

# Referencia Rápida de XPath

Ejes, predicados, funciones, operadores, selección de nodos

## Sintaxis

### Expresiones de Ruta

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

### Ejemplos Básicos

```
/html/body/div # ruta absoluta a <div>

```

### Combinar Rutas

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

## Ejes

### Direcciones de Eje

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

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

### Predicados Encadenados

```
//div[@class='list']//a[1] # primer <a> en div.list

```

## Funciones

### Funciones de Cadena

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

### Funciones Numéricas

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

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

### Lógicos y Aritméticos

<b>and</b>	AND lógico
<b>or</b>	OR lógico
<b>not()</b>	NOT lógico (función)
<b>+</b>	Suma
<b>-</b>	Resta
<b>*</b>	Multiplicación
<b>div</b>	División
<b>mod</b>	Módulo
	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

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

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

## Abreviaciones

### Forma Corta vs Larga

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

# Referencia Rápida de XPath

---

## 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']")
```