Softy EDUCATION

&

Taki Academy
www.takiacademy.com

python™

contact@softyeducation.com

Web Scraping

python™

Content :

01 Introduction to HTML

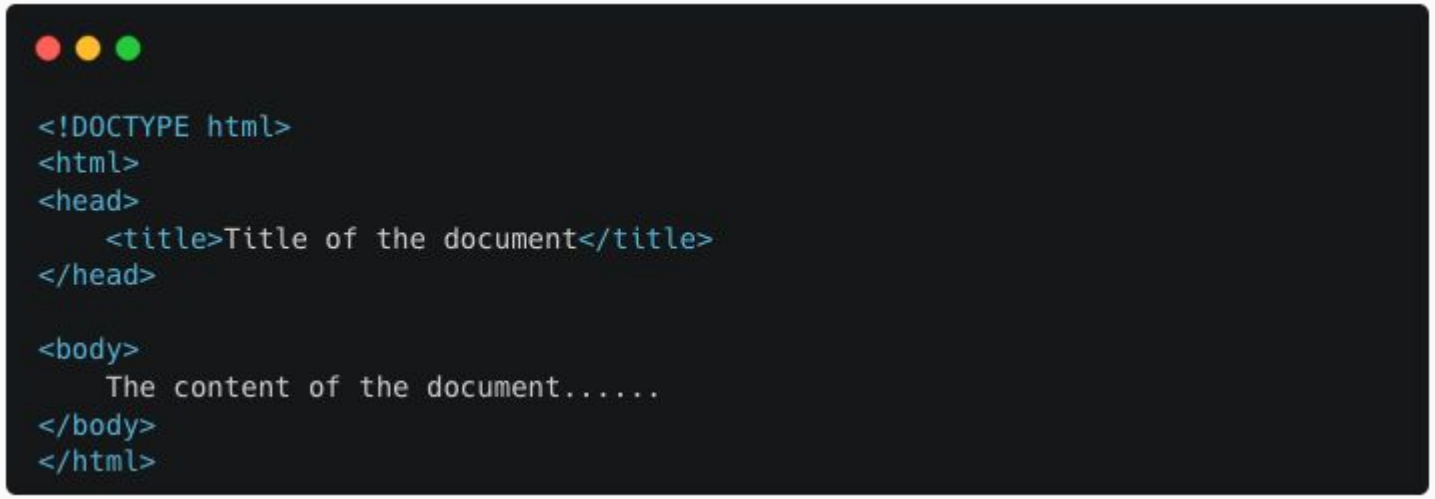
01 - Introduction to HTML

What is HTML?

HTML stands for HyperText Markup Language. It's the standard language for creating web pages. When you visit a website, you're essentially viewing a rendered version of the HTML code of that page. Web browsers like Chrome, Firefox, and Safari interpret this code to display the site's content.

Basic Structure

Every HTML document has a similar basic structure:

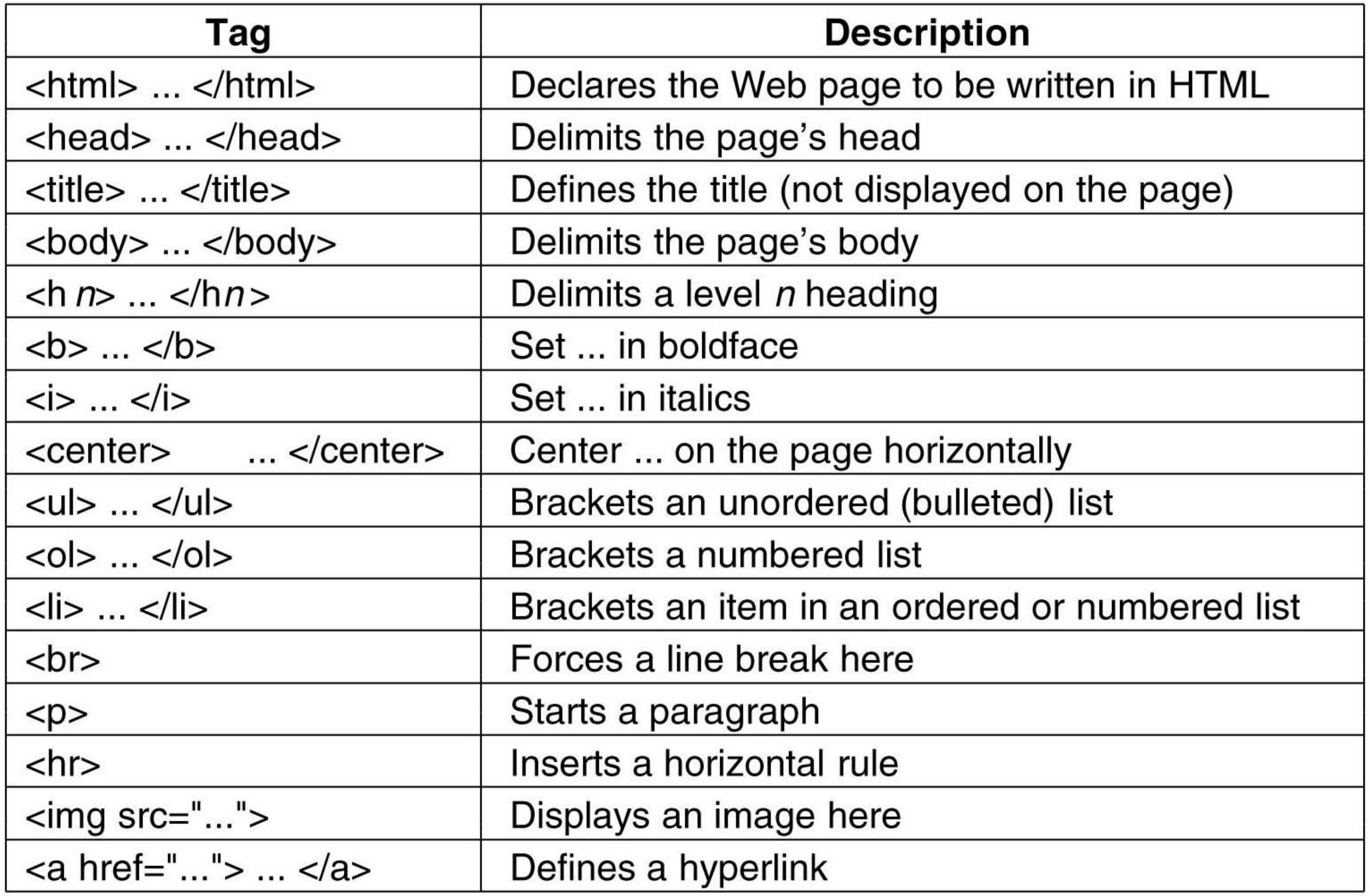


```
<!DOCTYPE html>
<html>
<head>
    <title>Title of the document</title>
</head>

<body>
    The content of the document......
</body>
</html>
```

01 - Introduction to HTML

- **<!DOCTYPE html>**: This declaration defines the document to be HTML5.
- **<html>:** The root element of an HTML page.
- **<head>**: Contains meta-information about the document.
- **<title>**: Specifies a title for the web page, which you'll see on the browser's title bar or tab.
- **<body>**: Contains the visible page content.

01 - Introduction to HTML



Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li> ... </li>	Brackets an item in an ordered or numbered list
<br>	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
<img src="...">	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

Why is HTML Important for Web Scraping?

When you're web scraping, what you're actually doing is parsing the raw HTML of a web page to extract the data you need. Understanding the structure and common patterns in HTML is crucial to being effective at web scraping.

For example: If you know that the data you want is inside a <div> element with a certain class attribute, you can direct your scraping tool to look specifically for that, making your task more straightforward and efficient.

Softy EDUCATION

02 Python Modules

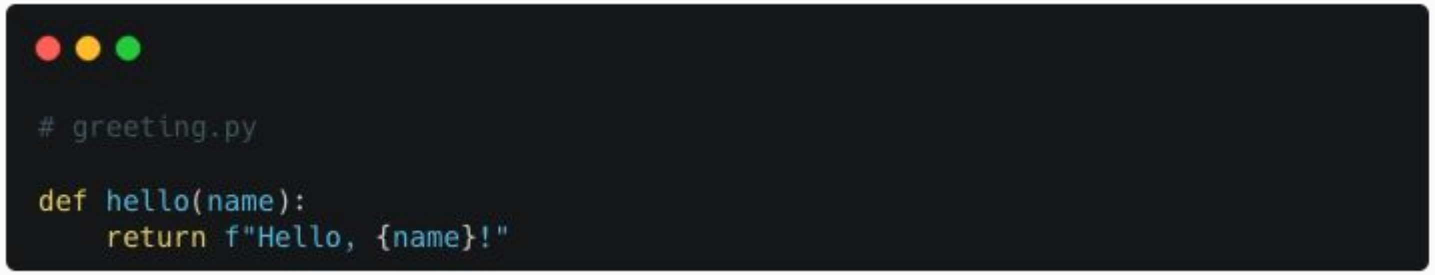
python™

02 - Python Modules

A module in Python is simply a file containing Python definitions and statements. The filename is the module name followed by the **.py** extension. For example, a file named **greeting.py** has a module name of **greeting**.

1. Creating a Basic Module:

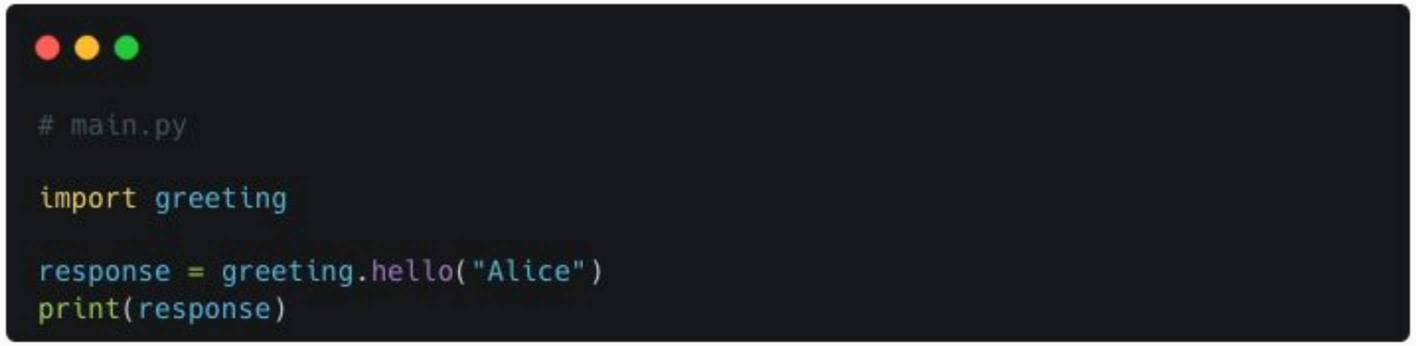
Let's begin by creating a Python file, which we'll treat as a module. Name this file greeting.py.



```python
# greeting.py

def hello(name):
    return f"Hello, {name}!"
```

2. Importing a Module:

Now, let's assume you have another file named **main.py** in the same directory as **greeting.py**. Here's how you can import and use the hello function from the greeting module:

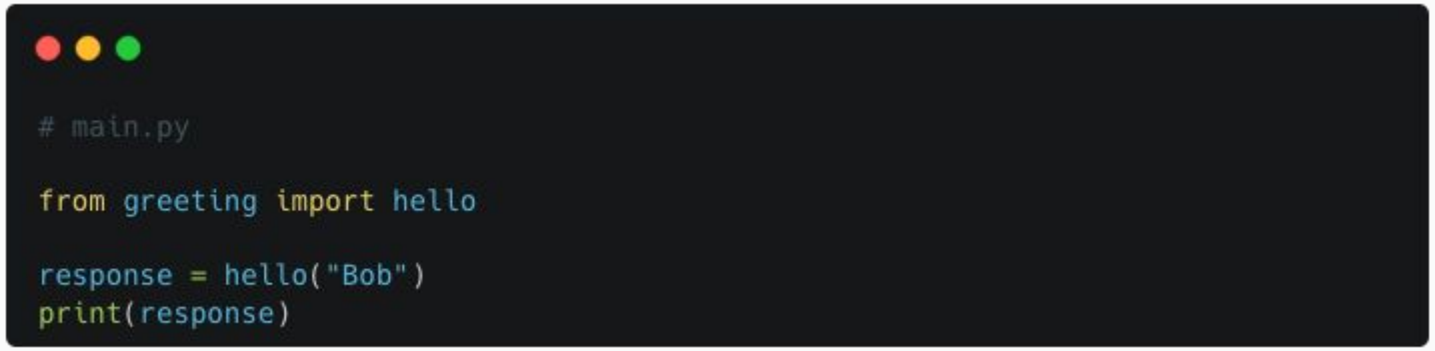


```
# main.py

import greeting

response = greeting.hello("Alice")
print(response)
```

02 - Python Modules

3. Importing a Specific Function:

Instead of importing the entire module, you can just import the specific function:



```python
# main.py

from greeting import hello

response = hello("Bob")
print(response)
```

Softy EDUCATION

03

Introduction to Web Scraping

python™

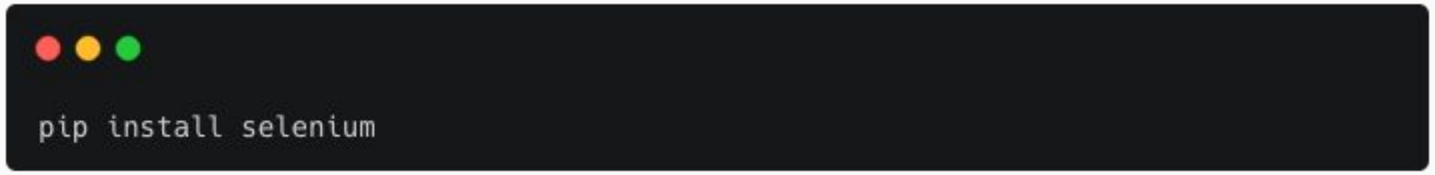
Introduction to Python and Selenium

Web scraping is a method used to extract data from websites. Python, with its rich ecosystem, offers several libraries to facilitate this, one of which is Selenium. While primarily known for automating web applications for testing purposes, Selenium can be used effectively for web scraping tasks.

03 - Introduction to Web Scraping

Prerequisites

1. **Python**: Ensure Python is installed.
2. **Selenium**: Install Selenium using pip:



```
pip install selenium
```

3. Web driver: Selenium requires a web driver to interface with a browser. Download the web driver for your preferred browser (Chrome, Firefox, etc.) and add its location to your system's PATH.

3. Web driver: Selenium requires a web driver to interface with a browser. Download the web driver for your preferred browser (Chrome, Firefox, etc.) and add its location to your system’s PATH.

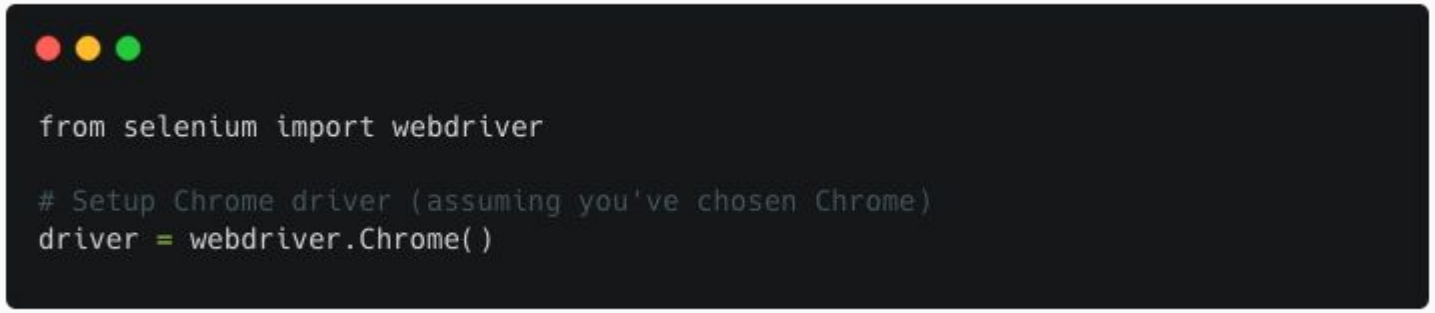
Softy EDUCATION

Basic Web Scraping using Selenium

python™

1. Initialization:

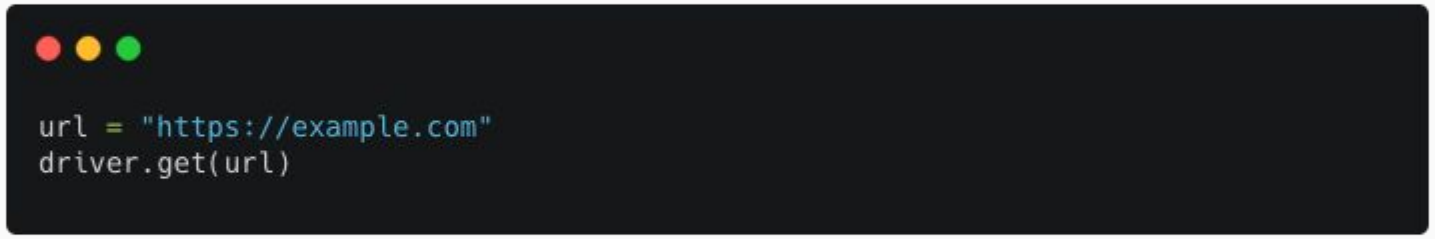
First, set up Selenium to use the desired browser.



```python
from selenium import webdriver

# Setup Chrome driver (assuming you've chosen Chrome)
driver = webdriver.Chrome()
```

2. Navigating to a Webpage:

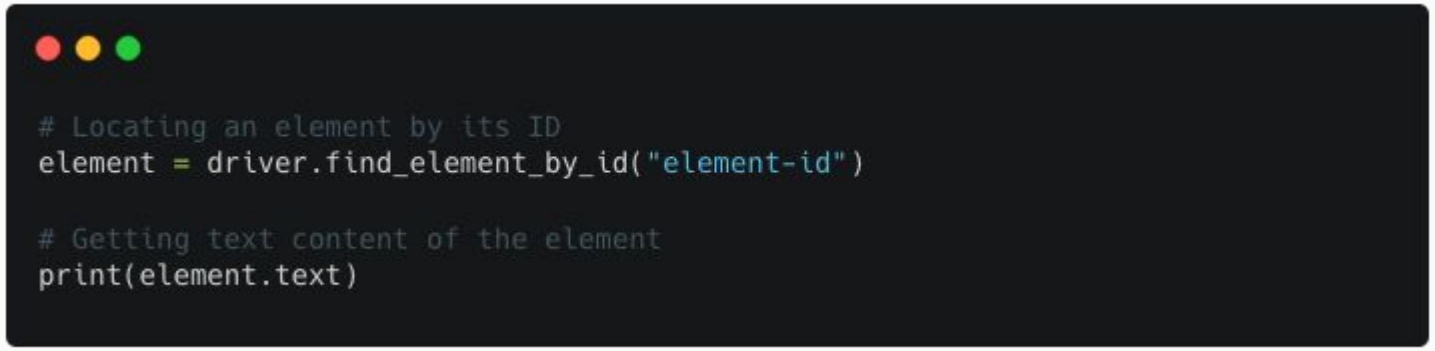
Use the .get() method to navigate to a URL.



```
url = "https://example.com"
driver.get(url)
```

3. Extracting Information:

Locate elements using various methods like find_element_by_id, find_element_by_name, etc. For multiple elements, methods like find_elements_by_class_name can be used.

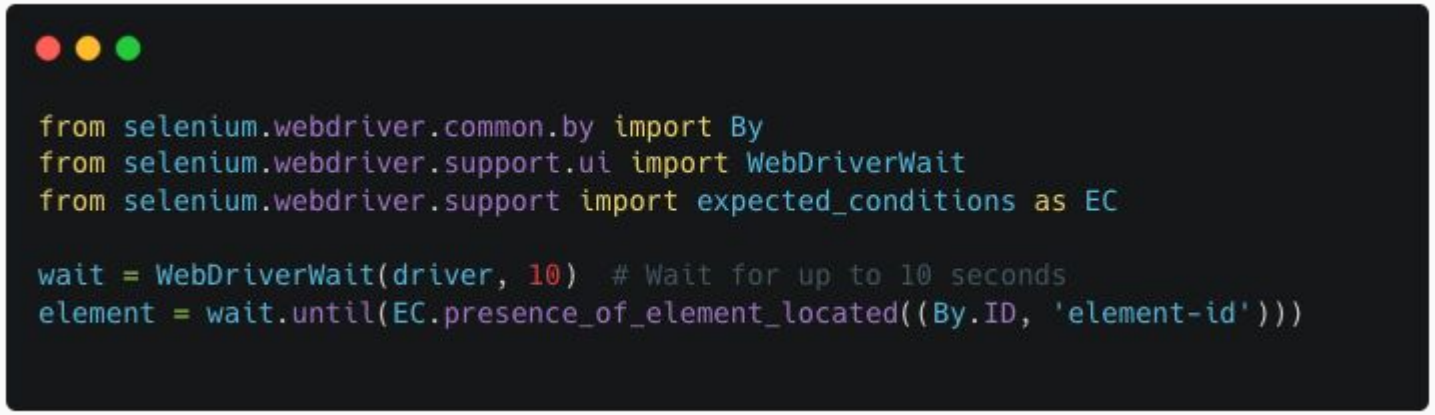


```
# Locating an element by its ID
element = driver.find_element_by_id("element-id")

# Getting text content of the element
print(element.text)
```

Handling Dynamic Content

1. Waiting for Elements:

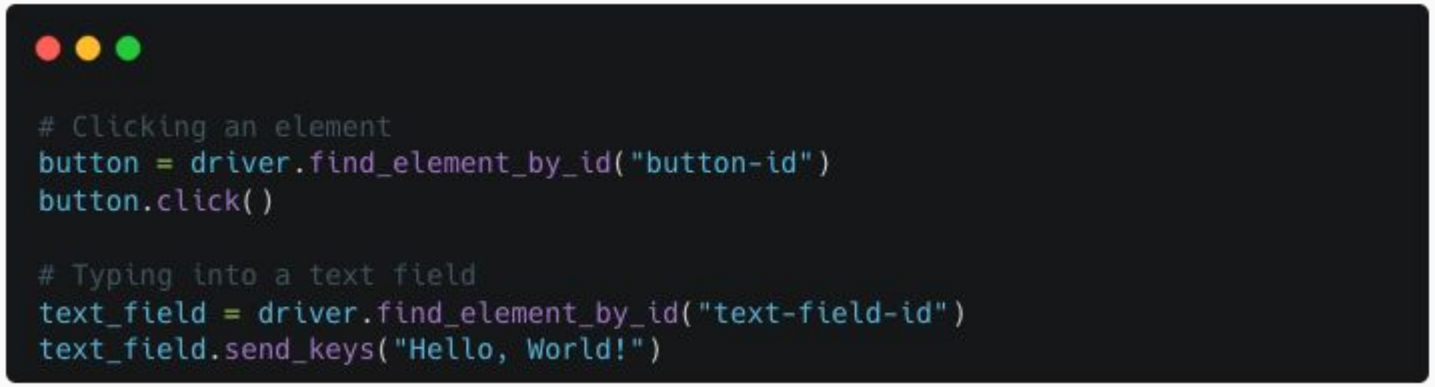
Sometimes, you need to wait for elements to load. Selenium offers explicit waits for this.



```python
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

wait = WebDriverWait(driver, 10)  # Wait for up to 10 seconds
element = wait.until(EC.presence_of_element_located((By.ID, 'element-id')))
```

2. Interacting with the Page:

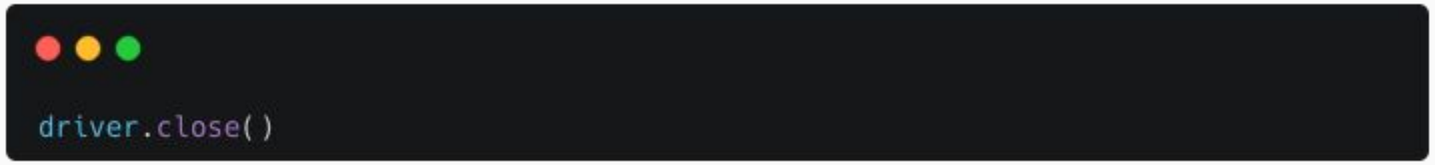
You can simulate interactions like clicks, input text, etc.



```python
# Clicking an element
button = driver.find_element_by_id("button-id")
button.click()

# Typing into a text field
text_field = driver.find_element_by_id("text-field-id")
text_field.send_keys("Hello, World!")
```

Closing the Browser

Always remember to close the browser once your tasks are done.



```
driver.close()
```

Thank you