Slide URL

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

# Web情報システム構成法
# 第10回　XML

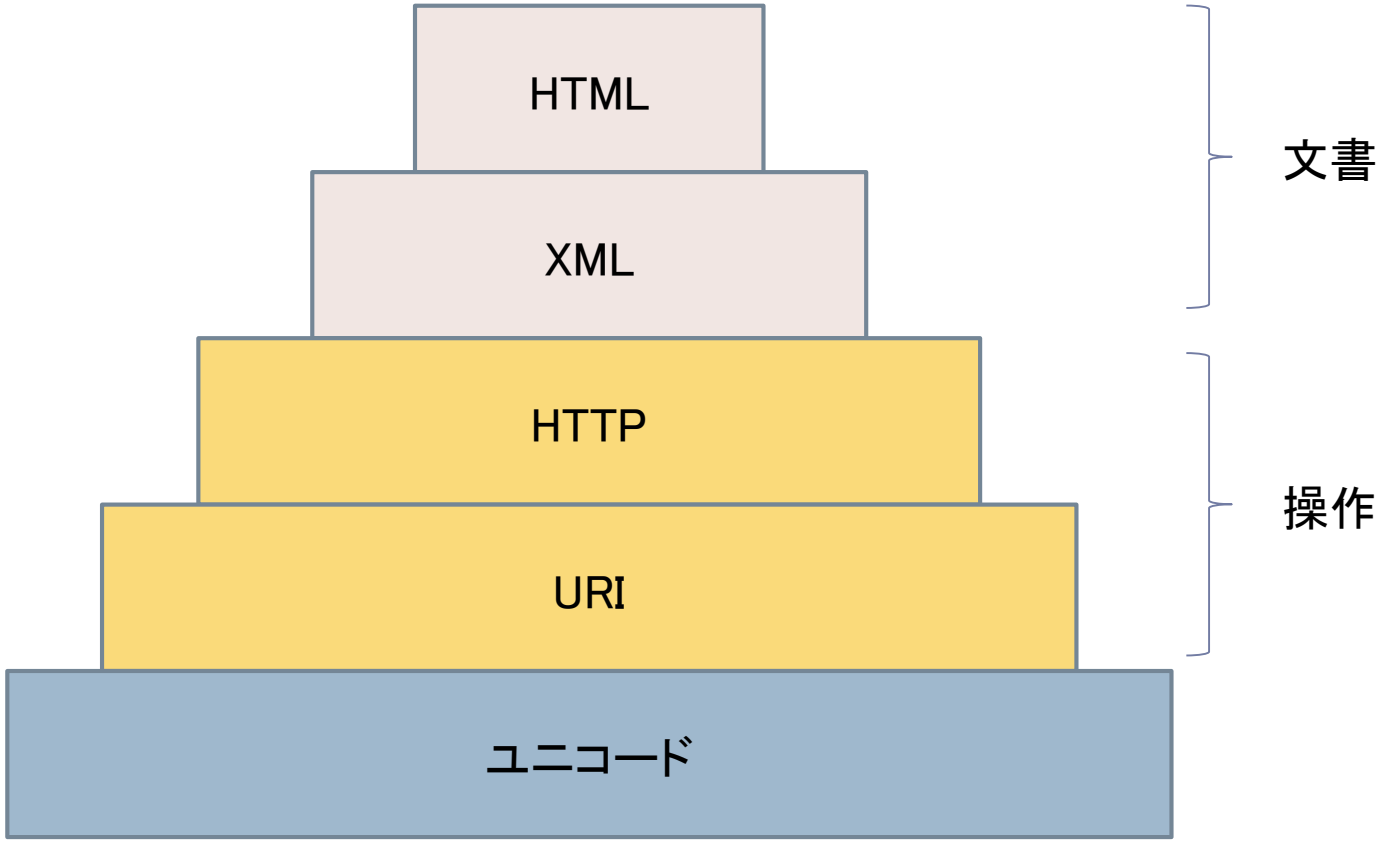
萩野 達也（hagino@sfc.keio.ac.jp）

# Webアーキテクチャ

- アーキテクチャ(Longmanによる定義)
  1. the style and design of a building or buildings
  2. the art and practice of planning and designing buildings
  3. the structure of something
  4. the structure of a computer system and the way it works

- Webアーキテクチャ
  - Webの構造
  - Webの基本原理
  - Webの互換性を保つためのもの
  - Webに関する質問に答える
  - Architecture of the World Wide Web, Volume One
    - http://www.w3.org/TR/webarch （15 December 2004）

# Web文書階層



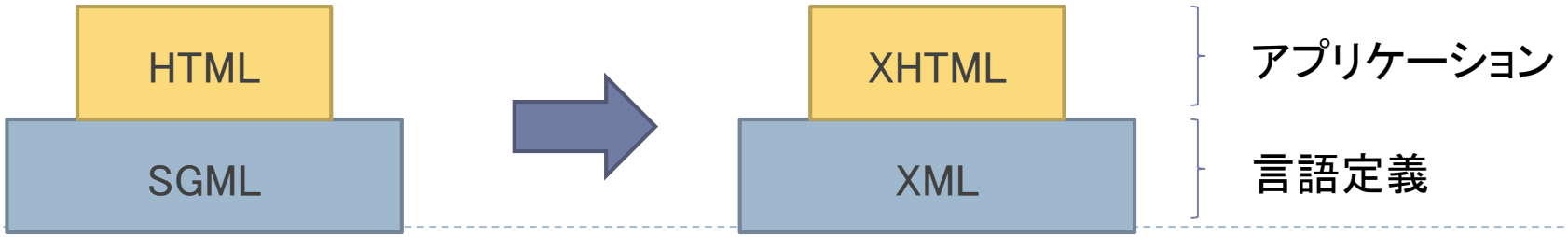
# Web文書に関する基本原理（Webアーキテクチャから）

- バージョンを付ける
  - 文書フォーマットはバージョン情報を含むべきである.
  - XML形式の使用ではXML名前空間の変更に関する情報を含むべきである.
- 拡張性
  - だれでもが拡張できる機能を提供するべきである.
  - 拡張は元の仕様の適合性に影響してはならない.
  - 不明の拡張の対するユーザエージェントの振る舞いを規定すべきである.
- 内容と表現とインタラクションの分離
  - 表現やインタラクションと分離して文章を書くことができるべきである.
- ハイパーテキスト
  - 他の資源をリンクする方法を含むべきである.
  - 内部文書だけでなく，Web全体にリンクできるべきである.
  - URIを特定のスキーマだけにするなどの利用に制限を設けず，著者が自由に利用できるべきである.
  - ハイパーテキストの場合にはハイパーリンクを利用すべきである.

# XMLとは

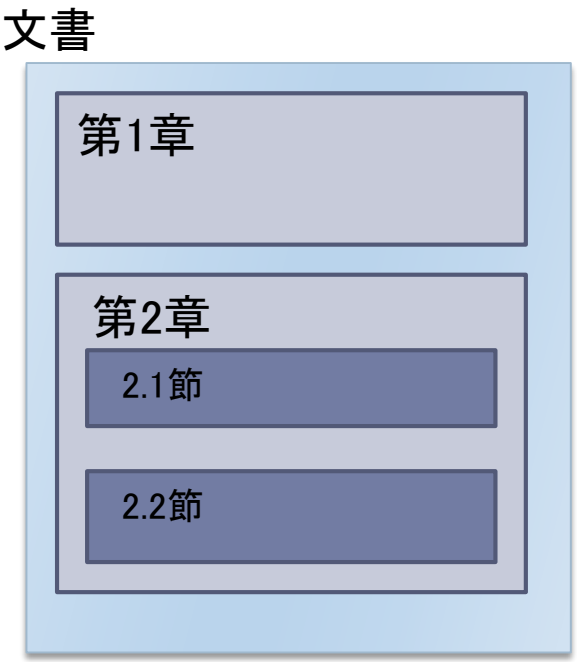
- XML (Extensible Markup Language)
  - マークアップ言語を定義する言語
  - XML 1.0はW3C によって1998年に制定
  - 現在は2つのバージョン：1.0と1.1

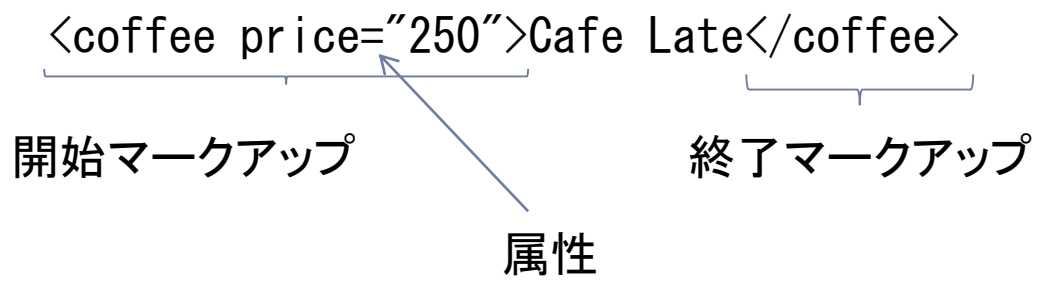
- Design goals for XML:
  1. XMLはインターネットでの利用に問題がないこと.
  2. XMLは広範囲のアプリケーションをサポートすること.
  3. XMLはSGMLと互換性を持つこと.
  4. XML文書を処理するプログラムを簡単に書くことができること.
  5. XMLのオプショナルな部分はできるだけ少ないこと. できれば0であること.
  6. XML文書は人間にも読みやすく理解しやすいこと.
  7. XMLの設計は迅速に行うこと.
  8. XMLの設計は形式的で簡潔であること.
  9. XML文書は簡単に作成できること.
  10. XMLマークアップの簡潔さはあまり重要ではない. .



# 構造化文書

- 文書は構造を持つ
  - 段落，章，索引
  - 履歴書，願書，などなど
- マークアップ
  - 構造を示すための埋め込まれたコード
  - Marking Upが語源
  - SGMLではタグをマークアップとして利用



# XML文書

- XML宣言
  - XML文書であることを示す
  - 文字エンコーディングの指定
- 要素
  - 開始タグと終了タグでマークアップ
  - 空要素タグ
- CharData
  - 文字データ
- Reference
  - 文字参照: <, A, ...
- CDSect
  - CDATAセクション
- PI
  - 処理命令
- コメント
  - 注釈

```
<?xml version="1.0" encoding='iso-2022-jp'?>
<!-- 慶応SFCレストラン  -->
<restaurant>
  <name>慶応レストラン</name>
  <place>SFC キャンパス</place>
  <menu>
    <item  price="150">コーヒー</item>
    <item  price ="250">カフェラテ</item>
    <item  price="400">サンドイッチ</item>
    <item  price="700">スパゲッティ</item>
  </menu>
  <open from="10:00" to="17:00" />
  <networks>
    <network type="無線LAN" />
    <network type="優先LAN" />
  </networks>
  <misc>
    休憩や会合にご利用ください.
  </misc>
</restaurant>
```

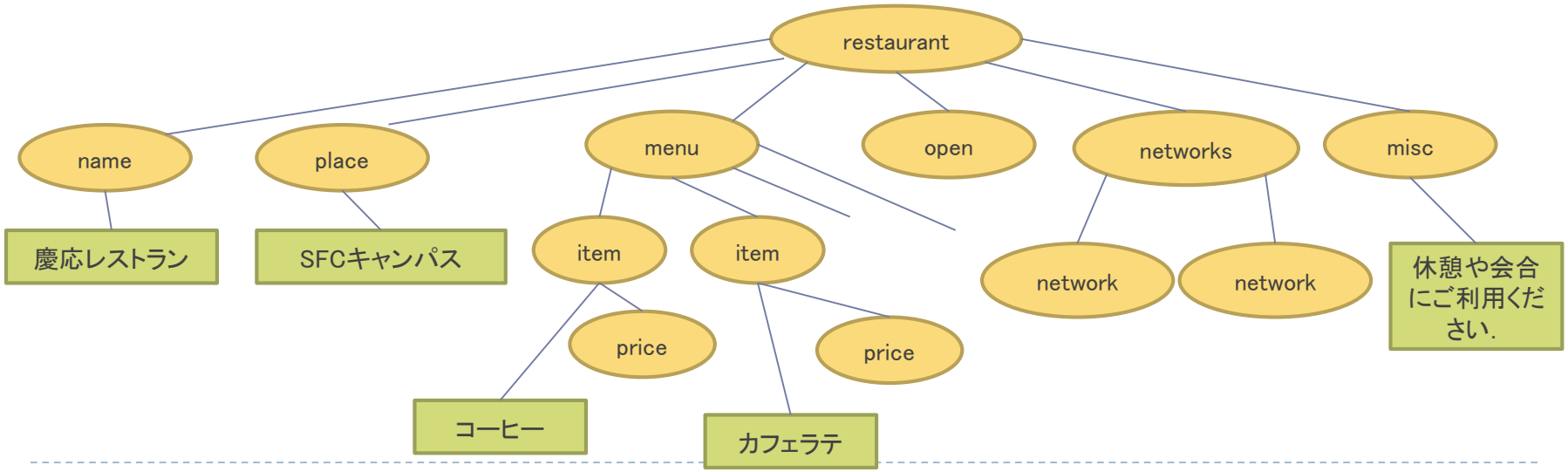
# 要素

- XML文書は要素からなる
  - 要素はタグでマークアップされる.

```
<element attribute="値">内容</element>

<element attribute="値" />
```

  - 要素は他の要素や文字データを含んでもかまわない
  - タグは正しくネストしなくてはならない
  - XML文書をパースすると一つの木構造になる

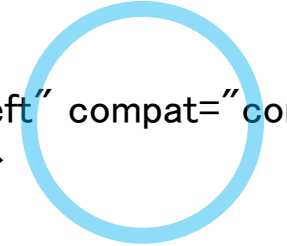


# 要素と属性

- 要素
  - 2つの要素が重なることはない　=　正しくネストすること
  - 要素名では大文字と小文字は区別される
  - 閉じタグを省略することはできない
  - 「<empty />」は「<empty></empty>」の省略形

- 属性
  - 属性名の大文字と小文字は区別される
  - 属性の値は「'」あるいは「″」でくぎられなくてはならない
  - 一つの要素の中で属性名はユニークでなくてはならない
  - 属性の順番は関係ない

<P class=error compat class=left>エラー
次の段落



<p class=″error left″ compat=″compat″>エラー
次の段落

# CDATAセクション

- XMLでは「<」を内容として書くのが大変

<h1 class='abc'>ヘッダ</h1>

文字参照で書くと

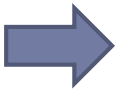
<h1 class="abc">ヘッダ</h1>

CDATAセクションを利用

<![CDATA[<h1 class='abc'>ヘッダ</h1>]]>

- Javascriptでの利用

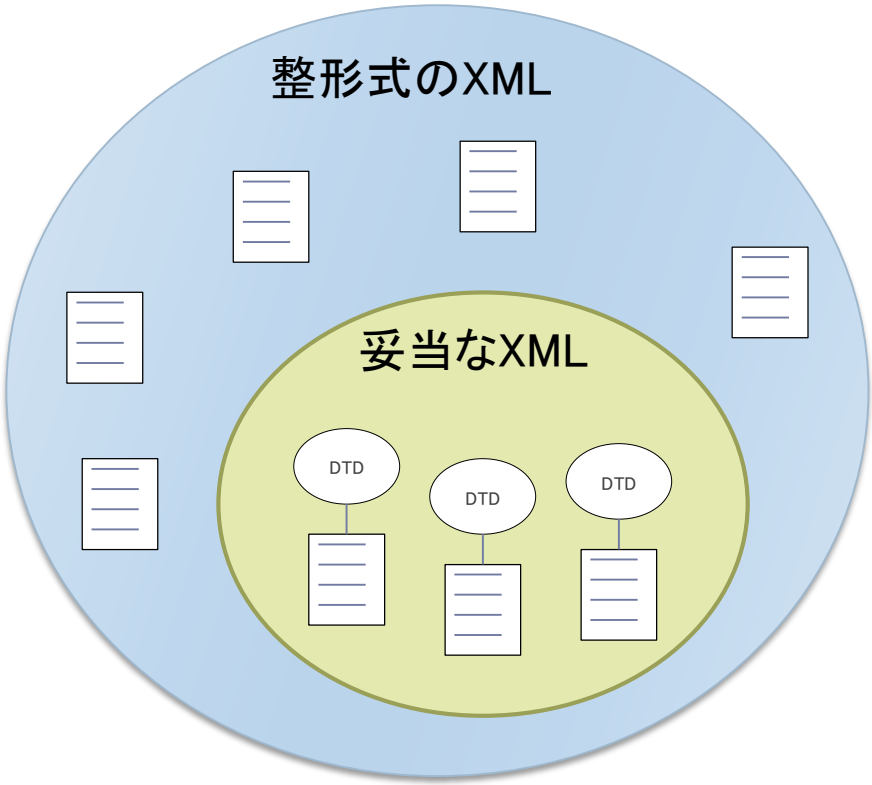
```
<script type="text/javascript">
//<!--
  for (i = 0; i < 10; i++)
    document.write(i);
//--></script>
```



```
<script type="text/javascript">
<![CDATA[
  for (i = 0; i < 10; i++)
    document.write(i);
]]></script>
```

# 2つの種類のXML文書

- 整形式のXML
  - タグのネストが正しく行われている
  - 属性が正しく指定されている
  - その他整形式の条件を満たす
- 妥当なXML
  - 整形式であり，かつ，
  - 文書型に一致する
- 整形式のXMLはどうして必要か
  - 文書型(DTD)の設計が難しい
  - DTDに従うのが難しい
  - DTDを拡張したい
  - DTDのないデータも許したい
  - 整形式であればパースは可能



# DTDとは？

- Document Type Definition
  - XMLの文書構造
  - 要素の中身を指定
  - 要素のコンテキストを指定
  - 要素の属性を指定

- DTDにより文書群が定義される
  - HTMLはHTML DTDにより定義されたXML文書
  - DTDは**XMLアプリケーション**を規定する

```
<!ELEMENT html (head,body)>
<!ELEMENT head (title|base|script|style|meta|link)*>
<!ELEMENT title (#PCDATA)>
<!ATTLIST title id   ID        #IMPLIED
                lang NMTOKEN   #IMPLIED
                dir  (ltr|rtl) #IMPLIED>
<!ELEMENT base EMPTY>
<!ATTLIST base href  CDATA     #REQUIRED
               id    ID        #IMPLIED>
```

# DTDの指定方法

文書内で指定

```
<?xml version="1.0">
<!DOCTYPE example SYSTEM [
  <!ELEMENT example (title,date)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
]>
<example>
  <title>This is an example.</title>
  <date>2014 April 28th</date>
</example>
```

外部ファイルとして指定

```
<?xml version="1.0">
<!DOCTYPE example SYSTEM "example.dtd">
<example>
  <title>This is an example.</title>
  <date>2014 April 28th</date>
</example>
```

公開識別子を利用

```
<!DOCTYPE html PUCLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

# XMLを作ってみよう

- 個人情報(名前，住所，電話番号など)のXMLを作ってみましょう.

```
<?xml version="1.0"?>
<person>
  <name>
    <familyName>萩野</familyName>
    <givenName>達也</givenName>
    <familyNameKana>はぎの</familyNameKana>
    <givenNameKana>たつや</givenNameKana>
  </name>
  <adress>住所</address>
  <email>hagino@sfc.keio.ac.jp</email>
  <tel kind="オフィス">0466-49-3446</tel>
  <tel kind="携帯">090-3060-3465</tel>
</person>
```

# 個人情報のDTDを書きなさい

```
<!ELEMENT person (name,address,email*,tel*)>

<!ELEMENT name (familyName,givenName,familyNameKana?,givenNameKana?)>

<!ELEMENT familyName (#PCDATA)>

....
```

# XML名前空間

- 背景
  - XMLが色々な分野で使われている
  - 複数のXML文書を混ぜて使いたい
  - 要素名や属性名が重ならないようにしないといけない
- 仕様
  - Namespaces in XML 1.0 (Third Edition)
    - W3C勧告：2009/12/8
  - Namespaces in XML 1.1 (Second Edition)
    - W3C勧告：2006/8/16
- 基本
  - Qualified names (QNames) を用いる
    - (namespace prefix) : (local part)
  - プレフィックスはURI (IRI)を参照
  - プレフィックスは展開されたURIとして比較される
  - 同じURIに展開されるとプレフィックスが異なっても同じとみなされる

# 名前空間の宣言

- 宣言
  - xmlns属性を利用
  - プレフィックスとURIをバインド

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <!-- prefix edi is bound to http://ecommerce.org/schema
       in element x and its content -->
</x>
```

- 要素名として利用

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <edi:price units='yen'>3218</edi:price>
</x>
```

- 属性名として利用

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <lineItem edi:taxClass="free">Baby food</lineItem>
</x>
```

# 名前空間のスコープ

- 名前空間の宣言は子要素に継承される

```
<?xml version="1.0"?>
<html:html xmlns:html='http://www.w3.org/TR/REC-html40'>
  <html:head><html:title>Frobnostication</html:title></html:head>
  <html:body>
    <html:p>Moved to <html:a href='http://frob.com'>here.</html:a>
    </html:p>
  </html:body>
</html:html>
```

- 複数の名前空間を同時に宣言することも可能

```
<?xml version="1.0"?>
<bk:book xmlns:bk='urn:loc.gov:books'
         xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <bk:title>Cheaper by the Dozen</bk:title>
  <isbn:number>1568491379</isbn:number>
</bk:book>
```

# デフォールト名前空間

- デフォールト名前空間
  - プレフィックスを何回も書くのは面倒
  - デフォールト名前空間は要素にしか影響しない
  - 属性にはデフォールト名前空間はない

```
<?xml version="1.0"?>
<html xmlns='http://www.w3.org/TR/REC-html40'>
  <head><title>SFC Home Page</title></head>
  <body>
   <p>SFC is one of the campus of
      <a href='http://www.keio.ac.jp/'>Keio university</a>.</p>
  </body>
</html>
```

デフォールトをまぜる

```
<?xml version="1.0"?>
<book xmlns='urn:loc.gov:books'
      xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <title>Encouragement of Learning</title>
  <isbn:number>9784003310236</isbn:number>
</book>
```

```
<?xml version="1.0"?>
<book xmlns='urn:loc.gov:books'
      xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <title>Encouragement of Learning</title>
  <isbn:number>9784003310236</isbn:number>
  <notes>
    <p xmlns='urn:w3-org-ns:HTML'>
      Students <em>must</em> read this book.
    </p>
  </notes>
</book>
```

# 予定に関するXML

- 予定（会議や授業）に関するXMLを作成しなさい

```
<?xml version="1.0">
<schedule>
  <date>2014/4/29</date>
  <time>19:00</time>
  <place>i506</place>
  <issue>会議</issue>




</schedule>
```

# 個人情報と予定をまぜる

- 予定のXMLの会議参加者を個人情報XMLで書きなさい
  - 予定XMLの名前空間: http://example.org/schedule
  - 個人情報XMLの名前空間: http://example.org/personalInfo

```
<?xml version="1.0">
<schedule xmlns="http://example.org/schedule"
          xmlns:pi="http://example.org/personalInfo">
  <date>2014/4/29</date>
  <time>19:00</time>
  <place>i506</place>
  <issue>meeting</issue>
  <pi:person>
    <pi:name><pi:familyName>萩野</pi:familyName></pi:name>
  </pi:person>
</schedule>
```

# 課題:授業シラバスのXMLを作ってみよう

- 授業シラバスをXMLで作ってみなさい
  - 授業のシラバスの構成要素を考え，構造化しなさい
  - DTDを考える前に，例としてタグ（要素）を考えて，シラバスを書いてみましょう.
  - そのあとで，DTDを決めて，必須な項目や，繰り返しなどを考えましょう.

- 提出
  - https://vu5.sfc.keio.ac.jp/kadai/
  - XMLを提出
  - DTDはXMLに埋め込むこと
  - 自分の履修している授業のシラバスをサンプルとしてXMLで書くこと
  - 萩野の担当している授業は対象外
  - 締め切り：6月24日正午

syllabus.xml

```
<?xml encoding="UTF-8"?>
<!DOCTYPE syllabus SYSTEM [
  <!ELEMENT syllabus (...)>
  ...
]>
<syllabus>
  ...
</syllabus>
```

# まとめ

- Webアーキテクチャ
  - Web文書の基本原理
- XML
  - 構造化文書
  - XML名前空間