

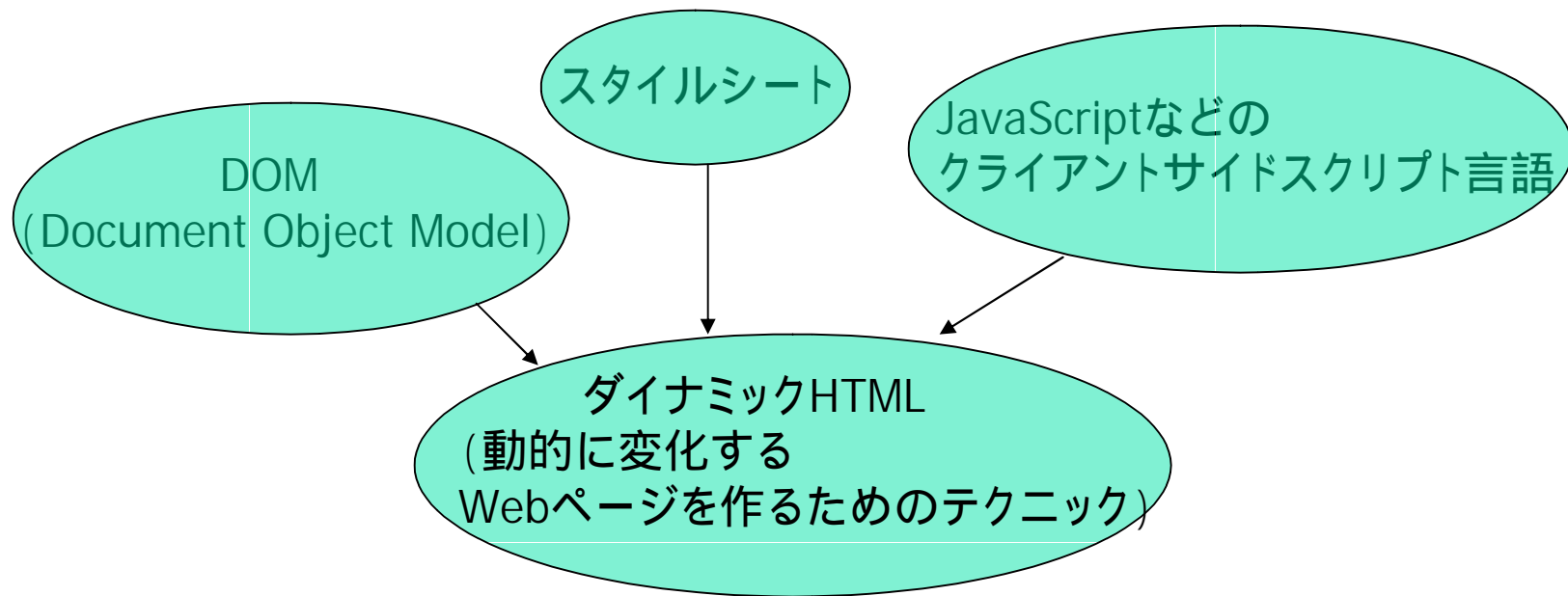


JavaScript第11回目

8-1 W3CによるDOMの 概要と基本操作

発表者 加藤 友宏
発表日 2006/12/22

ダイナミックHTMLとは？



Netscape NavigatorとInternet Explorerは、
ダイナミックHTMLに対応するため独自のDOMを実装

両者で同じように表示するのは、多大な労力が必要



W3CのDOMについて

- HTMLやXMLドキュメントにアクセスするためのAPI (アプリケーションプログラミングインターフェース)
 - 言語に依存しない仕様
 - OMG IDLで仕様自体が定義されている
 - JavaScript (ECMAScript) やJava用のバインディングが用意されている
 - 現在、「レベル1」、「レベル2」、「レベル3」の三段階のレベルがある



W3CのDOMレベル

拡張

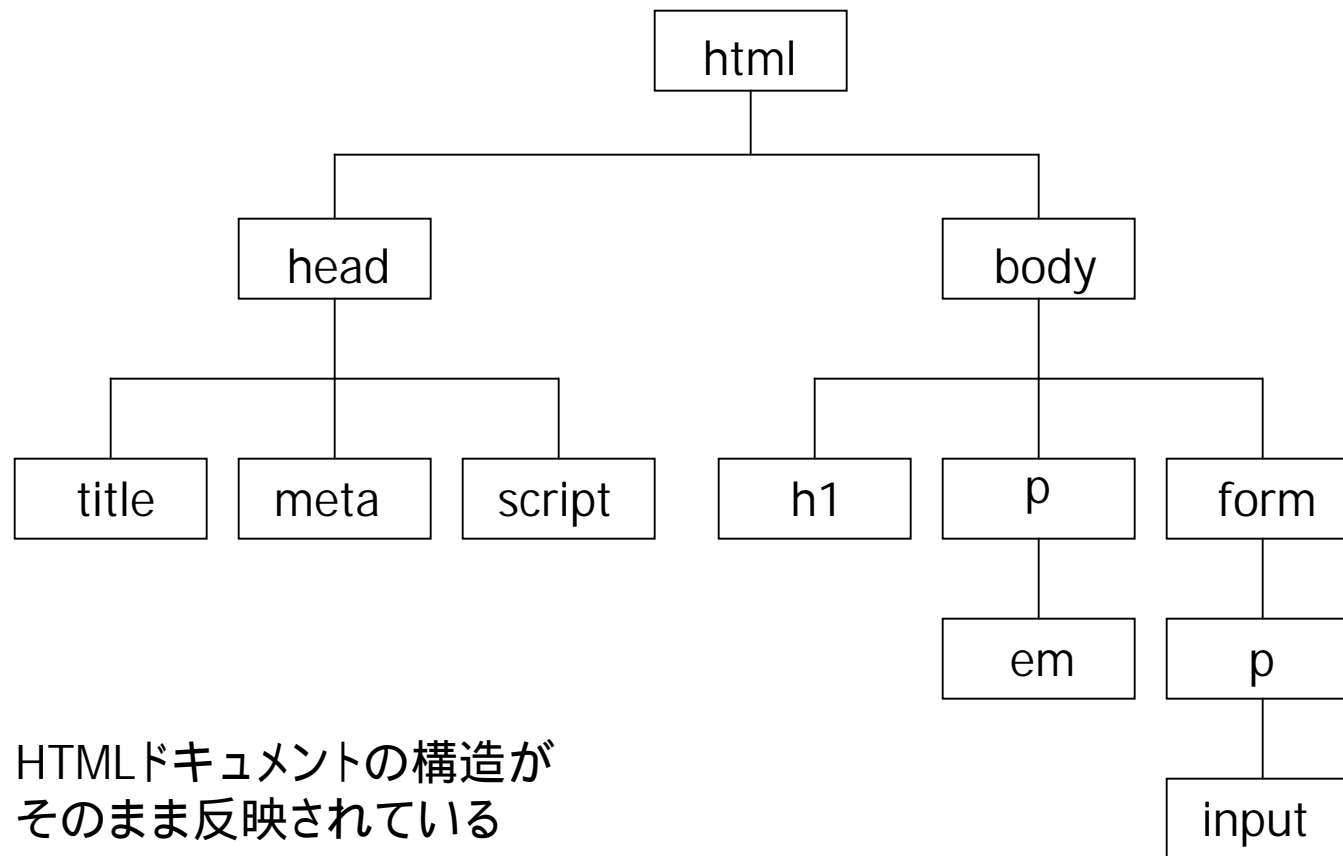
DOMのレベル	説明
レベル0	W3Cによって仕様が決められる前のDOM。 HTMLドキュメントの限られた要素のみ変更可能。
レベル1 (1998年)	「Core」というDOMを構成する要素についての汎用性のある定義と、それをHTML特有のオブジェクトメソッドに適応した「HTML」の定義。 「Java言語バインディング」、「ECMAScriptバインディング」
レベル2 (2000年)	「ビュー」、「スタイルシート」、「名前空間」、「イベント」、「検索」、「範囲」が追加。 要素の動的なポジショニングが可能。
レベル3 (2004年)	XML1.1に対応、XMLの読み書き機能など。



ドキュメント内の任意の要素にアクセス可能

- レベル0のDOM
 - フォームの要素やイメージなど限られたもののみ
- レベル1以降のDOM
 - ドキュメント内の個々の要素を「ノード」と呼び、JavaScriptなどを通してあらゆるノードにアクセス可能

サンプル1のDOMのツリー構造



HTMLドキュメントの構造が
そのまま反映されている

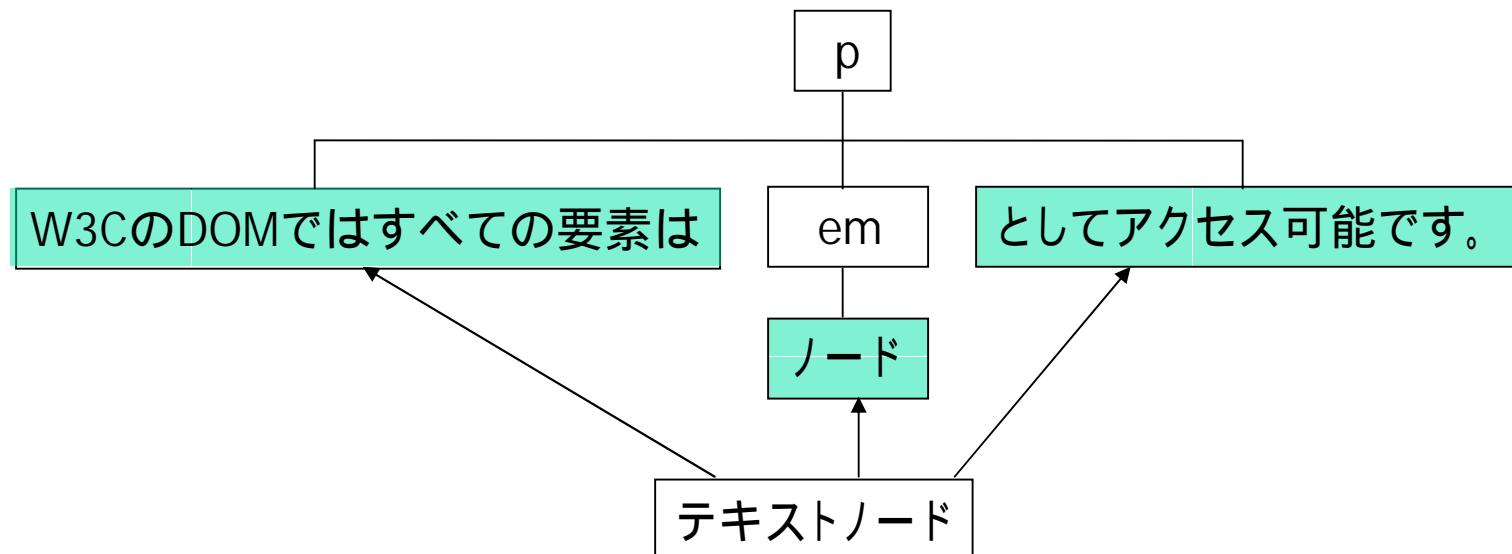
サンプル1

テキストノードとアトリビュート

- 「テキストノード」・・・Webブラウザで表示される文字列

例:

<p>W3CのDOMではすべての要素はノードとしてアクセス可能です。</p>



- タグのアトリビュートは、対応するノードのアトリビュートになる。



DOMツリーをたどるには？

- DOMツリーの親子関係をたどるために、各ノードに用意されているプロパティ

プロパティ	説明
childNodes	子ノードの配列
parentNode	親ノード
firstChild	最初の子ノード
lastChild	最後の子ノード
nodeName	ノード名
nodeValue	ノードの値
nodeType	ノードの種類
attributes	アトリビュートの配列



documentオブジェクトからたどってノードにアクセスする

- documentオブジェクトを起点に、下位の階層にたどる方法
(ブラウザによって若干ノードの階層が異なることに注意)

例: bodyエレメントのノードのnodeNameプロパティへのアクセス

1 : document.lastChild.childNodes[1].nodeName

↑
htmlエレメント

↑
bodyエレメント

2 : document.lastChild.lastChild.nodeName

3 : document.body.nodeName

↑
直接bodyエレメントの指定も可

nodeNameの値は、BODYとなる。(大文字になることに注意)



documentオブジェクトからたどってノードにアクセスする

- テキストノードの文字列自体は、nodeValueプロパティにある

例: h1要素の文字列「DOMツリー」にアクセス (Firefoxの場合)

```
1 : document.lastChild.lastChild.childNodes[1].firstChild.nodeValue
```

html要素 body要素

```
2 : document.body.childNodes[1].firstChild.nodeValue
```

注: Internet Explorerの場合は、以下のようなになる

```
document.body.childNodes[0].firstChild.nodeValue
```



idによるノードの取得

- HTMLタグのidアトリビュートの値から、目的のノードを取得できる。

メソッド	getElementById()
構文	document.getElementById(ID)
引数	ID-HTMLタグのidアトリビュートの値
説明	引数IDで指定されたノードを取得

- `<h1 id="firstHeader">DOMツリー</h1>`の文字列「DOMツリー」の取得方法(body要素の最初の要素の場合)

```
var node = document.getElementById("firstHeader");  
var str = node.firstChild.nodeValue;
```

- getElementByTagName()メソッドを使用しても可能

```
var node = document.getElementsByTagName("h1")[0];  
var str = node.firstChild.nodeValue;
```



ノードのプロパティや アトリビュートの変更

- W3CのDOMを利用すると、HTMLドキュメントがロードされた後で、テキストやグラフィックの内容や配置を変更可能

[サンプル2](#)

- アトリビュートの変更について
 - 個々のノードのアトリビュートの変更は、`setAttribute()`メソッドを使用

メソッド	<code>setAttribute()</code>
構文	<code>ノード.setAttribute(atr, val)</code>
引数	atr-アトリビュート名 val-設定する値
説明	ノードのアトリビュートの値を設定

[サンプル3](#)

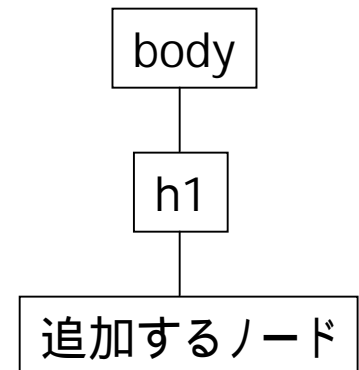
ノードの追加

- ノードの追加に関する主なメソッド

メソッド	説明
createElement()	新しくノードを作成する
createTextNode()	テキストノードを作成する
appendChild()	エレメントを子ノードとして追加する

例:HTMLドキュメントのボディ部の最後に、新たにh1エレメントを追加

```
var txt = document.createTextNode("追加するノード");  
var h1 = document.createElement("h1");  
h1.appendChild(txt);  
document.body.appendChild(h1);
```





ノードの削除

- 子ノードを削除するremoveChild()メソッドを使用

メソッド	removeChild()
構文	ノード.removeChild(node)
引数	node - 削除する子ノード
説明	ノードから指定した子ノードを削除する



innerHTMLプロパティの利用

- 各ノードのHTMLタグを含むテキストを動的に変更できる
- W3CのDOMの機能ではなく、Internet Explorerの拡張機能
- innerHTMLは、読み書き可能なプロパティ

innerHTMLとouterHTMLの違い

innerHTMLには、いくつかの仲間がある。

- ・innerHTMLは、そのノード自身のタグを含まない。
- ・outerHTMLは、そのノード自身のタグを含む。

Firefoxじゃ使えない...

[サンプル5](#)

[サンプル6](#)



DOMによるイベント処理

- 「イベントリスナー」と呼ばれる概念を使用。
- 手順としては...
 - (1) イベントが発生したときに呼び出される関数を定義する。
これが、「イベントリスナー」となる。
 - (2) どのイベントが発生したときに、
どの「イベントリスナー」が呼び出されるかを登録する。
- ただ、Internet Explorerとそれ以外のWebブラウザでは、イベントリスナーを登録するメソッドが違う...



DOMによるイベント処理

- Internet Explorer以外のブラウザのイベントリスナーを登録するメソッドは、addEventListener()メソッドを使用

メソッド	addEventListener()
構文	オブジェクト.addEventListener(event, func, capture)
引数	event-つかまえるイベント func-イベントリスナーとして登録する関数 capture-伝搬途中のイベントのキャプチャを行うか (ture or false)
説明	指定したイベントに対するイベントリスナーの登録

- Internet Explorerは、attachEvent()メソッドで同様のことが可能

```
attachEvent("on" + event, func);
```

メソッド名と引数capture以外は、addEventListener()メソッドと同じ



練習問題

- DOMを利用してページを作成してください。
 - 表示するものすべてをDOMを使って作成すること。
 - 表示させるものは、何でもいいです。ただし...
 - 何かしらの文字列
 - ボタン
 - 何かしらのイベント処理
 - 画像の表示
 - ノードの削除
 - を含めてください。
 - その他は、ご自由に