# css selector选择元素的基本方法

# 1. 元素选择器

通过标签名直接选择元素。

# 选择所有 <div> 元素  
divs = driver.find\_elements(By.CSS\_SELECTOR, 'div')

# 2. ID 选择器

通过元素的 id 属性选择，使用 #id 语法。

# 选择 id 为 "username" 的元素  
element = driver.find\_element(By.CSS\_SELECTOR, '#username')

# 3. 类选择器

通过元素的 class 属性选择，使用 .class 语法。

# 选择 class 为 "btn" 的元素  
buttons = driver.find\_elements(By.CSS\_SELECTOR, '.btn')  
  
# 选择同时包含 "btn" 和 "primary" 类的元素  
primary\_button = driver.find\_element(By.CSS\_SELECTOR, '.btn.primary')

# 4. 属性选择器

通过元素的任意属性选择，使用 [attribute=value] 语法。

# 选择所有 name="email" 的元素  
email\_input = driver.find\_element(By.CSS\_SELECTOR, '[name="email"]')  
  
# 选择所有 type 为 "submit" 的按钮  
submit\_buttons = driver.find\_elements(By.CSS\_SELECTOR, '[type="submit"]')  
  
# 选择所有包含 disabled 属性的元素（不指定值）  
disabled\_elements = driver.find\_elements(By.CSS\_SELECTOR, '[disabled]')

# 5. 层级选择器

通过元素的层级关系选择。

## 子元素选择器（直接子元素）

# 选择 class 为 "parent" 的元素下的直接子 <div>  
child\_divs = driver.find\_elements(By.CSS\_SELECTOR, '.parent > div')

## 后代元素选择器（所有层级的子元素）

# 选择 class 为 "container" 的元素内的所有 <a> 标签  
links = driver.find\_elements(By.CSS\_SELECTOR, '.container a')

# 6. 组合选择器

将多种选择器组合使用。

# 选择 id 为 "form" 内的所有 input 标签，且 type 为 "text"  
text\_inputs = driver.find\_elements(By.CSS\_SELECTOR, '#form input[type="text"]')

# 7. 伪类选择器

通过元素的状态或位置选择。

# 选择第一个 <li> 元素  
first\_li = driver.find\_element(By.CSS\_SELECTOR, 'li:first-child')  
  
# 选择最后一个 <li> 元素  
last\_li = driver.find\_element(By.CSS\_SELECTOR, 'li:last-child')  
  
# 选择偶数位置的 <tr> 元素  
even\_rows = driver.find\_elements(By.CSS\_SELECTOR, 'tr:nth-child(even)')  
  
# 选择当前聚焦的元素  
focused\_element = driver.find\_element(By.CSS\_SELECTOR, ':focus')

# 8. 相邻兄弟选择器

选择紧接在另一个元素后的兄弟元素。

# 选择 <h2> 后面的第一个 <p> 元素  
next\_p = driver.find\_element(By.CSS\_SELECTOR, 'h2 + p')

# 9. 通用兄弟选择器

选择同一父元素下的所有后续兄弟元素。

# 选择 <div> 后面的所有 <span> 元素  
all\_spans = driver.find\_elements(By.CSS\_SELECTOR, 'div ~ span')

# 10. 属性值匹配模式

更灵活地匹配属性值。

## 包含某个字符串（\*=）

# 选择 href 包含 "example" 的链接  
links = driver.find\_elements(By.CSS\_SELECTOR, 'a[href\*="example"]')

## 以某个字符串开头（^=）

# 选择 src 以 "https" 开头的图片  
images = driver.find\_elements(By.CSS\_SELECTOR, 'img[src^="https"]')

## 以某个字符串结尾（$=）

# 选择所有 .pdf 文件链接  
pdf\_links = driver.find\_elements(By.CSS\_SELECTOR, 'a[href$=".pdf"]')

# 对比 XPath 和 CSS Selector：

|  |  |  |
| --- | --- | --- |
| **场景** | **CSS Selector** | **XPath** |
| 按 ID 选择 | #id | //\*[@id="id"] |
| 按类选择 | .class | //\*[@class="class"] |
| 选择父元素的子元素 | parent > child | //parent/child |
| 模糊匹配属性值 | [attr\*="value"] | //\*[contains(@attr, "value")] |

CSS Selector 的语法更简洁，性能通常优于 XPath，尤其在处理复杂层级时更具优势。建议优先使用 CSS Selector 进行元素定位。