

RUTAS Xpath

//bookstore/book: Selecciona todos los elementos book que son hijos directos de bookstore.

//bookstore/book[1]: Selecciona el primer elemento book que es hijo directo de bookstore.

//bookstore/book[@category='children']: Selecciona todos los elementos book que tienen un atributo category con valor 'children' y que son descendientes de bookstore.

//title[@lang='en']: Selecciona todos los elementos title que tienen un atributo lang con valor 'en'.

//book[contains(title, 'Harry Potter')]: Selecciona todos los elementos book cuyo elemento title contiene la cadena 'Harry Potter'.

//bookstore/book[price>35.00]: Selecciona todos los elementos book que tienen un elemento price con un valor mayor a 35.00.

//bookstore/book[position()<=3]: Selecciona los primeros tres elementos book que son hijos directos de bookstore.

//bookstore/book[last()]: Selecciona el último elemento book que es hijo directo de bookstore.

//bookstore/book[author='J.K. Rowling' and @lang='en']: Selecciona todos los elementos book que tienen un elemento author con valor 'J.K. Rowling' y un atributo lang con valor 'en'.

//bookstore/book[1]/title | //bookstore/book[2]/title: Selecciona los elementos title de los dos primeros elementos book que son hijos directos de bookstore.

Para seleccionar el primer elemento div dentro del elemento body:

/html/body/div[1]

Para seleccionar el último elemento p en el documento:

(//p)[last()]

Para seleccionar todos los elementos a con un atributo href que comienza con "https":

//a[starts-with(@href,'https')]

Para seleccionar todos los elementos img con un atributo alt que contiene la palabra "logo":

```
//img[contains(@alt,'logo')]
```

Para seleccionar el elemento table que tiene un elemento caption con un texto igual a "Lista de usuarios":

```
//table[caption/text()='Lista de usuarios']
```

Para seleccionar el último elemento td dentro del último elemento tr dentro del último elemento table en el documento:

```
(//table[last()]/tr[last()]/td[last()])[1]
```

Operator	Description	Example
	Computes two node-sets	//book //cd
+	Addition	6 + 4
-	Subtraction	6 - 4
*	Multiplication	6 * 4
div	Division	8 div 4
=	Equal	price=9.80
!=	Not equal	price!=9.80
<	Less than	price<9.80
<=	Less than or equal to	price<=9.80
>	Greater than	price>9.80
>=	Greater than or equal to	price>=9.80
or	or	price=9.80 or price=9.70
and	and	price>9.00 and price<9.90
mod	Modulus (division remainder)	5 mod 2