

# 知識プログラミング方法論

## 第9回：XMLの基礎

ライフスタイルデザイン研究センター 金井秀明

1

## XMLの基礎

2

### XMLとは？#1

- XML(eXtensible Markup Language)
  - SGML(Standard Generalized Markup Language,1985)をWeb上で利用できるようにした.
  - 1996年にDraft発表
- 背景：
  - 異なるコンピュータシステム間で、文書交換ができないことが、非常に問題だった.

3

3

### XMLとは？#2

- Markup Languageへの要求（60年代後半, IBM, Chales Goldfarbらによる）
  - 物理的なレイアウトでなく、文書の構造を反映したもの.
  - コンピュータと人間の両方が理解可能な形式でなければならない.

4

4

## XMLとは？#3

### ・ SGMLの例

- ・ タグ<~>で文書を構造化
- ・ タグはDTD(Document Type Definition)で定義される.

```
<PERSON>
<FIRSTNAME>Hideaki</FIRSTNAME>
<LASTNAME>Kanai</LASTNAME>
</PERSON>
```

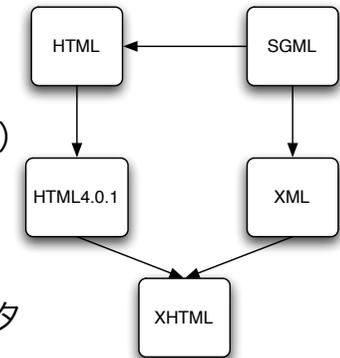
5

5

## XMLとは？#4

### ・ HTMLとXML

- ・ HTMLは、情報（データ）の「表示」（Webページの記述）を目的としたもの.
- ・ XMLは、情報（データ）の「構造や意味」（情報やデータの記述）を目的としたもの.



6

6

## XMLとは？#5

### ・ XMLの特徴

- ・ アプリケーションによって処理可能
- ・ 独自にマークアップ言語（タグ）が定義可能
- ・ 構造化フォーマットされたテキスト形式の文書
- ・ OSやネットワークなどのプラットフォームに非依存

7

7

## XML文書の例I

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<cars>
  <car>
    <name>VW Golf</name>
    <価格>300</価格>
  </car>
  <car>
    <name>VW Polo</name>
    <価格>250</価格>
  </car>
</cars>
```

8

8

## XML文書の基本構成

- XML文書の構成要素
  - XML宣言
  - XMLデータ
  - DTD(Document Type Definition,文書型定義)

9

## XML文書の種類

- XML文書には以下の2種類がある.
  - 整形XML文書 (well-formed XML document)
  - 妥当なXML文書(valid XML document)

10

## 整形XML文書

XML宣言

```
<?xml version="1.0" encoding="Shift_JIS" ?>  
<!-- XML文書の例 sample1.xml -->
```

XMLデータ

```
<cars>  
<car>  
  <name>VW Golf</name>  
  <価格>300</価格>  
</car>  
<car>  
  <name>VW Polo</name>  
  <価格>250</価格>  
</car>  
</cars>
```

11

## XML宣言

- XMLのversionとXML文書で使う文字コードの宣言, コメント文,
  - 文字コード : UTF-8, UTF-16, Shift\_JIS, EUC-JP,...

```
<?xml version="1.0" encoding="Shift_JIS" ?>  
<!-- XML文書の例 sample1.xml -->
```

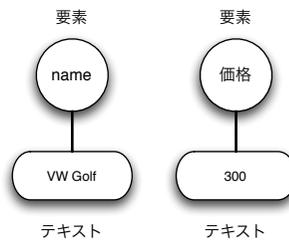
12

12

## XMLデータ：要素#1

- 要素：<要素名>内容</要素名>
- <要素名>内容</要素名>
- 要素名：英数字，漢字OK，空白はNG，最初の文字が数字はNG.

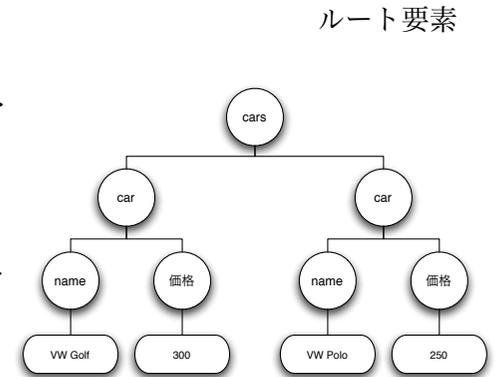
```
<name>VW Golf</name>
<価格>300</価格>
```



13

## XMLデータ：要素#2

```
<cars>
<car>
  <name>VW Golf</name>
  <価格>300</価格>
</car>
<car>
  <name>VW Polo</name>
  <価格>250</価格>
</car>
</cars>
```



14

## XMLデータ：属性

- 属性：要素に付属するデータ
- <要素名 属性名  
1="属性値1", 属性名  
2="属性値2", ... >内容  
</要素名>

```
<cars>
<car id="1" country="ドイツ">
  <name>VW Golf</name>
  <価格>300</価格>
</car>
<car id="2" country="ドイツ">
  <name>VW Polo</name>
  <価格>250</価格>
</car>
</cars>
```

15

## 妥当なXML文書

XML宣言

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!-- XML文書の例 sample2.xml -->
```

DTD

```
<!DOCTYPE cars [
  <!ELEMENT cars (car+)>
  <!ELEMENT car (name, 価格)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT 価格 (#PCDATA)>
]>
```

XMLデータ

```
<cars>
<car>
  <name>VW Golf</name>
  <価格>300</価格>
</car>
<car>
  <name>VW Polo</name>
  <価格>250</価格>
</car>
</cars>
```

valid\_cars1.xml

16

## DTD：文書型定義#1

- Document Type Definition: DTD: 要素, 属性, 実体, 記法など, XMLの構造を規定するもの.

- XMLのschema言語である.

- 文書型宣言 

```
<!DOCTYPE cars[
```
- 要素型宣言 

```
<!ELEMENT cars (car+)>
```
- 属性リスト宣言 

```
<!ELEMENT car (name, 価格)>
```
- 実体宣言 

```
<!ELEMENT name (#PCDATA)>
```
- 実体宣言 

```
<!ELEMENT 価格 (#PCDATA)>
```

17

## DTD：文書型定義#2

- 文書型宣言：XML文書本体の構造を定義

```
<!DOCTYPE ルート要素名[
```

```
...  
...  
]>
```

- 要素型宣言：XML文書本体の各要素の定義. 要素名と内部モデルからなる.

```
<!ELEMENT 要素名 (内部モデル)>
```

18

## 内部モデル#1

- 要素やテキストの

- 出現順序の指定する.

```
<!ELEMENT 要素名 (要素名1, 要素名2, ...)>
```

- いずれかが出現するように指定する.

```
<!ELEMENT 要素名 (要素名1 | 要素名2 | ...) >
```

19

## 内部モデル#2

- 要素やテキストの

- 出現回数を指定する.

符号	意味
なし	必ず1回出現する
+	1回以上出現する
*	0回以上出現する
?	0回 or 1回出現する

20

## DTD : 文書型定義#3-1

- ・ 属性リスト宣言
- ・ 属性の定義

```
<!DOCTYPE cars[
  <!ELEMENT cars (car+)>
  <!ELEMENT car (name, 価格)>
  <!ATTLIST car id CDATA #REQUIRED>
  <!ATTLIST car country (ドイツ|日本) "ドイツ">
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT 価格 (#PCDATA)>
]>
<cars>
<car id="1" country="ドイツ">
  <name>VW Golf</name>
  <価格>300</価格>
</car>
<car id="2">
  <name>VW Polo</name>
  <価格>250</価格>
</car>
</cars>
```

valid\_cars2.xml

21

## DTD : 文書型定義#3-2

- ・ 属性リスト宣言
- ・ 属性の定義

<!ATTLIST 要素名 属性名 属性値の候補 デフォルトの指定>

- ・ 例

<!ATTLIST car id CDATA #REQUIRED>

<!ATTLIST car country (ドイツ|日本) "ドイツ">

22

## DTD : 文書型定義#4-1

- ・ 実体宣言
- ・ 文字列参照
- ・ ファイル参照

```
<!DOCTYPE cars[
  <!ELEMENT cars (car+)>
  ...
  <!ATTLIST car country CDATA #REQUIRED>
  <!ENTITY g "ドイツ">
  <!ENTITY other SYSTEM "other.xml">
  ...
]>
<cars>
<car id="1" country="ドイツ">
  <name>VW Golf</name>
  <価格>300</価格>
</car>
<car id="2" country="&g;" >
  <name>VW Polo</name>
  <価格>250</価格>
</car>
&other;
</cars>
```

23

## DTD : 文書型定義#4-2

- ・ 実体宣言
- ・ 文字列 (実体) <!ENTITY 実体名 "文字列">
- ・ ファイル <!ENTITY 実体名 SYSTEM "ファイルの場所">
- ・ 実体参照 &実体名

24

## 名前空間#1

- DTDによって、柔軟にタグを定義できる.
- 一方、複数のXML文書間で、タグが重複したりすると、混乱が生じる.

誰がどこで定義したタグなのかを  
明確にする必要がある.

25

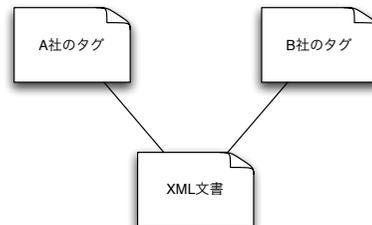
## 名前空間#2

- 名前空間の役割としては,
  - XML文書のコンテンツを正確に共有・交換することを促進できる.
  - 規格化されたタグセットを組み合わせることがで、相互運用性が高まる.
    - グラフはSVG、数式はMathMLで...

26

## 名前空間#3

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!-- XML文書 sample7.xml -->
<sale xmlns:A="http://books.softbank.co.jp/ACompany"
      xmlns:B="http://books.softbank.co.jp/BCompany">
  <A:car>
    <A:name>乗用車</A:name>
    <A:price>150</A:price>
  </A:car>
  <B:name>
    タクシー社
  </B:name>
  <amount>2</amount>
</sale>
```



27

## XML Schema

- XML文書向けのスキーマ言語
- DTDよりも厳密な記述ができる.

28

## DTDの問題点

- DTD自体をXML向けのツールで解析できない。  
DTDはXMLでないため。
- 記述力について
  - XMLの構造を定義できるが、型 (string, decimal, float, boolean,...) を指定できない。
  - 出現回数を「\*\*\*回まで」という記述ができない。

29

## XMLの例

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<cars>
  <car>
    <name>乗用車</name>
    <価格>150</価格>
  </car>
</cars>
```

30

## DTDの例

```
<!DOCTYPE cars [
  <!ELEMENT cars (car+)>
  <!ELEMENT car (name, 価格)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT 価格 (#PCDATA)>
]>
```

31

## XML Schemaの例

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="cars">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="car">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="name" type="xsd:string">
              <xsd:element name="価格" type="xsd:decimal">
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

32

## 演習：教科書

- ・「20ページ練習2」と「56ページ練習5」をして、作成したXML文書をブラウザで確認してください。
- ・作成したXML文書を妥当なXML文書として作成してください。

## まとめ

- ・XMLの基本構成
  - ・XML宣言, XMLデータ, DTD
  - ・名前空間