

# XPath チートシート

- XPath においては、配列は 0 からではなく1から始まる ([1]で1番目のノード)
- 先頭に / があれば絶対パス、なければ相対パス、//なら全要素が対象。
- () で評価順序を明記できる。

	省略構文	「完全な」構文 (//,@ など一部省略あり)
全要素	//*	/descendant-or-self::node()/child::*
全 div 要素	//div	/descendant-or-self::node()/child::div
class 属性が "hoge" な div 要素	//div[@class="hoge"]	//child::div[attribute::class="hoge"]
id 属性が "hoge" な要素	//*[@id="hoge"]	//child::*[attribute::id="hoge"]
Title 属性が "hoge" で class 属性が "fuga" でない要素	//*[@title="hoge" and @class!="fuga"]	/descendant::*[@title="hoge" and @class!="fuga"]
Form 要素の 3 番目の input 要素	//form/descendant::input[3]	/descendant::form/descendant::input[3]
チェックされたチェックボックスの親要素	//input[@checked="checked"]/..	//input[@checked="checked"]/parent::node()
a 要素の href 属性	//a/@href	//descendant::a/attribute:href
ルート of A 要素中の B 要素中の C 要素全て	/A/B/C	/child::A/child::B/child::C
A 要素の子孫の B 要素の 1 番目の子要素	A/B/*[1]	child::A//child::B/child::*[1]
href 属性が 'help.php'、親要素が div 要素でかつ親要素 class 属性が 'header' な a 要素の target 属性	//a[@href='help.php'][name(.)='div']/. /@class='header']/@target	(「完全な」構文は省略)
Price 属性の数が discount 属性の 2 倍以上な item 要素	//item[@price >= 2*@discount]	略
子に x か y をもつ親 v と子 z をもつ w の和集合	v [ x or y ]   w [ z ]	略
5 番目の Page 要素	Page[ 5 ]	Page[position() = 5]

軸 (全てコンテキストノード基準)	完全な構文	省略構文
子ノード	Child	(何も書かない)
子孫ノード	Descendant	
親ノード	Parent	..
祖先ノード	Ancestor	
兄弟ノードのうち前方のノード	preceding-sibling	
兄弟ノードのうち後方のノード	following-sibling	
前方にある全てのノード	preceding	
後方にある全てのノード	following	
要素の場合は属性ノード	attribute	@
要素の場合は名前空間ノード (コンテキストノード) 自身	namespace	.
自身とその子孫ノード	descendant-or-self	//
自身とその祖先ノード	ancestor-or-self	

## ノードテスト

### 名前

その名前の全ての要素、属性、名前空間ノード

### text()

ノード中の文字列。例: <k>こんにちは</k> の

中の 'こんにちは'

### comment()

全ての XML コメントノードを指定する。例: <!--

コメント -->

### processing-instruction()

全ての XML 処理命令ノードを指定する。<?xsl:stylesheet href="article.css" ?> を指定するには processing-instruction('xsl:stylesheet') と記述すればよい。

node() 全てのノードを指定する。

\*

主ノード型の全てのノードを指定する。要するに、要素、属性、名前空間ノード。

関数	型	説明
ノード集合関数	Position()	評価中のノードが兄弟ノードの何番目か返す
	count( <i>node-set</i> )	<i>number</i> 引数のノード集合 (もしくはノード集合を返す式) のノードの数を返す。
	id( <i>object</i> )	<i>node-set</i> 引数のオブジェクトの文字列値を ID 型の属性値としてもつノードの集合を返す。
	name( <i>node-set</i> ?)	<i>string</i> 引数として渡されたノード集合の最初のノードの名前を返す (ノードが要素の場合は要素名、属性の場合は属性名)。
文字列関数	string( <i>object</i> ?)	仕様で定められた変換規則により文字列に変換する。引数として XPath 式も渡すことができる。
	substring( <i>string</i> , <i>number</i> , <i>number</i> ?)	<i>string</i> 引数として渡す文字列の部分文字列を返す。
	concat( <i>string</i> , <i>string</i> , <i>string</i> *)	引数として渡す複数の文字列を連結して返す。
	normalize-space( <i>string</i> ?)	引数の文字列を正規化 (文字列の前後の空白文字と文字列中に連続して現れる空白文字を一つの空白で置き換え) し、返す。
	string-length( <i>string</i> ?)	<i>number</i> 引数として渡す文字列の長さ (文字の数) を返す。
contains( <i>string1</i> , <i>string2</i> )	<i>boolean</i> 文字列 <i>string1</i> に文字列 <i>string2</i> が含まれていた場合、true を返す。	
論理関数	not( <i>boolean</i> )	<i>boolean</i> 引数の論理値の逆の値を返す。
数値関数	sum( <i>node-set</i> )	<i>number</i> 引数として渡されたノード集合の各ノードの文字列値を、仕様で定められた変換規則にしたがって数値に変換し、合計した値を返す。

# XPath

## 関数一覧

製作 : Satosi

### ノード集合関数

<i>number last()</i>
コンテキストサイズを返す。
<i>number position()</i>
コンテキスト位置を返す。
<i>number count(node-set)</i>
引数に指定したノード集合に含まれるノード数を返す。
<i>node-set id(object)</i>
ID によりエレメントを選択する。
<i>string local-name(node-set?)</i>
<b>local-name</b> 関数は引数に指定したノード集合内のノードのうち、ドキュメント順で展開された名前の最初のノードのローカルパートを返す。引数を省略した場合は、コンテキストノードを唯一のメンバに持つノード集合をデフォルトとして使用する。
<i>string namespace-uri(node-set?)</i>
<b>namespace-uri</b> 関数は引数に指定したノード集合内のノードのうち、ドキュメント順で最初のノードの展開された名前のネームスペース URI を返す。引数を省略した場合は、コンテキストノードを唯一のメンバに持つノード集合をデフォルトとして使用する。返す文字列は、エレメントノードとアトリビュートノード以外は空になる。
<i>string name(node-set?)</i>
<b>name</b> 関数は引数に指定したノード集合内のノードのうち、ドキュメント順で最初のノードの展開された名前を表現する QName を持つ文字列を返す。引数を省略した場合は、コンテキストノードを唯一のメンバに持つノード集合をデフォルトとして使用する。エレメントノードとアトリビュートノードを除き、 <b>name</b> 関数が返す文字列は <b>local-name</b> 関数が返す文字列と同じである。

### 文字列関数

<i>string string(object?)</i>
オブジェクトを文字列に変換する。
<i>string concat(string, string, string*)</i>
引数を連結して返す。
<i>boolean starts-with(string, string)</i>
1番目の文字列が2番目の文字列で始まっている場合に真を返し、それ以外は偽を返す。
<i>boolean contains(string, string)</i>
1番目の文字列が2番目の文字列を含んでいる場合に真を返し、それ以外は偽を返す。
<i>string substring-before(string, string)</i>
2番目の文字列が1番目の文字列内で最初に見つかった場合に、その文字列よりも前にある文字列 を返す。
<i>string substring-after(string, string)</i>
2番目の文字列が1番目の文字列内で最初に見つかった場合に、その文字列よりも後にある文字列 を返す。
<i>string substring(string, number, number?)</i>
1番目の文字列のうち、2番目に指定した位置から始まる文字列を、3番目で指定した長さだけ返す。3番目を指定しないと、2番目に指定した位置から最後まで文字列 を返す。
<i>number string-length(string?)</i>
文字列内の文字数を返す。引数を省略した場合は、コンテキストノードの文字列値をデフォルトとして使用する。
<i>string normalize-space(string?)</i>
引数に指定した文字列の空白文字を正規化して返す。つまり前後の空白文字を取り除き、連続する空白文字を1つの空白文字に置き換える。引数を省略した場合は、コンテキストノードの文字列値をデフォルトとして使用する。
<i>string translate(string, string, string)</i>
1番目の引数に指定した文字列内に2番目の引数に指定した文字列内の文字があった場合、その文字を3番目の引数に指定した文字列内の対応する位置の文字で置き換えて返す。たとえば <code>translate("bar", "abc", "ABC")</code> は <code>BAR</code> を返す。2番目の引数に指定した文字列が3番目の引数に指定した文字列よりも長いために、2番目の引数に指定した文字列内の文字に対応する文字が3番目の引数に指定した文字列内に存在しない場合は、1番目の引数に指定した文字列内の対応する文字を削除する。たとえば <code>translate("--aaa--", "abc-", "ABC")</code> は <code>"AAA"</code> を返す。2番目の文字列内に同じ文字が複数ある場合には、最初の文字を使用して置き換えが行われる。

### ブール関数

<i>boolean boolean(object)</i>
引数をブール値に変換する。
<i>boolean not(boolean)</i>
引数が偽の場合に真を、それ以外は偽を返す。
<i>boolean true()</i>
真を返す。
<i>boolean false()</i>
偽を返す。
<i>boolean lang(string)</i>
<b>lang</b> 関数は、コンテキストノードの <code>xml:lang</code> という名前のアトリビュートで指定したコンテキストノードの言語が引数に指定したものと同じ言語であるか、または引数に指定した言語のサブ言語であるかにより、真または偽を返す。コンテキストノードの言語は、コンテキストノードの <code>xml:lang</code> という名前のアトリビュートの値で決まる。あるいは、コンテキストノードが <code>xml:lang</code> という名前のアトリビュートを持たない場合は、コンテキストノードに最も近い祖先エレメントが持つ <code>xml:lang</code> という名前のアトリビュートの値で決まる。そのようなアトリビュートがない場合は、 <b>lang</b> 関数は偽を返す。そのようなアトリビュートがある場合、 <b>lang</b> 関数は大文字と小文字を区別せずに比較して、アトリビュート値と引数が等しい場合は真を返す。またアトリビュート値が <code>-</code> で始まるサフィックスを持つ場合は、そのサフィックスは考慮せず、そして大文字と小文字を区別することなく比較して、アトリビュート値が引数に等しい場合にも真を返す。たとえばコンテキストノードが以下の5つのエレメントのいずれかの場合に、 <code>lang("en")</code> は真を返す。 <pre>&lt;para xml:lang="en"/&gt; &lt;div xml:lang="en"&gt;&lt;para/&gt;&lt;/div&gt; &lt;para xml:lang="EN"/&gt; &lt;para xml:lang="en-us"/&gt;</pre>

### 数値関数

<i>number number(object?)</i>
引数を数値に変換する。
<i>number sum(node-set)</i>
各ノードの文字列値を数値に変換し、合計した値を返す。
<i>number floor(number)</i>
引数に指定した数値よりも大きくない範囲で、最も大きい整数を返す。
<i>number ceiling(number)</i>
引数に指定した数値よりも小さくない範囲で、最も小さい整数を返す。
<i>number round(number)</i>
引数に指定した値に最も近い整数を返す。

# XPath for Javascript

```
// for Javascript
var result = document.evaluate(
    '//div', // ここに Xpath
    document, // 評価対象。document.body とすると head 以下の要素は取得されない
    null, // HTML 文書の場合や、名前空間接頭辞が使われていない場合は null
    7, // 結果の種類を指定する。別表の「定義済みの結果型定数」と「値」を参照。
        // よく分からないなら 7 か、ORDERED_NODE_SNAPSHOT_TYPE。
    null // XPathResult オブジェクトを再利用するならばそれを、しないなら null
);
result.snapshotLength; // 取得した要素(正確にはノード)の数。
result.snapshotItem(0); // 1 個目の要素
result.snapshotItem(1); // 2 個目の要素
```

定義済みの結果型定数	値	戻り値関係	解説
ANY_TYPE	0	resultType から型 (この表の値)が取得可	式の評価によって導き出される適切な型を格納した結果の集合。結果がノード集合ならば、結果の型は常に UNORDERED_NODE_ITERATOR_TYPE となるので注意が必要。
NUMBER_TYPE	1	numberValue	一つの数値を格納した結果。count() 関数を使用した XPath 式などで有用。
STRING_TYPE	2	stringValue	一つの文字列を格納した結果。
BOOLEAN_TYPE	3	booleanValue	一つの真偽値を格納した結果。not() 関数を使用した XPath 式などで有用。
UNORDERED_NODE_ITERATOR_TYPE	4	iterateNext() イテレータとして使える。	式にマッチした全てのノードを格納した結果ノード集合。 ノードの順番は文書内に現れる順番と必ずしも一致しない。
ORDERED_NODE_ITERATOR_TYPE	5	反復終了後は null を返す	同上。ノードの順番は文書内に現れる順番に一致する。
UNORDERED_NODE_SNAPSHOT_TYPE	6	snapshotItem(番号) snapshotLength	式にマッチした全てのノードのスナップショットを格納した結果ノード集合。 ノードの順番は文書内に現れる順番と必ずしも一致しない。
ORDERED_NODE_SNAPSHOT_TYPE	7	文書が変更しても変更 されない。	同上。ノードの順番は文書内に現れる順番に一致する。
ANY_UNORDERED_NODE_TYPE	8	singleNodeValue	式にマッチしたノードのうちのどれか一つを格納した結果ノード集合。 これは必ずしも文書内で式にマッチした最初のノードというわけではない。
FIRST_ORDERED_NODE_TYPE	9	singleNodeValue	文書内で式にマッチした最初のノードを格納した結果ノード集合。

## 参考資料

<http://www.infoteria.com/jp/contents/xml-data/REC-xpath-19991116-jpn.htm>

[http://developer.mozilla.org/ja/docs/Introduction\\_to\\_using\\_XPath\\_in\\_JavaScript](http://developer.mozilla.org/ja/docs/Introduction_to_using_XPath_in_JavaScript)

<http://d.hatena.ne.jp/amachang/20071112/1194856493>

[http://ja.wikipedia.org/wiki/XML\\_Path\\_Language](http://ja.wikipedia.org/wiki/XML_Path_Language)

<http://svn.coderepos.org/share/lang/javascript/javascript-xpath/trunk/test/functional/index.html>