_element(By.ID, "element_id")` `element.screenshot("element_screenshot.png")`

Working with Files

Upload a File	file_input = driver.find_element(By.ID, "file_upload_id") # Replace with the actual locator
	file_input.send_keys("/path/to/your/file.txt")

Read Data from a Text File	**Read Entire File Content**
	with open("example.txt", "r") as file:
	  content = file.read()
	  print("File Content:")
	  print(content)

	Read File Line by Line
	with open("example.txt", "r") as file:
	  for line in file:
	  print("Line:", line.strip())

	Use File Data in Selenium with open("test_data.txt", "r") as file: data = file.readlines()

Read Data from a CSV File	import csv
	
	# Read the entire CSV file with
	open("example.csv", "r") as file:
	  reader = csv.reader(file)
	  for row in reader:
	  print("Row:", row)

Read Data from an Excel File	**Using openpyxl**
	
	pip install openpyxl
	
	from openpyxl import load_workbook
	# Load the workbook
	workbook = load_workbook("example.xlsx")
	
	# Select the active worksheet
	sheet = workbook.active
	
	# Read all rows
	for row in sheet.iter_rows(values_only=True):
	  print(row)

	Using pandas
	
	pip install pandas
	
	import pandas as pd
	# Load the Excel file
	data = pd.read_excel("example.xlsx")
	
	# Iterate over rows
	for index, row in data.iterrows():
	  print(f"Row {index}: {row}"))

JavaScript Execution

Execute JavaScript	driver.execute_script("JavaScript code here")
Disable a Field (set the 'disabl	driver.execute_script("arguments[0].setAttribute('disabled', 'true')", field)

ed' attribute)	
Enable a Field (remove the 'disabled' attribute)	`driver.execute_script("arguments[0].removeAttribute('disabled')", field)`

Selenium Grid

Start Hub	`java -jar selenium-server.jar hub`
Start Node	`java -jar selenium-server.jar node --hub http://localhost:4444`

Server	http://localhost:4444/ui

Best Practices

- Use explicit waits over implicit waits for better performance.

Handle exceptions to avoid flaky tests:

```
try:
  element = driver.find_element(By.ID, "nonexistent_id")

except Exception as e:
```

- `    print("Error:", e)`
- Always quit the WebDriver:
  `driver.quit()`