

REFERÊNCIA RÁPIDA DE XPATH

Eixos, predicados, funções, operadores, seleção de nós

Sintaxe

Expressões de Caminho

<code>/</code>	Nó raiz (início de caminho absoluto)
<code>/bookstore/book</code>	Seleção de filho direto
<code>//book</code>	Selecionar todos os nós book em qualquer lugar
<code>.</code>	Nó de contexto atual
<code>..</code>	Pai do nó atual
<code>@lang</code>	Atributo chamado lang
<code>node()</code>	Qualquer nó de qualquer tipo
<code>*</code>	Qualquer nó elemento
<code>@*</code>	Qualquer atributo

Exemplos Básicos

```
/html/body/div # caminho absoluto para <div>
//input[@type='text'] # todos os inputs de texto
//div[@class='main']/* # filhos de div.main
//a/@href # todos os atributos href
```

Combinando Caminhos

```
//book/title | //book/price # união de dois caminhos
//h1 | //h2 | //h3 # múltiplos tipos de elemento
```

Eixos

Direções de Eixo

<code>child::</code>	Filhos diretos (eixo padrão)
<code>parent::</code>	Pai direto
<code>ancestor::</code>	Todos os ancestrais até a raiz
<code>ancestor-or-self::</code>	Todos os ancestrais + nó atual
<code>descendant::</code>	Todos os descendentes
<code>descendant-or-self::</code>	Descendentes + nó atual
<code>following::</code>	Todos os nós após o atual no documento
<code>following-sibling::</code>	Irmãos após o atual
<code>preceding::</code>	Todos os nós antes do atual no documento
<code>preceding-sibling::</code>	Irmãos antes do atual
<code>self::</code>	Apenas o nó atual
<code>attribute::</code>	Atributos do nó atual
<code>namespace::</code>	Nós de namespace

Exemplos de Eixo

```
//div/child::p # filhos <p> de <div>
//td/parent::tr # pai <tr> de <td>
//h2/following-sibling::p # <p> após <h2>
//li/ancestor::ul # <ul> contendo <li>
```

Predicados

Filtrando com Predicados

```
//book[1] # primeiro elemento book
//book[last()] # último elemento book
//book[position() < 3] # primeiros dois books
//book[@lang='en'] # books com lang='en'
//book[price > 30] # books com preço > 30
```

Padrões de Predicado

<code>[n]</code>	Elemento na posição n (base 1)
<code>[last()]</code>	Último elemento
<code>[last()-1]</code>	Penúltimo
<code>[@att]</code>	Tem atributo
<code>[@att='val']</code>	Atributo igual ao valor
<code>[element]</code>	Tem elemento filho
<code>[element='text']</code>	Elemento filho contém texto
<code>[not(@att)]</code>	Não tem atributo

Predicados Encadeados

```
//div[@class='list']/a[1] # primeiro <a> em div.list
//input[@type='text'][@name='q'] # condição AND
//book[price>30][@lang='en'] # múltiplas condições
```

Funções

Funções de String

<code>contains(s, sub)</code>	Verdadeiro se s contém sub
<code>starts-with(s, pre)</code>	Verdadeiro se s começa com pre
<code>string-length(s)</code>	Comprimento da string
<code>normalize-space(s)</code>	Remover espaços extras e colapsar
<code>concat(a, b, ...)</code>	Concatenar strings
<code>substring(s, pos, len)</code>	Extrair substring (base 1)
<code>translate(s, from, to)</code>	Substituição caractere por caractere

Funções Numéricas

<code>sum(node-set)</code>	Soma de valores numéricos
<code>count(node-set)</code>	Número de nós
<code>floor(n)</code>	Arredondar para baixo
<code>ceiling(n)</code>	Arredondar para cima
<code>round(n)</code>	Arredondar para o inteiro mais próximo
<code>number(val)</code>	Converter para número

Exemplos de Funções

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

Operadores

Operadores de Comparação

<code>=</code>	Igual
<code>!=</code>	Diferente
<code><</code>	Menor que
<code><=</code>	Menor ou igual
<code>></code>	Maior que
<code>>=</code>	Maior ou igual

Lógicos e Aritméticos

<code>and</code>	AND lógico
<code>or</code>	OR lógico
<code>not()</code>	NOT lógico (função)

<code>+</code>	Adição
<code>-</code>	Subtração
<code>*</code>	Multiplicação
<code>div</code>	Divisão
<code>mod</code>	Módulo
<code> </code>	União de conjuntos de nós

Exemplos de Operadores

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

Testes de Nó

Tipos de Nó

<code>node()</code>	Qualquer nó (elemento, texto, comentário, PI)
<code>text()</code>	Apenas nó de texto
<code>comment()</code>	Apenas nó de comentário
<code>processing-instruction()</code>	Nó de instrução de processamento
<code>*</code>	Qualquer nó elemento
<code>@*</code>	Qualquer nó atributo
<code>element-name</code>	Elemento com nome específico

Exemplos de Teste de Nó

```
//p/text() # conteúdo de texto de <p>
//div/comment() # comentários dentro de <div>
//body/node() # todos os nós filhos de <body>
//div/* # todos os filhos elemento de <div>
```

Funções Booleanas

<code>true()</code>	Booleano verdadeiro
<code>false()</code>	Booleano falso
<code>boolean(expr)</code>	Converter para booleano
<code>not(expr)</code>	Negar booleano
<code>lang(code)</code>	Verdadeiro se o idioma do nó corresponde

Abreviações

Forma Curta vs Longa

<code>(nenhuma)</code>	child:: (eixo padrão)
<code>@</code>	attribute::
<code>//</code>	/descendant-or-self::node()/
<code>.</code>	self::node()
<code>..</code>	parent::node()
<code>[n]</code>	[position]=n

Exemplos Abreviados

```
# Estes pares são equivalentes:
child::div -> div
attribute::href -> @href
/descendant-or-self::node()/p -> //p
self::node() -> .
parent::node() -> ..
```

Padrões Abreviados Commons

```
//div[@id='main'] # div com id="main"
//table/td # todos os <td> em qualquer <table>
../sibling # irmão via pai
../span # descendentes span do contexto
```

Padrões Comuns

Web Scraping / Testes

```
//input[@name='username'] # input de formulário por nome
//button[text()='Submit'] # botão por texto
//div[contains(@class, 'error')] # elemento por classe parcial
//a[contains(@href, 'login')] # link por href parcial
```

Seleção Condicional

```
//div[@class='item'][../span[@class='price']]
//tr[td[1]='Active'] # linha onde 1ª célula = Active
//*[contains(text(), 'Warning')] # qualquer elemento com texto
```

XPath em Python (lxml)

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

XPath no Selenium

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