E4X in Firefox

nanto_vi (TOYAMA Nao)

自己紹介

- 外山真（とやまなお）
  - a.k.a. nanto_vi (なんと)
  - http://www.ne.jp/asahi/nanto/moon/
  - http://nanto.asablo.jp/blog/
- 肩書き： 学生
- Not in ~~Education, Employment or~~ Training

E4X とは？

- ECMAScript for XML
- ECMAScript にネイティブ XML サポートを追加
  - ネイティブ → Date, Array and etc.
- Firefox 1.5/JavaScript 1.6 で実装

型

- XML 型
  - DOM Node に相当
  - 属性・テキストノードなども含む
- XMLList 型
  - DOM NodeList/DocumentFragment に相当
  - 1 項目の XMLList は XML として扱える

XML/XMLList コンストラクタ

- new XML("<a><b/>")
  - → <parent xmlns="[デフォルト名前空間]"><a><b/></parent>
  - XML 宣言・文書型宣言は含められない
- new XMLList("<a/><b/><c/>")

リテラル表記

- <a><b/>
- <><a/><b/><c/></>
- 式 (Expression) の埋め込み
  - <{elem} {attr}={value}>{content}</{elem}>
    - →
    - <a>{ "<b/>" } → <a><b/>

演算子

- 文字列化
  - <a>content → "content"
  - <a><b/> → "<a>\n <b/>\n"
- typeof <a/> == typeof <></> == "xml"
- 等価演算子
  - <a/> == <a/> (cf. {} != {}, [] != [])
  - <a/> !== <a/>
- <a/> + <b/> == <><a/><b/></>

ノードの取得 (1)

- x.child, x["child"]
  - 子要素を収めた XMLList
  - x が XMLList のときは、x 中の各 XML の子要素
- x..descendant
  - 子孫要素を収めた XMLList
- x.*, x..*
  - 子ノード、子孫ノードを収めた XMLList
  - テキストノードを含む

ノードの取得 (2)

- list[n]
  - XMLList 中の n 番目の XML
  - XML 型の x に関して、x[0] === x
  - x.*[n]
    - n 番目の子ノード
- x.@attr, x.@["attr"]
  - 属性 (XML 型) を収めた XMLList
- 存在しなければ空 XMLList
  - <a><b/>.c → <></>

フィルタリング

- list.(expr)
  - フィルタリング結果の XMLList
  - expr 評価時には list 中の各要素がそれぞれスコープチェーンに追加される
  - books.book.(@author == "Alan")
    - author 属性が "Alan" である book 要素の集合

プロパティへの代入

- XML/XMLList 型を代入
  - 子要素が置き換わる
  - x = <a>c
  - x.b = <d/>
    - → <a><d/>
- その他の型を代入
  - 子要素の内容が置き換わる
  - x.b = "other than xml"
    - → <a>other than xml

存在しないプロパティへの代入

- XML/XMLList 型を代入
  - 要素を最後に追加
  - x.d = <d/> → <a>c<d/>
- その他の型を代入
  - 要素を作成し最後に追加
  - x.d = "e" → <a>c<d>e</d>
- 途中の要素も作成される
  - x = <a/>; x.b.c.d = "";
    - → <a><c><d/></c>

ノードの削除

- delete 演算子
  - x = <c/>
    - delete x.b → <c/>
    - delete x.*[0] → <c/>
    - delete x.@attr → <a><c/>
- 空文字列の代入
  - x = <a><c/>; x.b = "";
    - → <a>

メソッド呼び出し

- x = <a><b/><c/><d/>
  - x.*.length → <></>
  - x.*.length() → 3
- メソッドの移譲
  - XMLList → XML
    - <><a/></>.name() → a
  - XML → String
    - <a>b.toUpperCase() → "B"
    - replace メソッド (既に XML にある) 以外

各種メソッド

- appendChild, prependChild, insertChildBefore, insertChildAfter
  - x.appendChild(xml) ≒ x.* += xml
  - E4X: x.appendChild(c) == x
  - DOM: x.appendChild(c) == c
- nodeKind()
  - "element", "attribute", "text" and etc.
- copy(), parent(), toXMLString() and etc.

列挙

- x = <a><b/><c/><d/>
- for (var i in x.*)
  - → i = "0", "1", "2"
- for each (var i in x.*)
  - → i = <b/>, <c/>, <d/>
  - 一般のオブジェクトにも使用可能

XML 名前空間

- Namespace 型
  - ns = new Namespace([[prefix,] uri])
  - x.ns::child, x.@ns::attr, x.@*::*
    - ns は最終的には文字列として評価される
- QName 型
  - qn = new QName([[ns,] name])
  - x.hasOwnProperty(qn)
  - x.ns::[name] == x[new QName(ns, name)]

デフォルト名前空間

- default xml namespace = ns;
  - ns をデフォルト名前空間に設定
- リテラル表記で作成される要素
  - <a/>.namespace() == ns
- 非修飾名の解決
  - x = <ns:a xmlns:ns={ ns }><ns:b/></ns:a>
    - x.ns::b[0] === x.b[0]
- QName オブジェクトの作成
  - new QName(name).uri == ns.uri

SpiderMonkey による拡張

- function 名前空間
  - XML [[Get]]/[[Put]] → Object [[Get]]/[[Put]]
    - x.name → <></>
    - x.function::name → function name() { }
  - html..*.(/h[1-6]$/.test(function::name()))
- AttributeName 型 / コンストラクタ
  - x = <a ns:attr="val" xmlns:ns={ ns }/>
  - x.hasOwnProperty(
    new AttributeName(ns, "attr"))

SpiderMonkey のバグ (1)

- 関数内でのデフォルト名前空間
  - default xml namespace = ns;
  - x = <a>DN
    - x.b → N
    - x.ns::b → D
- 内容にゴミが混じることがある
  - <{ "a" } attr="value">c
    - → >c

SpiderMonkey のバグ (2)

- 名前空間宣言が消失することがある
  - <b xmlns=""/>
    - → '<b/>'
  - x = <a/>; x.@ns::attr = "value";
    - → '<a ns:attr="value"/>'
- XML オブジェクトに対する列挙
  - for each (var i in <a><b/><c/><d/>)
    - → i = <b/>, <c/>, <d/>

DOMとの変換

- DOMノード → E4X XML
  - XMLSerializer → XML/XMLList
  - new XML(new XMLSerializer()
.serializeToString(node));
- E4X XML → DOMノード
  - DOMParser → importNode
  - document.importNode(new DOMParser()
.parseFromString(x.toXMLString(),
"application/xml").documentElement, true)

使用例 (1)

- Greasemonkey
  - 要素の作成
- userChrome.js
  - Overlay ファイルの動的生成
    - doc.loadOverlay("data:application/vnd.mozilla.xul+xml;charset=utf-8," + encodeURI(x.toXMLString()), null)

使用例 (2)

- Calendar (Sunbird/Lightning)
  - Google Calendar のフィードのパース
  - HTML エクスポート
- コンパイル成功 → 整形式の保証
- ヒアドキュメントとして
  - 改行も含められる
  - <>foo { bar }</>, <![CDATA[baz]]>

資料

- 仕様書
  - http://www.ecma-international.org/publications/standards/Ecma-357.htm
  - http://www.ne.jp/asahi/nanto/moon/specs/ecma-357.html (日本語？ 訳)
- 簡単！ 快適！E4X で Happy Firefox Life を！