

Slide URL

<https://vu5.sfc.keio.ac.jp/slide/>

Web情報システム構成法
No.12 その他のWeb技術

萩野 達也 (hagino@sfc.keio.ac.jp)

これまで

- ▶ Web全般
 - ▶ Webの歴史
 - ▶ Webアーキテクチャ
 - ▶ Webサーバ

- ▶ Webページ
 - ▶ HTML
 - ▶ CSS
 - ▶ JavaScript
 - ▶ フォームインタラクション
 - ▶ XML

画像

▶ Webページでの画像の利用

▶ img要素

```

```

▶ 画像形式

▶ GIF

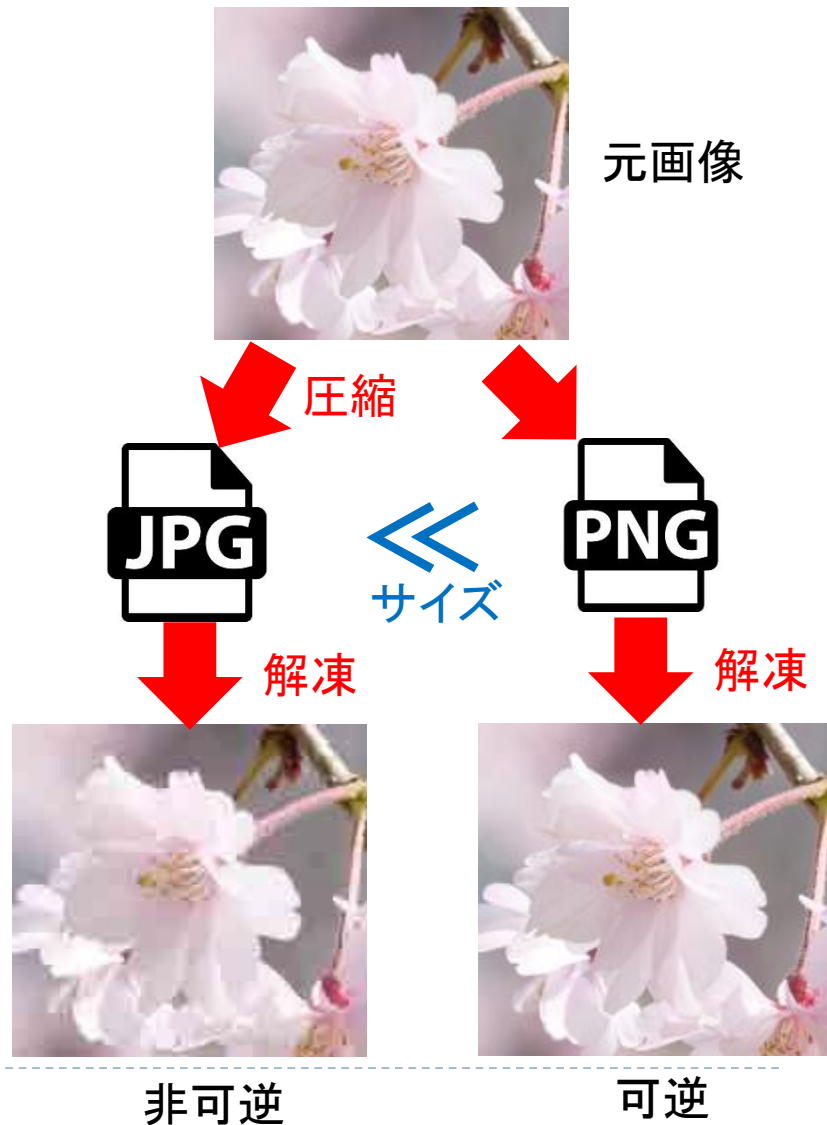
- ▶ Graphics Interchange Format
- ▶ 可逆圧縮
- ▶ 256色(インデックスカラー)
- ▶ 透過あり

▶ PNG

- ▶ Portable Network Graphics
- ▶ 可逆圧縮
- ▶ PNG-8: 256色インデックスカラー
- ▶ PNG-24: フルカラー(RGB8bit)
- ▶ PNG-32: アルファ値(透過度)あり

▶ JPEG

- ▶ Joint Photographic Experts Group
- ▶ 非可逆圧縮
- ▶ フルカラー



JPEGの圧縮技術

▶ DCT(離散コサイン変換)

- ▶ 時系列データを周波数成分に分ける

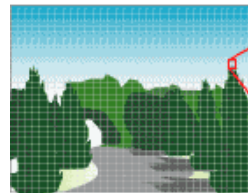
▶ 音声の圧縮

- ▶ MP3, AAC, ATRAC
- ▶ 音声をサンプリング
- ▶ 時系列データをDCTで周波数成分に
- ▶ 高音部分は削除する

▶ 画像の圧縮

- ▶ 色も縦横方向に変化するデータで時系列のように扱うことができる
- ▶ 2次元のDCTで縦横の周波数成分に
- ▶ 高周波部分は削除する

原画(静止画)を
8×8個の点から成る
正方形ブロックに分解



ブロックごとに周波数
に変換

150	146	142	138	134	130	126	122
148	144	140	136	132	128	124	120
142	138	134	130	126	122	118	114
132	128	124	120	116	112	108	104
118	114	110	106	102	98	94	90
100	96	92	88	84	100	96	92
78	74	70	66	62	78	74	70
52	48	44	40	36	52	48	44

* DCT 演算した係数

830	48	10	11	-7	5	4	-4
230	28	-11	-4	8	-3	-5	6
-62	-11	4	2	-3	1	2	-2
30	-4	2	1	-1	0	1	-1
-23	8	-3	-1	2	-1	-1	2
10	-3	1	0	-1	0	0	-1
-1	-5	2	1	-1	0	1	-1
-3	6	-2	-1	2	-1	1	1

高周波成分と低周波成分に分けます

→ 低周波成分 (数字が高くなります)

→ 高周波成分 (「0」に近い数字になります)

★ 高周波成分を「0」とみなすことで
情報量削減

→ 低周波成分のみ、ジグザグに読み取り、01信号に変換します。
「010011010101111011010101 ~」

* DCT 演算した係数は青色と赤色の差分、緑色と赤色の差分、明るさを計算した結果の数字です。

<https://www.fujitsu.com/jp/group/labs/about/resources/tech/techguide/list/image-compression/>

画像形式の正しい選択

▶ GIFおよびPNG

- ▶ イラスト
- ▶ アイコン
- ▶ グラフ
- ▶ はっきりと色が変わっているデータ
- ▶ 写真には不向き

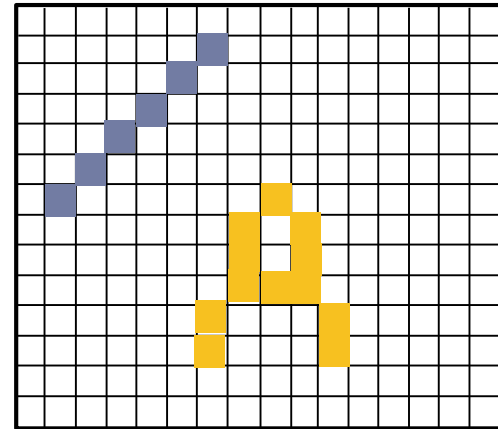
▶ JPEG

- ▶ 写真
- ▶ 色がだんだんと変化しているデータ
- ▶ イラストなどエッジをはっきり出すのには不向き
- ▶ 論文の図には使ってはいけない

ビットマップ画像とベクトル画像

▶ ビットマップ画像

- ▶ 縦と横のピクセル数が決まっている
- ▶ ピクセル単位で色を決める
- ▶ 文字もピクセルで表している



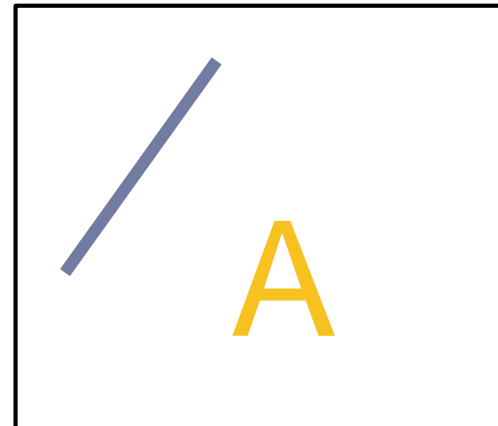
Photoshop



ビットマップ画像の
編集

▶ ベクトル画像

- ▶ 描画要素の集まりで表す
 - ▶ 四角
 - ▶ 直線
 - ▶ 円
 - ▶ ベジエ曲線
 - ▶ 文字



Illustrator



ベクター画像の
編集

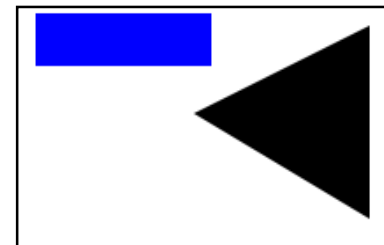
SVG

- ▶ Scalable Vector Graphics (SVG) 1.1 (Second Edition)
 - ▶ 2011年8月16日勧告
- ▶ Scalable Vector Graphics (SVG) Tiny 1.2
 - ▶ 2008年12月22日勧告
- ▶ Webにおけるベクター形式のグラフィックス
 - ▶ 図形の組み合わせで表現
 - ▶ 拡大・縮小が可能

```
<?xml version="1.0" ?>
<svg xmlns="http://www.w3.org/2000/svg"
  width="300"
  height="300">
  <text style="fill:red;" y="2cm">This is SVG.</text>
</svg>
```

This is SVG.

```
<?xml version="1.0" ?>
<svg xmlns="http://www.w3.org/2000/svg"
  width="300"
  height="300">
  <rect style="fill:blue" width="100" height="30" x="10" y="3"/>
  <path d="M 100 60 L 200 10 L 200 120 z"/>
</svg>
```



SVG in HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <title>SVG Example</title>
  </head>
  <body>
    <svg width="500" height="200">
      <rect style="stroke:black;fill:none" width="500" height="200" x="0" y="0" />
      <path style="fill:red" d="M 100 60 L 400 150 L 20 50 L 300 150 L 50 30 L 150 40 z" />
      <path style="fill:#88ff88" d="M 100 160 L 420 50 L 200 150 L 350 50 L 150 130 L 40 180 z" />
      <circle cx="100" cy="120" r="20" style="fill:#8888ff" />
      <circle cx="400" cy="150" r="20" style="fill:#ff88ff" />
      <text style="fill:#8822ff" x="150" y="130" font-family="Times" font-size="100">SFC</text>
    </svg>
  </body>
</html>
```



HTML5で追加された画像関連要素

▶ canvas

- ▶ JavaScriptにより描画できる2Dグラフィックス
- ▶ Flashによらないアニメーションの作成
- ▶ ゲームなども作成可能

▶ video

- ▶ YouTubeなどでの動画の普及
- ▶ Flashなどに特定されない動画
- ▶ JavaScriptから制御可能
- ▶ ビデオ形式はブラウザ依存

▶ audio

- ▶ 音声
- ▶ JavaScriptから制御可能
- ▶ 音声形式はブラウザ依存



<http://www.html5.jp/canvas/what.html>

video要素

▶ これまで

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="425" height="344"
  codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,40,0">
  <param name="allowFullScreen" value="true" />
  <param name="allowscriptaccess" value="always" />
  <param name="src" value="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&" />
  <param name="allowfullscreen" value="true" />
  <embed type="application/x-shockwave-flash" width="425" height="344"
    src="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&"
    allowscriptaccess="always" allowfullscreen="true">
  </embed>
</object>
```

▶ video要素

```
<video width="640" height="360"
  src="http://www.youtube.com/demo/google_main.mp4" controls autobuffer>
  <p> Try this page in Safari 4! Or you can
  <a href="http://www.youtube.com/demo/google_main.mp4">download
  the video</a> instead.</p>
</video>
```

MathML

▶ Mathematical Markup Language (MathML) Version 3.0

- ▶ 2010年10月21日勧告
- ▶ 数式をWebで用いる
- ▶ 数式の記述だけでなく、意味を表すことができる

▶ Presentation Element

- ▶ 30の要素
- ▶ 約50の属性
- ▶ 数学記法をコーディング
 - ▶ 例: `mfrac`は分数を表す
- ▶ 数式の表示を思うように表すことができる
- ▶ 見た目だけになり、音声ブラウザなどで読み上げると分けが分からないものになることも多い

▶ Content Element

- ▶ 約100の要素
- ▶ 約12の属性
- ▶ 数学の関数や演算子を表す
 - ▶ 例: `plus`, `sin`
- ▶ 数学のオブジェクトを表すものもある。
 - ▶ 例: `set`, `vector`
- ▶ 表示以外の目的で使う: 音声ブラウザ, 数式処理
- ▶ 表示がどのようになるかコントロールするのが難しい

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
<msup>
  <mfenced>
    <mi>a</mi>
    <mo>+</mo>
    <mi>b</mi>
  </mfenced>
  <mn>2</mn>
</msup>
```

$(a + b)^2$

```
<apply>
  <power/>
  <apply>
    <plus/>
    <ci>a</ci>
    <ci>b</ci>
  </apply>
  <cn>2</cn>
</apply>
```

$(a + b)^2$

HTMLに数式を埋め込む

```
<?xml version="1.0" encoding="iso-2022-jp"?>
<?xml-stylesheet type="text/xsl" href="mathml.xsl"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>MathML Test</title></head>
<body>
<p>
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <msup>
      <mfenced>
        <mi>a</mi>
        <mo>+</mo>
        <mi>b</mi>
      </mfenced>
      <mn>2</mn>
    </msup>
  </math>
</p>
</body>
</html>
```

XMLの活用

- ▶ XMLの表示形式
 - ▶ XSL
- ▶ DTDに代わるもの
 - ▶ XML Schema
- ▶ ハイパーリンクの一般形
 - ▶ XLink
- ▶ XML形式での表現
 - ▶ 動画編集
 - ▶ プロトコル記述
 - ▶ その他

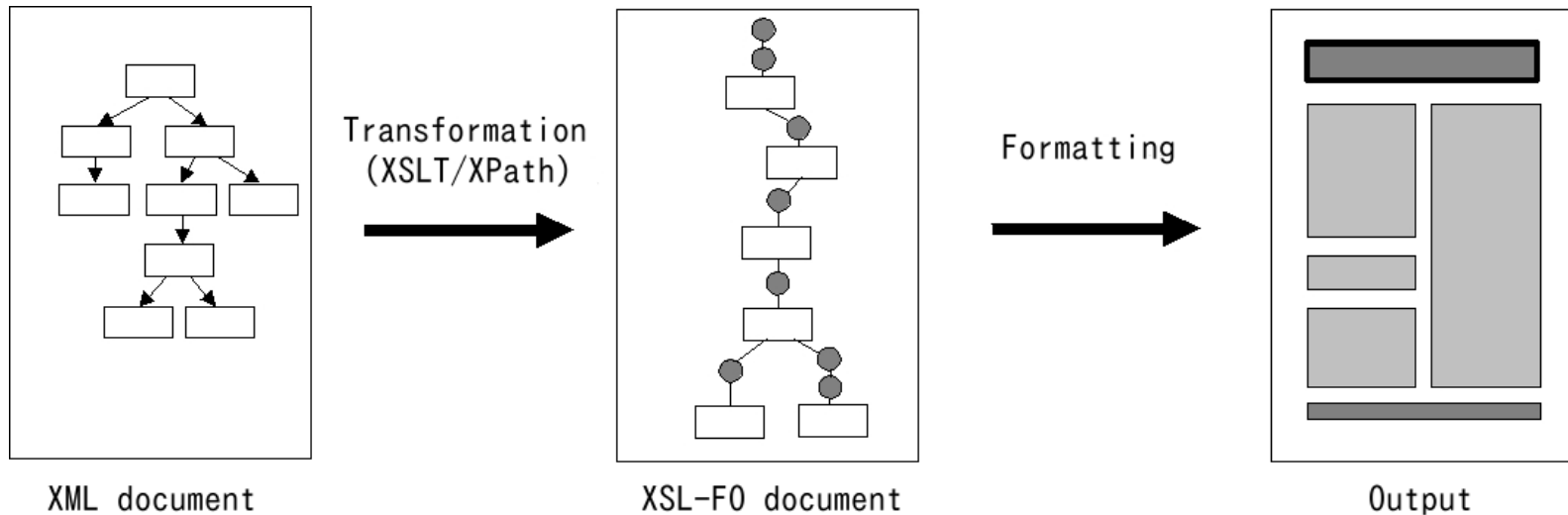
XML文書のスタイル

- ▶ HTMLのスタイル指定
 - ▶ CSSによる指定
 - ▶ element毎に属性と値の対を与える単純なもの

- ▶ XMLのスタイル指定
 - ▶ 単純なCSSでは不十分
 - ▶ elementが自由に定義できるため、それに対処しなくてはいけない
 - ▶ 単にscreen上でブラウザできるだけでなく、紙に印刷するなども行いたい
 - ▶ flexibleにするにはかなりの記述能力が必要
 - ▶ SGMLにはDSSSLがあった
 - ▶ DSSSLはLISPを基本とする
 - ▶ WEBではできればdeclarativeをくずしたくない

XSL

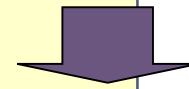
- ▶ XSL Transformations (XSLT) Version 2.0
 - ▶ 2007年1月23日 W3C勧告
- ▶ Extensible Stylesheet Language (XSL) Version 1.1
 - ▶ 2006年12月5日 W3C勧告



XSLTの例

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/著者">
    <html>
      <head>
        <title>著者</title>
      </head>
      <body bgcolor="white">
        <b>
          <xsl:value-of select="名前" />
        </b>
        <br />
        <xsl:value-of select="所属" />
        <br />
        <i>
          <xsl:value-of select="電子メール" />
        </i>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<著者>
  <名前>萩野 達也</名前>
  <所属>慶應義塾大学</所属>
  <電子メール>hagino@sfc.keio.ac.jp</電子メール>
</著者>
```



```
<html>
  <head><title>著者</title></head>
  <body bgcolor="white">
    <b>萩野 達也</b><br/>
    慶應義塾大学<br/>
    <i>hagino@sfc.keio.ac.jp</i>
  </body>
</html>
```


XMLの文書形式DTDは分かりにくい

▶ 電子メールのDTD

```
<!ELEMENT 電子メール (ヘッダー, 本文)>
<!ELEMENT ヘッダー (送信者, 宛先+, cc*, 見出し)>
<!ELEMENT 送信者 EMPTY>
<!ATTLIST 送信者
  名前 CDATA #IMPLIED
  アドレス CDATA #REQUIRED>
<!ELEMENT 宛先 EMPTY>
<!ATTLIST 宛先
  名前 CDATA #IMPLIED
  アドレス CDATA #REQUIRED>
<!ELEMENT cc EMPTY>
<!ATTLIST cc
  名前 CDATA #IMPLIED
  アドレス CDATA #REQUIRED>
<!ELEMENT 見出し (#PCDATA)>
<!ELEMENT 本文 (テキスト, 添付ファイル*)>
<!ELEMENT テキスト (#PCDATA)>
<!ELEMENT 添付ファイル EMPTY>
<!ATTLIST 添付ファイル
  エンコード (mime|binhex) "mime"
  ファイル CDATA #REQUIRED>
```

XML Schema

- ▶ XML自身で書かれたDTD
- ▶ より詳しい制約を記述可能

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">

  <element name="電子メール" type="電子メールの型" />

  <complexType name="電子メールの型">
    <sequence>
      <element name="ヘッダー" type="ヘッダー型" />
      <element name="本文" type="本文型" />
    </sequence>
  </complexType>

  . . . .

</xsd:schema>
```

XML Schema (例：電子メール)

```
<complexType name="ヘッダ型">
  <sequence>
    <element name="送信者" type="名前とアドレス" />
    <element name="宛先" type="名前とアドレス" minOccurs="1" maxOccurs="unbounded" />
    <element name="cc" type="名前とアドレス" minOccurs="0" maxOccurs="unbounded" />
    <element name="見出し" type="string" />
  </sequence>
</complexType>

<complexType name="名前とアドレス">
  <attribute name="名前" type="string" use="optional" />
  <attribute name="アドレス" type="string" use="required" />
</complexType>

<complexType name="本文型">
  <sequence>
    <element name="テキスト" type="string" />
    <element name="添付" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <attribute name="エンコード" use="default" value="mime">
          <simpleType>
            <restriction base="string">
              <enumeration value="mime" />
              <enumeration value="binhex" />
            </restriction>
          </simpleType>
        </attribute>
        <attribute name="ファイル" type="string" use="required" />
      </complexType>
    </element>
  </sequence>
</complexType>
```

XLink

- ▶ XML Linking Language (XLink) Version 1.0
 - ▶ 2001年6月27日勧告
- ▶ 目的
 - ▶ 2つ以上のリソース間の関係を表す
 - ▶ リンクにメタデータを関係付ける
 - ▶ リンクだけを分離して外部に置くことができる

```
<my:crossReference
  xmlns:my="http://example.com/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="simple" xlink:href="students.xml"
  xlink:role="http://www.example.com/linkprops/studentlist"
  xlink:title="Student List"
  xlink:show="new"
  xlink:actuate="onRequest">
  ...
  Current List of Students
  ...
</my:crossReference>
```

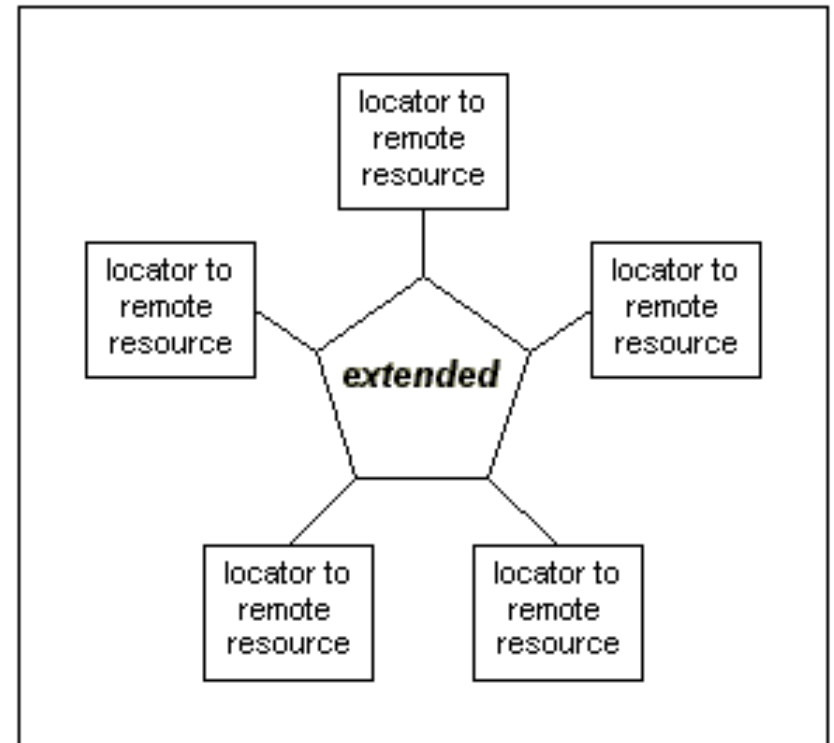
XLinkの種類

▶ Simple link

- ▶ 2つのリソースの関係のみ
- ▶ HTMLのA要素やIMG要素と同じ

▶ Extended link

- ▶ 3つ以上のリソースの関係を記述できる



Extended Linkの例

```
<?xml version="1.0"?>
<performers xmlns:xlink="http://www.w3.org/1999/xlink">
  <item xlink:type="extended">
    <!-- link definition (local) -->
    <link xlink:type="resource" xlink:label="overview" xlink:title="Information on Sinatra">
      Frank Sinatra</link>

    <!-- link definitions (remote) - Sinatra's biography, songs and articles -->
    <link xlink:type="locator" xlink:href="bio.xml" xlink:label="bio" xlink:title="Biography" />
    <link xlink:type="locator" xlink:href="songs.xml" xlink:label="songs" xlink:title="Songs"/>
    <link xlink:type="locator" xlink:href="press.xml" xlink:label="press" xlink:title="Press articles" />

    <!-- local to remote arc - from name to biography -->
    <arc xlink:type="arc" xlink:from="overview" xlink:to="bio" xlink:show="replace"
      xlink:actuate="onRequest" />

    <!-- remote to remote arc - from biography to song list -->
    <arc xlink:type="arc" xlink:from="bio" xlink:to="songs" xlink:show="replace"
      xlink:actuate="onRequest" />

    <!-- remote to remote arc - from biography to press archive -->
    <arc xlink:type="arc" xlink:from="bio" xlink:to="press" xlink:show="replace"
      xlink:actuate="onRequest" />

  </item>
</performers>
```

SMIL

▶ Synchronized Multimedia Integration Language 3.0

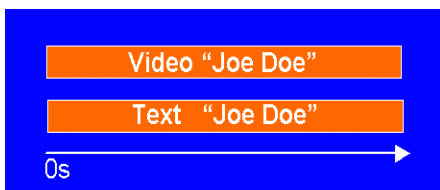
▶ 2008年12月1日勧告

▶ 目的

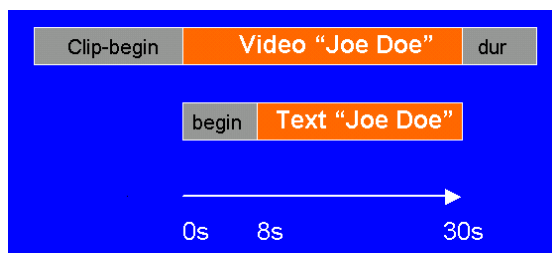
- ▶ ビデオや音声の同期をHTMLのように記述したい
- ▶ ビデオフォーマットについては規定しない



```
<seq>  
  <video src="media/joe-video"/>  
  <text src="media/joe-text"/>  
</seq>
```



```
<par>  
  <video src="media/joe-video"/>  
  <text src="media/joe-text"/>  
</par>
```



```
<par>  
  <video clip-begin="20s" dur="30s"  
    src="media/joe-video"/>  
  <text begin="8s" fill="freeze"  
    src="media/joe-text"/>  
</par>
```

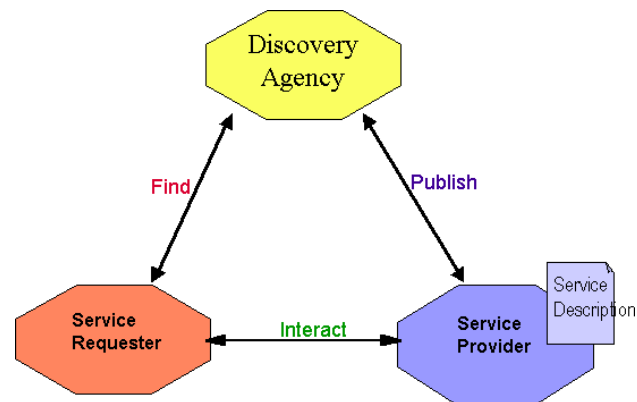
Web Services と Voice XML

▶ Web Services

- ▶ XMLによりネットワークプロトコルを記述する
- ▶ XML Protocol
 - ▶ SOAP (Simple Object Access Protocol)
- ▶ Web Service Description Language

▶ Voice XML

- ▶ 音声応答をXMLで記述
- ▶ Multimodalへ進化



```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns=http://www.w3.org/2001/vxml
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd" version="2.0">
  <form>
    <field name="drink">
      <prompt>Would you like coffee, tea, milk, or nothing?</prompt>
      <grammar src="drink.grxml" type="application/srgs+xml"/>
    </field>
    <block>
      <submit next="http://www.drink.example.com/drink2.asp"/>
    </block>
  </form>
</vxml>
```


特定分野のXML形式

▶ 気象庁防災情報XML

- ▶ 気象庁が発する気象および防災情報のXML形式
- ▶ <http://xml.kishou.go.jp/index.html>

▶ TravelXML

- ▶ 旅行業界がデータをやり取りするための形式
- ▶ http://xmlconsortium.org/wg/TravelXML/TravelXML_index.html

▶ ContactXML

- ▶ 名刺情報のXML
- ▶ <http://www.contactxml.org/>

▶ XBRL (Extensible Business Reporting Language)

- ▶ 財務諸表などのビジネスレポートを電子化して、ビジネスレポートの作成の効率化、比較・分析などの二次利用が行えることを目的とする。
- ▶ 金融庁、東京証券取引所、国税庁などが利用
- ▶ <https://www.xbrl.or.jp/>

まとめ

- ▶ HTML関係

- ▶ 画像形式
- ▶ SVG
- ▶ MathML

- ▶ XML関係

- ▶ XSL
- ▶ XML Schme
- ▶ その他

宿題: SVGでロゴ作成

- ▶ 作成している海外の街のWebページのオリジナルロゴをSVGで作成しなさい
 - ▶ 特別なソフトは使わずに、テキストエディタで作成しなさい。

```
<!DOCTYPE html>
<html>
  <head>
    <title>SVG Example</title>
  </head>
  <body>
    <svg width="500" height="200">
      <rect style="stroke:black;fill:none" width="500" height="200" x="0" y="0" />
      <path style="fill:red" d="M 100 60 L 400 150 L 20 50 L 300 150 L 50 30 L 150 40 Z" />
      <path style="fill:#88ff88" d="M 100 160 L 420 50 L 200 150 L 350 50 L 150 130 L 40 180 Z" />
      <circle cx="100" cy="120" r="20" style="fill:#8888ff" />
      <circle cx="400" cy="150" r="20" style="fill:#ff88ff" />
      <text style="fill:#8822ff" x="150" y="130" font-family="Times" font-size="100">SFC</text>
    </svg>
  </body>
</html>
```

