

THAM KHẢO NHANH XPATH

Axes, predicates, hàm, toán tử, chọn nodes

Cú pháp

Biểu thức đường dẫn

<code>/</code>	Node gốc (bắt đầu đường dẫn tuyệt đối)
<code>/bookstore/book</code>	Chọn con trực tiếp
<code>//book</code>	Chọn tất cả nodes book ở bất kỳ đâu
<code>.</code>	Node ngữ cảnh hiện tại
<code>..</code>	Cha của node hiện tại
<code>@lang</code>	Thuộc tính tên lang
<code>node()</code>	Bất kỳ node nào thuộc mọi kiểu
<code>*</code>	Bất kỳ node phân tử nào
<code>@*</code>	Bất kỳ thuộc tính nào

Ví dụ cơ bản

```
//html/body/div # absolute path to <div>
//input[@type='text'] # all text inputs
//div[@class='main']/* # children of div.main
//a/@href # all href attributes
```

Kết hợp đường dẫn

```
//book/title | //book/price # union of two paths
//h1 | //h2 | //h3 # multiple element types
```

Axes

Hướng Axes

<code>child::</code>	Con trực tiếp (axis mặc định)
<code>parent::</code>	Cha trực tiếp
<code>ancestor::</code>	Tất cả tổ tiên đến gốc
<code>ancestor-or-self::</code>	Tổ tiên + node hiện tại
<code>descendant::</code>	Tất cả con cháu
<code>descendant-or-self::</code>	Con cháu + node hiện tại
<code>following::</code>	Tất cả nodes sau node hiện tại trong tài liệu
<code>following-sibling::</code>	Anh em phía sau node hiện tại
<code>preceding::</code>	Tất cả nodes trước node hiện tại trong tài liệu
<code>preceding-sibling::</code>	Anh em phía trước node hiện tại
<code>self::</code>	Chỉ node hiện tại
<code>attribute::</code>	Thuộc tính của node hiện tại
<code>namespace::</code>	Nodes namespace

Ví dụ Axes

```
//div/child::p # <p> children of <div>
//td/parent::tr # <tr> parent of <td>
//h2/following-sibling::p # <p> after <h2>
//li/ancestor::ul # <ul> containing <li>
```

Predicates

Lọc bằng Predicates

```
//book[1] # first book element
//book[last()] # last book element
//book[position() < 3] # first two books
//book[@lang='en'] # books with lang="en"
//book[price > 30] # books with price > 30
```

Các mẫu Predicate

<code>[n]</code>	Phần tử ở vị trí n (bắt đầu từ 1)
<code>[last()]</code>	Phần tử cuối
<code>[last()-1]</code>	Thứ hai từ cuối
<code>[@attr]</code>	Có thuộc tính
<code>[@attr='val']</code>	Thuộc tính bằng giá trị
<code>[element]</code>	Có phần tử con
<code>[element='text']</code>	Phần tử con chứa văn bản
<code>[not(@attr)]</code>	Không có thuộc tính

Predicates chuỗi

```
//div[@class='list']/a[1] # first <a> in div.list
//input[@type='text'][@name='q'] # AND condition
//book[price>30][@lang='en'] # multiple conditions
```

Hàm

Hàm String

<code>contains(s, sub)</code>	True nếu s chứa sub
<code>starts-with(s, pre)</code>	True nếu s bắt đầu bằng pre
<code>string-length(s)</code>	Độ dài chuỗi
<code>normalize-space(s)</code>	Xóa khoảng trắng đầu cuối và nén khoảng trắng nội bộ
<code>concat(a, b, ...)</code>	Nối các chuỗi
<code>substring(s, pos, len)</code>	Trích chuỗi con (bắt đầu từ 1)
<code>translate(s, from, to)</code>	Thay thế từng ký tự

Hàm số

<code>sum(node-set)</code>	Tổng các giá trị số
<code>count(node-set)</code>	Số lượng nodes
<code>floor(n)</code>	Làm tròn xuống
<code>ceiling(n)</code>	Làm tròn lên
<code>round(n)</code>	Làm tròn đến số nguyên gần nhất
<code>number(val)</code>	Chuyển đổi sang số

Ví dụ hàm

```
//div[contains(@class, 'active')]
//a[starts-with(@href, 'https')]
//p[string-length(text()) > 100]
//ul[count(li) > 5]
```

Toán tử

Toán tử so sánh

<code>=</code>	Bằng
<code>!=</code>	Không bằng
<code><</code>	Nhỏ hơn
<code><=</code>	Nhỏ hơn hoặc bằng
<code>></code>	Lớn hơn
<code>>=</code>	Lớn hơn hoặc bằng

Logic & Số học

<code>and</code>	Logic AND
------------------	-----------

<code>or</code>	Logic OR
<code>not()</code>	Logic NOT (hàm)
<code>+</code>	Cộng
<code>-</code>	Trừ
<code>*</code>	Nhân
<code>div</code>	Chia
<code>mod</code>	Lấy dư
<code> </code>	Hợp của tập node

Ví dụ toán tử

```
//book[price > 20 and price < 50]
//item[etype='a' or etype='b']
//span[not(contains(@class, 'hidden'))]
```

Node Tests

Các loại Node

<code>node()</code>	Bất kỳ node nào (phần tử, văn bản, chú thích, PI)
<code>text()</code>	Chỉ node văn bản
<code>comment()</code>	Chỉ node chú thích
<code>processing-instruction()</code>	Node chỉ thị xử lý
<code>*</code>	Bất kỳ node phân tử nào
<code>@*</code>	Bất kỳ node thuộc tính nào
<code>element-name</code>	Phần tử có tên cụ thể

Ví dụ Node Test

```
//p/text() # text content of <p>
//div/comment() # comments inside <div>
//body/node() # all child nodes of <body>
//div/* # all element children of <div>
```

Hàm Boolean

<code>true()</code>	Boolean true
<code>false()</code>	Boolean false
<code>boolean(expr)</code>	Chuyển đổi sang boolean
<code>not(expr)</code>	Đảo ngược boolean
<code>lang(code)</code>	True nếu ngôn ngữ của node khớp

Viết tắt

Dạng ngắn vs Dạng đầy đủ

<code>(không có)</code>	child: (axis mặc định)
<code>@</code>	attribute::
<code>//</code>	/descendant-or-self::node()/
<code>.</code>	self::node()
<code>..</code>	parent::node()
<code>[n]</code>	[position]=n

Ví dụ viết tắt

```
# These pairs are equivalent:
child:div → div
attribute::href → @href
/descendant-or-self::node()/p → //p
self::node() → .
parent::node() → ..
```

Các mẫu viết tắt phổ biến

```
//div[@id='main'] # div with id="main"
//table/td # all <td> in any <table>
../sibling # sibling via parent
../span # span descendants of context
```

Các mẫu thường dùng

Web Scraping / Testing

```
//input[@name='username'] # form input by name
//button[text()='Submit'] # button by text
//div[contains(@class, 'error')] # element by partial class
//a[contains(@href, 'login')] # link by partial href
```

Chọn có điều kiện

```
//div[@class='item'][../span[@class='price']]
//tr[td[1]='Active'] # row where 1st cell = Active
//*[contains(text(), 'Warning')] # any element with text
```

XPath trong Python (lxml)

```
from lxml import html
tree = html.fromstring(page_content)
links = tree.xpath('//a/@href')
titles = tree.xpath('//h2/text()')
```

XPath trong Selenium

```
driver.find_element(By.XPATH, "//input[@id='search']")
driver.find_elements(By.XPATH, "//li[@class='result']")
```