

プログラミングキャンプ

- 1) 課題のテーマ:XMLについて
- 2) 開始日:5月5日
- 3) 終了日:5月5日
- 4) 締切日:5月5日
- 5) 作成者:三上 祥一郎

XMLに関する課題

1. 課題(暫定)

- Yahooショッピング APIを使用し, XMLドキュメントを取得しなさい

2. 課題背景

- ECサイトを作成する際に非常に苦勞したため

3. 背景理由

- Javaで書かれたサンプルコードが存在しないため (PHPのみ)
- 取得形式をXMLにした場合, それらの勉強が追加で必要なため
- PHPコードの読みこみからXMLドキュメントの取得, 画面表示までに時間が掛かったため (約3日以上)

上記の理由から, ECサイト作成とは別にXMLドキュメント取得の課題を作成しました.

JSONIC等が導入出来ればJSON形式で取得可能だと思います. その方が簡単だと思いますが, 私は出来ませんでした(^q^)

その際は[このサイト](#)が参考になると思います.

XML(Extensible Markup Language)について

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <department>
  - <employee>
    <name>John Doe</name>
    <job>Software Analyst</job>
    <salary>2000</salary>
  </employee>
  - <employee>
    <name>Jane Fletcher</name>
    <job>Designer</job>
    <salary>2500</salary>
  </employee>
</department>
```

1. 目的

- データを記述, 保存, 交換すること

2. 特徴

- タグ名を自由に決められる
- XMLを使用するには「XMLパーサ」というツールが必要

3. XMLが用いられている分野

- 電子商取引分野
- 企業内システム

※上記情報元「[5分で分かるXML入門](#)」

※5分で読みきれませんでした(へへ)

図1. XMLの例([adobeホームページ](#)より引用)

XML(Extensible Markup Language)について

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <department> } 親子関係
- <employee> }
    <name>John Doe</name> } 親子関係
    <job>Software Analyst</job> }
    <salary>2000</salary> }
  </employee>
- <employee> }
    <name>Jane Fletcher</name> } 兄弟関係
    <job>Designer</job> }
    <salary>2500</salary> }
  </employee>
</department>
```

図2. XMLの構成図の例

1. 構成について

- XMLは各要素(Element)で構成されている.
- <employee>は<department>の子要素
- <name>は<employee>の子要素
 - ※<job>, <salary>も同様
- <name>は<job>, <salary>の兄弟要素

2. 情報について

- <name>が持っている情報
[John Doe], [Jane Fletcher]
- <job>が持っている情報
[Software Analyst], [Designer]
- <salary>が持っている情報
[2000], [2500]

JAVAとXMLについて 1/3

*url*入力からXMLドキュメント(*doc*)取得まで

InputStream *is* = new FileInputStream(*url*); ①

DocumentBuilderFactory *dbf*
= new DocumentBuilderFactory(); ②

DocumentBuilder *db*
= *dbf*.newDocumentBuilder(); ③

Document *doc* = *db*.parse(*is*); ④

1. 使用するAPI

- javax.xml.parsers.*;
- org.w3c.dom.Document;
- ※注意) 誤ってjavax.swing.text.Documentをインポートしない様にしてください
- org.w3c.dom.*;

2. コードの補足

- ① *url*には接続先のURL(String型)を代入する
- ②, ③は一行にまとめることも可能
- ④でXMLドキュメントを取得している

図3. コード例1

JAVAとXMLについて 2/3

全体の要素(*department*)取得から
特定の要素(*name*)の情報取得まで

Element *department* = *doc*.getDocumentElement(); ⑤

Element *employee*
= findChildByTag(*department*, "employee"); ⑥

Element *name*
= findChildByTag(*employee*, "name"); ⑦

String *nameValue*
= *name*.getTextContent(); ⑧

1. 使用するAPI

- org.w3c.dom.Element

※org.w3.dom.*;が記述されていれば, 上記は必要ありません

2. 特有メソッド(findChildByTag)について

- 次ページで説明します

3. コードの補足

- ⑤*doc*から*department*要素を取得

- ⑥, ⑦では親要素から子要素を取得している

- ⑧では要素の情報を取得している

図4. コード例2

JAVAとXMLについて 3/3

findChildByTagメソッド

```
static Element findChildByTag
(Element el, String name) throws Exception {
    NodeList children = el.getChildNodes();
    for(int i = 0; i < children.getLength(); i++){
        if(children.item(i) instanceof Element){
            Element e = (Element)children.item(i);
            if(e.getTagName().equals(name)){
                return e;
            }
        }
    }
    return null; }
```

図5. コード例3

1. 使用するAPI

- org.w3c.dom.NodeList

※org.w3.dom.*;が記述されていれば, 上記は必要ありません

2. メソッドの補足

- これは親要素から特定の子要素を取り出すためのメソッドです

3. 情報元について

- コード例1～3は「[Java入門実践編](#)」のp297のコードを参考にしています