

REFERENCIA RÁPIDA DE XPATH

Ejes, predicados, funciones, operadores, selección de nodos

Sintaxis

Expresiones de Ruta

<code>/</code>	Nodo raíz (inicio de ruta absoluta)
<code>/bookstore/book</code>	Selección de hijo directo
<code>//book</code>	Seleccionar todos los nodos book en cualquier parte
<code>.</code>	Nodo de contexto actual
<code>..</code>	Padre del nodo actual
<code>@lang</code>	Atributo llamado lang
<code>node()</code>	Cualquier nodo de cualquier tipo
<code>*</code>	Cualquier nodo elemento
<code>@*</code>	Cualquier atributo

Ejemplos Básicos

```
//html/body/div # ruta absoluta a <div>
//input[@type='text'] # todos los inputs de texto
//div[@class='main']/* # hijos de div.main
//a/@href # todos los atributos href
```

Combinar Rutas

```
//book/title | //book/price # unión de dos rutas
//h1 | //h2 | //h3 # múltiples tipos de elemento
```

Ejes

Direcciones de Eje

<code>child::</code>	Hijos directos (eje predeterminado)
<code>parent::</code>	Padre directo
<code>ancestor::</code>	Todos los ancestros hasta la raíz
<code>ancestor-or-self::</code>	Ancestros + nodo actual
<code>descendant::</code>	Todos los descendientes
<code>descendant-or-self::</code>	Descendientes + nodo actual
<code>following::</code>	Todos los nodos después del actual en el documento
<code>following-sibling::</code>	Hermanos después del actual
<code>preceding::</code>	Todos los nodos antes del actual en el documento
<code>preceding-sibling::</code>	Hermanos antes del actual
<code>self::</code>	Solo el nodo actual
<code>attribute::</code>	Atributos del nodo actual
<code>namespace::</code>	Nodos de espacio de nombres

Ejemplos de Eje

```
//div/child::p # hijos <p> de <div>
//td/parent::tr # padre <tr> de <td>
//h2/following-sibling::p # <p> después de <h2>
//li/ancestor::ul # <ul> que contiene <li>
```

Predicados

Filtrar con Predicados

```
//book[1] # primer elemento book
//book[last()] # último elemento book
//book[position() < 3] # primeros dos books
//book[@lang='en'] # books con lang="en"
//book[price > 30] # books con precio > 30
```

Patrones de Predicado

<code>[n]</code>	Elemento en posición n (base 1)
<code>[last()]</code>	Último elemento
<code>[last()-1]</code>	Penúltimo
<code>[@attr]</code>	Tiene atributo
<code>[@attr='val']</code>	Atributo igual a valor
<code>[element]</code>	Tiene elemento hijo
<code>[element='text']</code>	Elemento hijo contiene texto
<code>[not(@attr)]</code>	No tiene atributo

Predicados Encadenados

```
//div[@class='list']/a[1] # primer <a> en div.list
//input[@type='text'][@name='q'] # condición AND
//book[price>30][@lang='en'] # múltiples condiciones
```

Funciones

Funciones de Cadena

<code>contains(s, sub)</code>	Verdadero si s contiene sub
<code>starts-with(s, pre)</code>	Verdadero si s comienza con pre
<code>string-length(s)</code>	Longitud de la cadena
<code>normalize-space(s)</code>	Recortar y colapsar espacios en blanco
<code>concat(a, b, ...)</code>	Unir cadenas
<code>substring(s, pos, len)</code>	Extraer subcadena (base 1)
<code>translate(s, from, to)</code>	Reemplazo carácter por carácter

Funciones Numéricas

<code>sum(node-set)</code>	Suma de valores numéricos
<code>count(node-set)</code>	Número de nodos
<code>floor(n)</code>	Redondear hacia abajo
<code>ceiling(n)</code>	Redondear hacia arriba
<code>round(n)</code>	Redondear al entero más cercano
<code>number(val)</code>	Convertir a número

Ejemplos de Funciones

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

Operadores

Operadores de Comparación

<code>=</code>	Igual
<code>!=</code>	No igual
<code><</code>	Menor que
<code><=</code>	Menor o igual que
<code>></code>	Mayor que
<code>>=</code>	Mayor o igual que

Lógicos y Aritméticos

<code>and</code>	AND lógico
------------------	------------

<code>or</code>	OR lógico
<code>not()</code>	NOT lógico (función)
<code>+</code>	Suma
<code>-</code>	Resta
<code>*</code>	Multiplicación
<code>div</code>	División
<code>mod</code>	Módulo
<code> </code>	Unión de conjuntos de nodos

Ejemplos de Operadores

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

Pruebas de Nodo

Tipos de Nodo

<code>node()</code>	Cualquier nodo (elemento, texto, comentario, PI)
<code>text()</code>	Solo nodo de texto
<code>comment()</code>	Solo nodo de comentario
<code>processing-instruction()</code>	Nodo de instrucción de procesamiento
<code>*</code>	Cualquier nodo elemento
<code>@*</code>	Cualquier nodo atributo
<code>element-name</code>	Elemento con nombre específico

Ejemplos de Prueba de Nodo

```
//p/text() # contenido de texto de <p>
//div/comment() # comentarios dentro de <div>
//body/node() # todos los nodos hijo de <body>
//div/* # todos los hijos elemento de <div>
```

Funciones Booleanas

<code>true()</code>	Booleano verdadero
<code>false()</code>	Booleano falso
<code>boolean(expr)</code>	Convertir a booleano
<code>not(expr)</code>	Negar booleano
<code>lang(code)</code>	Verdadero si el idioma del nodo coincide

Abreviaciones

Forma Corta vs Larga

<code>(ninguna)</code>	child:: (eje predeterminado)
<code>@</code>	attribute::
<code>//</code>	/descendant-or-self::node()/
<code>.</code>	self::node()
<code>..</code>	parent::node()
<code>[n]</code>	[position()=n]

Ejemplos Abreviados

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

Patrones Abreviados Comunes

```
//div[@id='main'] # div con id="main"
//table/td # todos los <td> en cualquier <table>
../sibling # hermano vía padre
//span # descendientes span del contexto
```

Patrones Comunes

Web Scraping / Testing

```
//input[@name='username'] # input de formulario por nombre
//button[text()='Submit'] # botón por texto
//div[contains(@class, 'error')] # elemento por clase parcial
//a[contains(@href, 'login')] # enlace por href parcial
```

Selección Condicional

```
//div[@class='item'][./span[@class='price']]
//tr[td[1]='Active'] # fila donde 1ra celda = Active
//*[contains(text(), 'Warning')] # cualquier elemento con texto
```

XPath en Python (lxml)

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

XPath en Selenium

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