# 选择元素的基本方法

### 1. 通过 ID 选择

使用元素的 id 属性，是最准确的选择方式。

element = driver.find\_element(By.ID, 'element\_id')

### 2. 通过 Class Name 选择

使用元素的 class 属性，适用于批量选择。

# 单个元素  
element = driver.find\_element(By.CLASS\_NAME, 'class\_name')  
# 多个元素（返回列表）  
elements = driver.find\_elements(By.CLASS\_NAME, 'class\_name')

### 3. 通过 Name 选择

使用元素的 name 属性，常见于表单元素。

element = driver.find\_element(By.NAME, 'input\_name')

### 4. 通过 Tag Name 选择

使用 HTML 标签名，如 div, input, a 等。

elements = driver.find\_elements(By.TAG\_NAME, 'div')

### 5. 通过 Link Text 选择

仅适用于链接（<a> 标签），使用完整文本内容。

element = driver.find\_element(By.LINK\_TEXT, '点击这里')

### 6. 通过 Partial Link Text 选择

仅适用于链接，使用部分文本内容（模糊匹配）。

element = driver.find\_element(By.PARTIAL\_LINK\_TEXT, '点击')

### 7. 通过 CSS Selector 选择

使用 CSS 选择器语法，功能强大且灵活。

# 示例：选择 class 为 "container" 下的所有 input 元素  
elements = driver.find\_elements(By.CSS\_SELECTOR, '.container input')

### 8. 通过 XPath 选择

使用 XPath 表达式，支持复杂的层级和条件选择。

# 绝对路径（不推荐）  
element = driver.find\_element(By.XPATH, '/html/body/div[1]/input')  
# 相对路径（推荐）  
element = driver.find\_element(By.XPATH, '//input[@name="username"]')

### 注意事项：

1. **导入必要的模块**：

from selenium.webdriver.common.by import By

1. **元素不存在的处理**：

* 1. find\_element() 找不到元素时会抛出 NoSuchElementException。

* 1. find\_elements() 找不到元素时返回空列表 []。