

Referensi Cepat XPath

Axis, predikat, fungsi, operator, pemilihan node

Sintaks

Ekspresi Path

/	Node akar (awal path absolut)
/bookstore/book	Pemilihan anak langsung
//book	Pilih semua node book di mana pun
.	Node konteks saat ini
..	Induk dari node saat ini
@lang	Atribut bernama lang
node()	Node apa pun dari tipe apa pun
*	Node elemen apa pun
@*	Atribut apa pun

Contoh Dasar

```
/html/body/div # path absolut ke <div>
//input[@type='text'] # semua input teks
//div[@class='main']/* # anak-anak div.main
//a/@href # semua atribut href
```

Menggabungkan Path

```
//book/title | //book/price # gabungan dua path
//h1 | //h2 | //h3 # beberapa tipe elemen
```

Axis

Arah Axis

child::	Anak langsung (axis default)
parent::	Induk langsung
ancestor::	Semua leluhur hingga akar
ancestor-or-self::	Leluhur + node saat ini
descendant::	Semua keturunan
descendant-or-self::	Keturunan + node saat ini
following::	Semua node setelah node saat ini dalam dokumen
following-sibling::	Saudara setelah node saat ini
preceding::	Semua node sebelum node saat ini dalam dokumen
preceding-sibling::	Saudara sebelum node saat ini
self::	Hanya node saat ini
attribute::	Atribut dari node saat ini
namespace::	Node namespace

Contoh Axis

```
//div/child::p # anak <p> dari <div>
//td/parent::tr # induk <tr> dari <td>
//h2/following-sibling::p # <p> setelah <h2>
//li/ancestor::ul # <ul> yang berisi <li>
```

Predikat

Pemfilteran dengan Predikat

```
//book[1] # elemen book pertama
//book[last()] # elemen book terakhir
//book[position() < 3] # dua book pertama
//book[@lang='en'] # book dengan lang="en"
//book[price > 30] # book dengan harga > 30
```

Pola Predikat

[n]	Elemen di posisi n (basis 1)
[last()]	Elemen terakhir
[last()-1]	Kedua dari terakhir
[@attr]	Memiliki atribut
[@attr='val']	Atribut sama dengan nilai
[element]	Memiliki elemen anak
[element='text']	Elemen anak berisi teks
[not(@attr)]	Tidak memiliki atribut

Predikat Berantai

```
//div[@class='list']//a[1] # <a> pertama dalam div.list
//input[@type='text'][@name='q'] # kondisi AND
//book[price>30][@lang='en'] # beberapa kondisi
```

Fungsi

Fungsi String

contains(s, sub)	Benar jika s mengandung sub
starts-with(s, pre)	Benar jika s dimulai dengan pre
string-length(s)	Panjang string
normalize-space(s)	Pangkas dan rapat-padatkan spasi
concat(a, b, ...)	Gabungkan string
substring(s, pos, len)	Ambil substring (basis 1)
translate(s, from, to)	Ganti karakter per karakter

Fungsi Numerik

sum(node-set)	Jumlah nilai numerik
count(node-set)	Jumlah node
floor(n)	Bulatkan ke bawah
ceiling(n)	Bulatkan ke atas
round(n)	Bulatkan ke integer terdekat
number(val)	Konversi ke angka

Contoh Fungsi

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

Operator

Operator Perbandingan

=	Sama dengan
!=	Tidak sama dengan
<	Kurang dari
<=	Kurang dari atau sama dengan
>	Lebih dari
>=	Lebih dari atau sama dengan

Logika dan Aritmatika

and	AND logika
or	OR logika
not()	NOT logika (fungsi)
+	Penjumlahan
-	Pengurangan
*	Perkalian
div	Pembagian
mod	Modulus
	Gabungan node set

Contoh Operator

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

Tes Node

Jenis Node

node()	Node apa pun (elemen, teks, komentar, PI)
text()	Hanya node teks
comment()	Hanya node komentar
processing-instruction()	Node instruksi pemrosesan
*	Node elemen apa pun
@*	Node atribut apa pun
element-name	Elemen dengan nama spesifik

Contoh Tes Node

```
//p/text() # konten teks dari <p>
//div/comment() # komentar dalam <div>
//body/node() # semua node anak dari <body>
//div/* # semua anak elemen dari <div>
```

Fungsi Boolean

true()	Boolean benar
false()	Boolean salah
boolean(expr)	Konversi ke boolean
not(expr)	Negasi boolean
lang(code)	Benar jika lang node cocok

Singkatan

Bentuk Pendek vs Panjang

(tidak ada)	child:: (axis default)
@	attribute::
//	/descendant-or-self::node()/
.	self::node()
..	parent::node()
[n]	[position()=n]

Contoh Singkatan

```
# Pasangan-pasangan ini setara:
child::div → div
attribute::href → @href
/descendant-or-self::node()/p → //p
self::node() → .
parent::node() → ..
```

Pola Singkatan Umum

```
//div[@id='main'] # div dengan id="main"
//table/td # semua <td> dalam <table> mana pun
../sibling # saudara via induk
//span # keturunan span dari konteks
```

Pola Umum

Web Scraping / Testing

```
//input[@name='username'] # input form berdasarkan nama
//button[text()='Submit'] # tombol berdasarkan teks
//div[contains(@class, 'error')] # elemen dengan class parsial
//a[contains(@href, 'login')] # tautan dengan href parsial
```

Pemilihan Kondisional

```
//div[@class='item'][./span[@class='price']]
//tr[td[1]='Active'] # baris di mana sel ke-1 = Active
//*[contains(text(), 'Warning')] # elemen apa pun dengan teks
```

Referensi Cepat XPath

XPath di Python (lxml)

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

XPath di Selenium

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