

Linguaxe de navegación no documento XML

Para poder navegar nun documento XML e acceder aos elementos e atributos necesitaremos **XPath**.

Sumario

- 1 Introducción a XPath
 - ◆ 1.1 Expresións XPath
 - ◆ 1.2 Funcións estándar XPath
 - ◆ 1.3 XPath é utilizado en XSLT
- 2 Nodos XPath
 - ◆ 2.1 Terminoloxía XPath
 - ◆ 2.2 Relacións entre nodos
- 3 Sintaxe XPath
 - ◆ 3.1 Seleccionando Nodos
 - ◆ 3.2 Predicados
 - ◆ 3.3 Selección de nodos descoñecidos
 - ◆ 3.4 Seleccionando varias rutas
- 4 Eixos XPath
 - ◆ 4.1 Eixos XPath
 - ◆ 4.2 Expresión de localización XPath
- 5 Operadores XPath
- 6 Funcións XPath
 - ◆ 6.1 Funcións Numéricas
 - ◆ 6.2 Funcións de Agregado
 - ◆ 6.3 Funcións de Cadea
 - ◆ 6.4 Funcións de Contexto
- 7 Exemplos XPath
 - ◆ 7.1 Cargando o documento XML
 - ◆ 7.2 Seleccionando nodos
 - ◆ 7.3 Seleccionando todos os títulos
 - ◆ 7.4 Seleccionando o título do primeiro libro
 - ◆ 7.5 Seleccionando todos os prezos
 - ◆ 7.6 Seleccionando o prezo dos nodos con prezo > 35
 - ◆ 7.7 Seleccionando o título dos nodos con prezo > 35

Introdución a XPath

Que é XPath?

- XPath é unha sintaxe para definir partes dun documento XML
- XPath emprega expresións para navegar nos documentos XML
- XPath contén unha librería de funcións estándar
- XPath é unha peza chave en XSLT
- XPath é unha recomendación do W3C

Expresións XPath

XPath usa expresións de rutas para seleccionar un nodo ou un conxunto de nodos nun documento XML. Estas expresións son moi semellantes ás expresións que empregamos cando traballamos cun sistema tradicional de arquivos.

Funcións estándar XPath

XPath inclúe sobre 100 funcións no seu propio linguaxe. Hai funcións para xestionar cadeas de texto, valores numéricos, horas, nodos, secuencias, valores boolean, etc.

XPath é utilizado en XSLT

XPath é a chave principal do estándar XSLT. Sen coñecementos previos de XPath seríamos incapaces de transformar documentos XML empregando XSLT.

Nodos XPath

Terminoloxía XPath

Nodos

En XPath temos 7 clases distintas de nodos: element, attribute, text, namespace, processing-instruction, comment e nodos do documento.

Os documentos XML son tratados como árbores de nodos. O nodo máis alto da árbore é chamado o elemento root.

Mira o seguinte exemplo:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>
  <book>
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

Exemplo de nodos no documento XML anterior:

```
<bookstore> (elemento nodo raíz)

<author>J K. Rowling</author> (elemento nodo)

lang="en" (atributo nodo)
```

Valores atómicos

Os valores atómicos son nodos que non teñen fillos ou pai. Exemplo de valores atómicos:

```
J K. Rowling

"en"
```

Relacións entre nodos

Pai: Cada elemento e atributos teñen un pai.

No seguinte exemplo; o elemento libro é o pai de title, author, year e prezo:

```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

Fillos: Os nodos elemento poden ter cero, un ou máis fillos.

No seguinte exemplo; title, author, year e price son fillos do elemento book:

```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
```

```
<year>2005</year>
<price>29.99</price>
</book>
```

Irmáns: Nodos que teñen o mesmo pai

No seguinte exemplo; title, author, year e price son todos irmáns.

```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

Antepasados: O pai dun nodo, o pai do pai dun nodo, etc.

No seguinte exemplo; os antepasados do elemento title son o elemento book e o elemento bookstore:

```
<bookstore>

<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

Descendentes: Os fillos dun nodo, os fillos dos fillos dun nodo, etc.

No seguinte exemplo ; os descendentes do elemento bookstore son book, title, author, year e o elemento price:

```
<bookstore>

<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

Sintaxe XPath

XPath emprega expresións de traxectoria ou camiño para seleccionar nodos ou conxuntos de nodos nun documento XML.

O nodo será seleccionado seguindo a traxectoria ou pasos indicados.

Exemplo de documento XML de referencia neste apartado:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book>
  <title lang="eng">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="eng">Learning XML</title>
  <price>39.95</price>
```

</book>

</bookstore>

Seleccionando Nodos

XPath emprega expresións de ruta ou camiño para seleccionar nodos ou conxuntos de nodos nun documento XML. O nodo é seleccionado empregando unha ruta ou camiño. As rutas máis usadas son:

Expresión	Descrición
nodename	Selecciona todos os nodos fillos do nodename indicado.
/	Selecciona dende o nodo raíz.
//	Selecciona nodos no documento dende o nodo actual que indica a selección, da igual onde estea.
.	Selecciona o nodo actual.
..	Selecciona o nodo pai do nodo actual.
@	Selecciona atributos.

Na táboa inferior temos escritas algunhas expresións XPath e o resultado das mesmas:

Expresión XPath	Resultado
/bookstore	Selecciona o elemento raíz bookstore. Nota: Se a ruta comeza con (/) sempre representa unha ruta absoluta a un elemento.
/bookstore/book	Selecciona todos os elementos book que son fillos de bookstore
//book	Selecciona todos os elementos book non importa onde se atopen no documento.
/bookstore//book	Selecciona todos os elementos book que son fillos de bookstore, non importa onde estean debaixo do elemento bookstore.
//@lang	Selecciona todos os atributos que teñen de nome lang

Predicados

Os predicados úsanse para encontrar un nodo específico ou un nodo que contén un valor específico.

Os predicados póñense sempre entre corchetes []

Na seguinte táboa temos un listado de rutas de expresión con predicados e o resultado de ditas expresións:

Expresión XPath	Resultado
/bookstore/book[1]	Selecciona o primeiro elemento book que é fillo do elemento bookstore. Nota: En IE5 e posteriores implementaron que [0] debería ser o primeiro nodo, pero de acordo có W3C debería haber sido [1]!
/bookstore/book[last()]	Selecciona o derradeiro elemento book que é fillo do elemento bookstore.
/bookstore/book[last()-1]	Selecciona o elemento book anterior ó derradeiro que é fillo do elemento bookstore.
/bookstore/book[position()<3]	Selecciona os dous primeiros elementos book que son fillos do elemento bookstore.
//title[@lang]	Selecciona todos os elementos title que teñen un atributo chamado lang.
//title[@lang='eng']	Selecciona todos os elementos title que teñen un atributo chamado lang cun valor de 'eng'.

Expresión XPath	Resultado
/bookstore/book[price>35.00]	Selecciona todos os elementos book do elemento bookstore que teñen un elemento price cun valor maior que 35.00
/bookstore/book[price>35.00]/title	Selecciona todos os elementos title dos elementos book do elemento bookstore que teñen un elemento price cun valor maior que 35.00

Selección de nodos descoñecidos

A continuación amosamos algunhas expresións e wildcards que nos permiten seleccionar elementos XML descoñecidos.

Wildcard	Descrición
*	Selecciona calquera nodo.
@*	Selecciona calquera atributo dos nodos.
node()	Selecciona calquera nodo de calquera clase.

Na seguinte táboa temos un exemplo de uso de eses wildcards:

Expresión XPath	Resultado
/bookstore/*	Selecciona todos os elementos fillos do elemento bookstore
//*	Selecciona todos os elementos no documento.
//title[@*]	Selecciona todos os elementos title que teñan algún atributo.

Seleccionando varias rutas

Empregando o operador | nunha expresión XPath podemos seleccionar varias traxectorias

Expresión XPath	Resultado
//book/title //book/price	Selecciona todos os elementos title e price dos elementos book.
//title //price	Selecciona todos os elementos title e price no documento.
/bookstore/book/title //price	Selecciona todos os elementos title dos elementos book do elemento bookstore e todos os elementos price no documento.

Eixos XPath

Empregaremos o seguinte documento XML na lista de exemplos máis abaixo.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>

  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>

</bookstore>
```

Eixos XPath

Un eixo define un conxunto de nodos relativos ó nodo actual.

Nome do Eixo	Resultado
ancestor	Selecciona todos os antecesoros (pai, avó,etc.) do nodo actual.
ancestor-or-self	Selecciona todos os antecesoros (pai, avó, etc.) de nodo actual e tamén o nodo actual.
attribute	Selecciona todos os atributos do nodo actual.
child	Selecciona todos os fillos do nodo actual.
descendant	Selecciona todos os nodos descendentes (fillos, netos, etc.) do nodo actual.
descendant-or-self	Selecciona todos os nodos descendentes (fillos, netos,etc.) do nodo actual e tamén o nodo actual.
following	Selecciona todo no documento a partir da etiqueta de peche do nodo actual.
following-sibling	Selecciona todos os irmáns a partir do nodo actual.
namespace	Selecciona todos os namespace nodos do nodo actual.
parent	Selecciona o pai do nodo actual.
preceding	Selecciona todo no documento que está antes da etiqueta de comezo do nodo actual.
preceding-sibling	Selecciona todos os irmáns antes do nodo actual.
self	Selecciona o nodo actual.

Expresión de localización XPath

Unha ruta de localización pode ser absoluta ou relativa.

Unha ruta absoluta comeza con (/) e unha ruta relativa non. En ambos casos a ruta de localización consiste nun ou máis pasos, cada un separado por (/):

Ruta de localización absoluta:
/paso/paso/...

Ruta de localización relativa:
paso/paso/...

Cada paso é avaliado con respecto ó nodos no conxunto actual de nodos. Un paso consiste en: - un eixo - un nodo test (identifica un nodo dentro dun eixo) - cero ou máis predicados (para refinar o conxunto de nodos actual)

A sintaxe para un paso de localización é:

axisname::nodetest[predicate]

Exemplos

Exemplo	Resultado
child::book	Selecciona todos nodos book que son fillos do nodo actual.
attribute::lang	Selecciona o atributo lang do nodo actual.
child::*	Selecciona todos os fillos do nodo actual.

Exemplo	Resultado
attribute::*	Selecciona todos os atributos do nodo actual.
child::text()	Selecciona todos os nodos de texto fillos do nodo actual.
child::node()	Selecciona todos os nodos fillos do nodo actual.
descendant::book	Selecciona todos os nodos book descendentes do nodo actual.
ancestor::book	Selecciona todos os nodos book antecesoros do nodo actual.
ancestor-or-self::book	Selecciona todos os nodos book antecesoros do nodo actual e tamén o nodo actual si é un nodo "book".
child::* / child::price	Selecciona todos os nodos "price" netos do nodo actual.

Operadores XPath

Unha expresión XPath devolve un conxunto de nodos, unha cadea, un Boolean ou un número.

A continuación temos unha lista de operadores que se poden empregar nas expresións XPath:

Operador	Descrición	Exemplo	Valor devolto
	Calcula dous conxuntos de nodos	//book //cd	Devolve un conxunto de nodos con todos os elementos "book" e "cd".
+	Adición	6 + 4	10
-	Sustraccións	6 - 4	2
*	Multipliación	6 * 4	24
div	Division	8 div 4	2
=	Igualdade	prezo=9.80	verdadeiro se prezo é 9.80, falso noutro caso.
!=	Distinto	prezo != 9.80	verdadeiro se prezo é distinto de 9.80, falso se prezo é igual a 9.80
<	Menor que	prezo < 9.80	verdadeiro se prezo é menor que 9.80, falso se prezo é igual o maior que 9.80
<=	Menor ou igual a	prezo <= 9.80	verdadeiro se prezo é menor ou igual a 9.80, falso se prezo é maior que 9.80
>	Maior que	prezo > 9.80	verdadeiro se prezo é maior que 9.80, falso se é igual ou menor que 9.80
>=	Maior ou igual a	prezo >= 9.80	verdadeiro se prezo é maior ou igual a 9.80, falso se é menor que 9.80
or	ou	prezo = 9.80 or prezo = 9.70	verdadeiro se prezo é igual a 9.80 ou igual a 9.70, falso noutro caso.
and	e	prezo>9.00 and prezo < 9.90	verdadeiro se prezo é maior que 9.00 e menor que 9.90
mod	resto da división	5 mod 2	1

Funcións XPath

Existen multitude de funcións XPath que podemos empregar para facer cálculos numéricos, extraer textos, funcións booleanas, de contexto, data e hora, etc.

Funcións Numéricas

Función	Descrición
number(arg)	Devolve o valor numérico do argumento. O argumento podería ser un valor booleano, cadea ou conxunto de nodos. Por exemplo: number("100") -> Resultado: 100
abs(num)	Devolve o valor absoluto do argumento. Exemplo: abs(-3.14) -> Resultado: 3.14
ceiling(num)	Devolve o valor integer inmediatamente superior ó argumento. Exemplo: ceiling(3.14) -> Resultado: 4
floor(num)	Devolve o valor integer inmediatamente inferior que non é superior ó argumento. Exemplo: floor(3.14) -> Resultado: 3
round(num)	Redondea o argumento ó integer máis próximo. Exemplo: round(3.14) -> Resultado: 3
round-half-to-even()	Exemplo: round-half-to-even(0.5) -> Resultado: 0; round-half-to-even(1.5) -> Resultado: 2; round-half-to-even(2.5) -> Resultado: 2

Funcións de Agregado

Función	Descrición
count((item,item,...))	Devolve o número de nodos
avg((arg,arg,...))	Devolve a media dos argumentos. Exemplo: avg((1,2,3)) -> Resultado: 2
max((arg,arg,...))	Devolve o argumento maior. Exemplo: max((1,2,3)) -> Resultado: 3; max(('b','t')) -> Resultado: 't'
min((arg,arg,...))	Devolve o argumento menor. Exemplo: min((1,2,3)) -> Resultado: 1; max(('b','t')) -> Resultado: 'b'
sum(arg,arg,...)	Devolve a suma dos valores numéricos

Funcións de Cadea

Función	Descrición
concat(string s1, string s2, string s3, ...)	Devolve un único string resultado de concatenar os parámetros
contains(string s1, string s2)	Devolve true se s1 contén a s2
normalize-space(string s)	Devolve s cos espazos normalizados (sen espazos consecutivos, etc.)
starts-with(string s1, string s2)	Devolve true se s1 empeza por s2, false en caso contrario
string-length(string s)	Devolve o número de caracteres de s
substring(string s, number posición, number numCaracteres)	Devolve o substring de s, de lonxitude numCaracteres empezando a extraer en posición
substring-after(string s1, string s2)	Devolve o substring de s1 logo da primeira ocorrencia de s2
substring-before(string s1, string s2)	Devolve a parte anterior de s1 antes da primeira ocorrencia de s2
translate(string s1, string s2, string s3)	Devolve s1 cos caracteres de s2 substituídos polos seus correspondentes de s3

Funcións de Contexto

Función	Descrición
current-dateTime()	Devolve a dataHora actual (con Timezone)
current-date()	Devolve a data actual (con Timezone)
current-time()	Devolve a hora actual (con Timezone)

Función	Descripción
implicit-timezone()	Devolve o valor do Timezone

Visita esta URL para máis referencias de funcións XPath.

Exemplos XPath

A continuación amosaremos algúns exemplos de uso de XPath.

Empregaremos o seguinte documento "books.xml":

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>

<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>

<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>

</bookstore>
```

Cargando o documento XML

Poderemos empregar dende Javascript unha petición Ajax XMLHttpRequest para cargar o documento XML.

O código empregado nos navegadores modernos:

```
var xmlhttp=new XMLHttpRequest()

// En navegadores antigos de Microsoft (IE 5 e 6) empregaríamos:

var xmlhttp=new ActiveXObject("Microsoft.XMLHTTP")
```

Seleccionando nodos

Para seleccionar nodos temos 3 xeitos diferentes de facelo. Aquí se amosa o código que debería funcionar na maioría dos navegadores. Internet Explorer emprega o método **selectNodes()** para seleccionar os nodos no documento XML:

```
xmlDoc.selectNodes(xpath);
```

Firefox, Chrome, Opera e Safari empregan o método **evaluate()** para seleccionar os nodos dun documento XML:

```
xmlDoc.evaluate(xpath, xmlDoc, null, XPathResult.ANY_TYPE,null);
```

Seleccionando todos os títulos

O seguinte exemplo selecciona todos os títulos dos libros:

```
/bookstore/book/title
```

Seleccionando o título do primeiro libro

O seguinte exemplo selecciona o título do primeiro libro debaixo do elemento bookstore:

```
/bookstore/book[1]/title
```

Hai un problema con isto. O exemplo anterior amosa resultados diferentes no IE e noutros browsers. IE e posteriores implementaron que [0] debería ser o primeiro nodo, pero de acordo ao standard W3C debería ser [1].

Para solucionar ese problema poderíase facer empregando como linguaxe de selección (SelectionLanguage) o XPath. Exemplo:

```
xml.setProperty("SelectionLanguage","XPath");  
xml.selectNodes("/bookstore/book[1]/title");
```

Seleccionando todos os prezos

Na seguinte instrución seleccionaremos o texto de todos os nodos prezo:

```
/bookstore/book/price/text()
```

Seleccionando o prezo dos nodos con prezo > 35

```
/bookstore/book[price>35]/price
```

Seleccionando o título dos nodos con prezo > 35

```
/bookstore/book[price>35]/title
```