

Apuntes XPath

Rutas

Ruta Absoluta

Para seleccionar un elemento del xml podemos usar una ruta absoluta introduciendo todo el camino hacia el elemento que queramos.

```
/nodo_padre/nodo_hijo/nodo_deseado
```

Ruta relativa

Cuando nos encontremos dentro de una ruta podemos usar la ruta relativa a nuestra posición actual dentro del xml.

```
./nodo_deseado o //nodo_deseado
```

Varias rutas

Si queremos usar dos rutas posibles las juntaremos con “|”.

```
/nodo_padre/nodo_hijo/nodo_deseado | /nodo_padre/nodo_deseado
```

Predicados

Por contenido

Nos extraerá los nodos que seleccionemos a partir de su contenido que sean hijos del mismo.

```
/nodo_padre/nodo_hijo[nodo_deseado="Contenido_elegido"]
```

Por atributo

Esto mismo también sirve para seleccionar los nodos hijos del nodo con el atributo que queramos.

```
/nodo_padre/nodo_hijo[nodo_deseado@Contenido_elegido]
```

Varios predicados

Se pueden añadir varios predicados para que se cumplan los dos dentro del mismo elemento padre, se cumplirá como una condición lógica AND.

```
/nodo_padre/nodo_hijo[nodo_deseado="Contenido_elegido"][nodo_deseado2="Contenido_elegido2"]
```

Funciones

Function	Description	Syntax
boolean()	The boolean function evaluates an expression and returns true or false.	boolean(expression)

Function	Description	Syntax
ceiling()	The ceiling function evaluates a decimal number and returns the smallest integer greater than or equal to the decimal number.	ceiling(number)
choose()	The choose function returns one of the specified objects based on a boolean parameter.	choose(boolean, object1, object2)
concat()	The concat function concatenates two or more strings and returns the resulting string.	concat(string1, string2 [,stringn]*)
contains()	The contains function determines whether the first argument string contains the second argument string and returns boolean true or false.	contains(haystack, needle)
count()	The count function counts the number of nodes in a node-set and returns an integer.	count(node-set)
element-available()	The element-available function determines if an element is available and returns true or false.	element-available(QName)
false()	The false function returns boolean false.	false()
floor()	The floor function evaluates a decimal number and returns the largest integer less than or equal to the decimal number.	floor(number)
function-available()	The function-available function determines if a given function is available and returns boolean true or false.	function-available(name)
id()	The id function finds nodes matching the given ids and returns a node-set containing the identified nodes.	id(expression)
lang()	The lang function determines whether the context node matches the given language and returns boolean true or false.	lang(string)
last()	The last function returns a number equal to the context size from the expression evaluation context.	last()
local-name()	The local-name function returns a string representing the local name of the first node in a given node-set.	local-name([node-set])
name()	The name function returns a string representing the QName of the first node in a given node-set.	name([node-set])
namespace-uri()	The namespace-uri function returns a string representing the namespace URI of the first node in a given node-set.	namespace-uri([node-set])
normalize-space()	The normalize-space function strips leading and trailing whitespace from a string, replaces sequences of whitespace characters by a single space, and returns the resulting string.	normalize-space([string])
not()	The not function evaluates a boolean expression and returns the opposite value.	not(expression)
number()	The number function converts an object to a number and returns the number.	number([object])

Function	Description	Syntax
position()	The position function returns a number equal to the context position from the expression evaluation context.	position()
round()	The round function returns a number that is the nearest integer to the given number.	round(decimal)
starts-with()	The starts-with checks whether the first string starts with the second string and returns true or false.	starts-with(haystack, needle)
string()	The string function converts the given argument to a string.	string([object])
string-length()	The string-length function returns a number equal to the number of characters in a given string.	string-length([string])
substring()	The substring function returns a part of a given string.	substring(string, start) substring(string, start, length)
substring-after()	The substring-after function returns a string that is the rest of a given string after a given substring.	substring-after(haystack, needle)
substring-before()	The substring-before function returns a string that is the part of a given string before a given substring.	substring-before(haystack, needle)
sum()	The sum function returns a number that is the sum of the numeric values of each node in a given node-set.	sum(node-set)
translate()	The translate function evaluates a string and a set of characters to translate and returns the translated string.	translate(string, abc, XYZ)
true()	The true function returns a boolean value of true.	true()