

平成22年度 ITとサービスの融合による新市場創出促進事業

## 流通フォーマットエンコーディングガイドライン

平成23年 3月

財団法人 日本情報処理開発協会  
インディゴ株式会社

エンコーディングガイドライン .....	4
1. SVG【フロア情報】.....	5
① SVG ドキュメントの基本構造 .....	8
② SVG ドキュメントの標準メタデータ.....	10
ドキュメントのタイトル <b>【recommended】</b> .....	10
ドキュメントの説明 <b>【recommended】</b> .....	11
③ SVG ドキュメントの拡張メタデータ.....	12
メタデータのフレームワーク <b>【required】</b> .....	12
RDF/XML の記述制約 <b>【required】</b> .....	14
RDF/XML の具体化 .....	14
④ SVG の一般描画要素 (use から参照される描画要素) .....	15
Picket 及び区画ポリゴンについて <b>【required】</b> .....	16
⑤ SVG の一般描画要素 (背景図となる図面・追記の描画要素を収録) .....	18
背景図について .....	19
区画ポリゴン及び区画ポリラインについて <b>【required】</b> .....	20
⑥ Picket のための描画要素.....	21
Picket の ID <b>【required】</b> .....	22
Picket のリンク <b>【required】</b> .....	24
Picket の メタ情報 <b>【optional】</b> .....	25
SVG の一般描画要素 (defs と use で定義した描画要素) の関係 <b>【参考】</b> .....	28
アイコンの利用 <b>【参考】</b> .....	29
2. RDF/XML in SVG【フロアとアンカーポイントの情報】 .....	32
① RDF/XML の基本構造.....	35
基本構造 <b>【recommended】</b> .....	35
ライセンスを考慮しない基本構造 <b>【optional】</b> .....	36
② ドキュメント管理系メタデータ.....	38
ドキュメントの種別 <b>【optional】</b> .....	38
ドキュメントの作成者 <b>【optional】</b> .....	38
ドキュメントの権利者 <b>【optional】</b> .....	39
ドキュメントの作成日時 <b>【optional】</b> .....	41
ドキュメントの最終更新日時 <b>【optional】</b> .....	41
③ ライセンス管理系メタデータ .....	42
ライセンスの宣言 <b>【recommended】</b> .....	42
④ 空間情報系メタデータ .....	43
関連ドキュメントへのリンク <b>【recommended】</b> .....	43
入れ子構造を持つドキュメントの表現 <b>【optional】</b> .....	43

SVGドキュメントの代表点 <b>【recommended】</b> .....	45
地理座標との関連付け <b>【required】</b> .....	46
地理座標系との関連付け① ～ SVG Tiny 1.2 <b>【optional】</b> .....	47
地理座標系との関連付け② ～ 地理情報 SVG 仕様 <b>【optional】</b> .....	48
地理座標系との関連付け③ ～ アンカーポイントの埋め込み <b>【optional】</b> .....	49
3. HTML <b>【建物情報】</b> .....	53
① HTML の基本構造.....	55
基本構造 <b>【recommended】</b> .....	55
② HTML のメタデータ.....	56
HTML ドキュメントのタイトル <b>【required】</b> .....	56
ドキュメントの作成日時 <b>【optional】</b> .....	56
ドキュメントの最終更新日時 <b>【optional】</b> .....	57
ドキュメントの作成者情報 <b>【optional】</b> .....	58
ドキュメントの権利者情報 <b>【optional】</b> .....	59
ドキュメントの代表点メタデータ <b>【optional】</b> .....	60
③ HTML の本文 .....	61
ドキュメントの構造化 <b>【required】</b> .....	61
構造化データへの付加情報 <b>【optional】</b> .....	63
ハイパーリンクへの付加情報 <b>【optional】</b> .....	64
4. CSV <b>【ネットワーク情報】</b> .....	66
①CSV の基本構造 .....	66
基本構造 <b>【recommended】</b> .....	66
②CSV のメタデータ.....	67
建物情報へのリンク <b>【required】</b> .....	67
起点ノード ID 及び終点ノード ID <b>【required】</b> .....	67
Link の メタ情報 <b>【required】</b> .....	68

## エンコーディングガイドライン

---

流通フォーマットは、既存の言語仕様やメタデータ記述仕様に準拠した形で記述することを前提としており、言語仕様に対する独自拡張や仕様策定は特に発生しない。これは、コンテンツの後方互換性を維持するために重要な指針である。ただし、単に仕様の範囲内で任意のコンテンツやメタデータを作成するだけでは、空間参照系データベースに要求される抽象モデルの実現には十分ではない。

言語仕様あるいはメタデータ記述仕様では、コンテンツの記述者に対して裁量が委ねられている部分が多く存在する。このような部分について、盛り込むべき情報や推奨される記述方法、遵守すべきガイドラインを定義することで、抽象モデルの表現を行うこととする。以下では、「フロア情報」「建物情報」「ネットワーク情報」の各コンテンツについて、記述のためのガイドラインを提示する。

SVG 上に配置された、URI で一意に参照できるポイントのことを、以下 **picket** (和訳:杭) と呼称することとします。

緯度経度が既知の実空間の地点のことを「アンカーポイント」あるいは「基準点」と呼称することとします。

## 1. SVG【フロア情報】

フロア情報は SVG Tiny 1.2 に準拠した形式で記述される。これにガイドラインを適用した記述例は以下のような形式となる。この実例に沿ってガイドラインの定義を行う。

```
<?xml version="1.0" encoding="utf-8"?>
<svg
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:cc="http://creativecommons.org/ns#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:svg="http://www.w3.org/2000/svg"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:crs="http://www.ogc.org/crs"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
version="1.2"
width="2000px"
height="1600px"
viewBox="2000 100000 300000 50000">
  <title>丸の内ビルディング 地下 1 階フロア図</title>
  <desc>このデータは東京都千代田区にある丸の内ビルディング【通称:丸ビル】の地下 1 階フロア図を SVG 化したものです。丸の内ビルディングの詳細な情報は関連ドキュメントをご覧ください</desc>
<!--
この文書に関する補足／改定履歴等の情報
2009-09-18 初版作成
2009-09-27 改訂。文書に関するメタデータ記述例を追加
-->
  <metadata>
    <rdf:RDF>
      <cc:Work rdf:about="">
        <dc:format>image/svg+xml</dc:format>
        <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
        <dc:creator>
          <foaf:Agent>
            <foaf:name>作成者</foaf:name>
            <foaf:mbox rdf:resource="mailto:somebody@somewhere.org"/>
            <foaf:phone rdf:resource="tel:+81-03-1234-5678"/>
            <foaf:homepage rdf:resource="http://somewhere.org/~somebody"/>
          </foaf:Agent>
        </dc:creator>
        <dc:rights>
          <foaf:Agent>
            <foaf:name>権利者</foaf:name>
            <foaf:mbox rdf:resource="mailto:somebody@somewhere.org"/>
            <foaf:phone rdf:resource="tel:+81-03-1234-5678"/>
            <foaf:homepage rdf:resource="http://somewhere.org/~somebody"/>
          </foaf:Agent>
        </dc:rights>
        <dcterms:created>2009-09-18</dcterms:created>
        <dcterms:modified>2009-09-29</dcterms:modified>
        <dcterms:isReferencedBy rdf:resource="index.html"/>
        <dcterms:conformsTo rdf:resource="{uri-to-specify-standard}"/>
        <cc:license rdf:resource="http://creativecommons.org/licenses/by-nc/3.0"/>
        <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
svg:transform="matrix(1,0,0,-1,0,0)"/>
        <foaf:topic>
          <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
        </foaf:topic>
      </cc:Work>
    </rdf:RDF>
  </metadata>
<!-- アンカーポイントと Picket の対応関係のエンコード -->
<!-- 基本的な対応関係の構造: スタンドアロン型 -->
<geo:Point geo:lat="35.00" geo:long="139">
```

```

        <foaf:depiction rdf:resource="#anchor1"/>
    </geo:Point>
    <!-- (参考) スタンドアロン形式に対して、さらに詳細な情報を付随させる応用例 -->
    <geo:Point geo:lat="35.00" geo:long="139" foaf:name="東側出入り口前基準点">
        <foaf:depiction rdf:resource="#anchor2"/>
        <dcterms:conformsTo rdf:resource="{uri-to-specify-standard}"/>
    </geo:Point>
    <!--(参考) 外部のDB・リポジトリ・文書などで定義され、URIによって参照できる基準点を対応付ける例 -->
    <geo:Point rdf:resource="{uri-to-specify-anchorpoint}">
        <foaf:depiction rdf:resource="#anchor3"/>
    </geo:Point>
</rdf:RDF>
</metadata>

<defs>
    <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
    <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
    <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
    <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

    <!-- エリアポリゴン形状 -->
    <g xml:id="AP_0000000001" class="area_polygon">
        <path
            fill="none" stroke="#00ff00" stroke-width="200" stroke-linecap="butt"
            stroke-linejoin="miter" stroke-miterlimit="4"
            d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
            transform="translate(-10,-20)" />
        <use xlink:href="#institution" x="0" y="0"/>
    </g>
</defs>

<g><!-- 図面に対応する描画要素を記述 --></g>

<!-- 輪郭 -->
<g class="frame_polygon">
    <path
        fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
        stroke-dasharray="none"
        d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
    <path
        fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
        stroke-dasharray="none"
        d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
    <path
        fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
        stroke-dasharray="none"
        d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

<!-- フロア代表点 -->
<use x="0" y="0" xlink:href="#floor" xml:id="id_0000000001" class="floor_point"/>

<!-- 出入口点 -->
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000002" class="entrance_point
id_0000000003 id_0000000004 indoor none normal normal"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000005" class="entrance_point
id_0000000006 id_0000000007 indoor none normal emergency-gate"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000009" class="entrance_point none
id_0000000010 outdoor none normal emergency-gate"/>

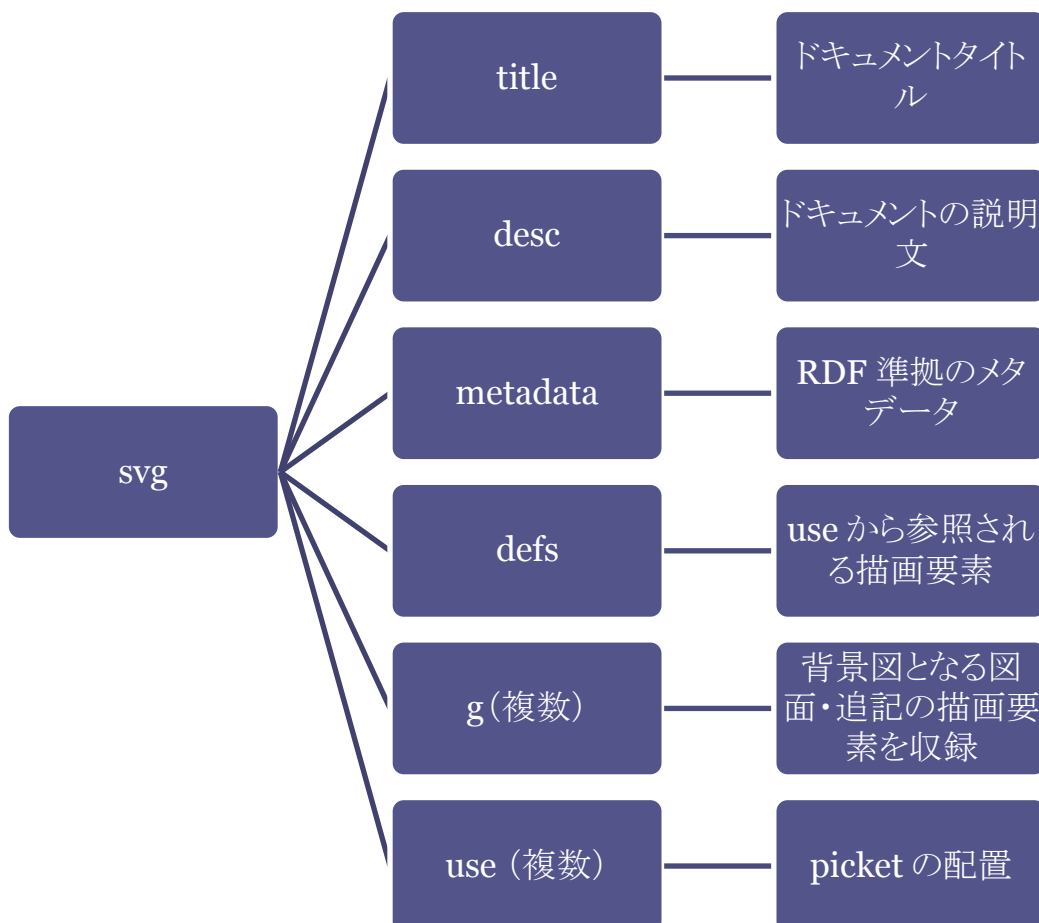
```

```
<!-- 施設代表点 -->
<use x="0" y="0" xlink:href="#institution" xml:id="id_000000003" class="institution_point A-1
tenant 飲食店"/>
<use x="0" y="0" xlink:href="#institution" xml:id="id_000000008" class="institution_point A-2
equipment 喫煙所"/>
<use x="0" y="0" xlink:href="#AP_000000001" xml:id="id_000000020" class="institution_point D-1
section 飲食店エリア"/>

<!-- ノード -->
<use x="0" y="0" xlink:href="#node" xml:id="id_000000007" class="node"/>
</svg>
```

## ① SVG ドキュメントの基本構造

SVG ドキュメントの基本的な構造についてガイドラインを設定する。



```
<svg>
  <title></title>
  <desc></desc>
  <metadata></metadata>
  <defs></defs>
  <g></g>
  <use/>
  ...
  <use/>
</svg>
```

ドキュメントの構造については、**SVG Tiny 1.2** の随所に「推奨」される形式が示唆されており、基本的にはこれらを踏襲したものとなる。

*An SVG document fragment can only contain one single 'svg' element, this means that 'svg' elements cannot appear in the middle of SVG content.*

## 5.1 Defining an SVG document fragment: the 'svg' element – SVG Tiny 1.2

スタンドアロンの SVG 文書の場合、ドキュメントのルート要素は必然的に、'svg' 要素となる。

*Authors should always provide at least a 'title', and preferably a 'desc', as an immediate child element to the 'svg' element within an SVG document, and to every significant individual graphical composition within the document.*

### 5.5 The 'title' and 'desc' elements – SVG Tiny 1.2

ドキュメントルートの 'svg' の直下に、'title' および 'desc' を配置することが推奨されている。

*Authors should provide a 'metadata' child element to the 'svg' element within a stand-alone SVG document. The 'metadata' child element to an 'svg' element serves the purposes of identifying document-level metadata.*

### 18.2 The 'metadata' element – SVG Tiny 1.2

'svg' 要素の子要素として、'metadata' 要素を配置してドキュメントレベルのメタデータを提供することが推奨されている。

*The 'defs' element is a container element for referenced elements. For understandability and accessibility reasons, it is recommended that, whenever possible, referenced elements be defined inside of a 'defs'. For performance reasons, authors should put the 'defs' element before other document content, so that all resources are available to be referenced.*

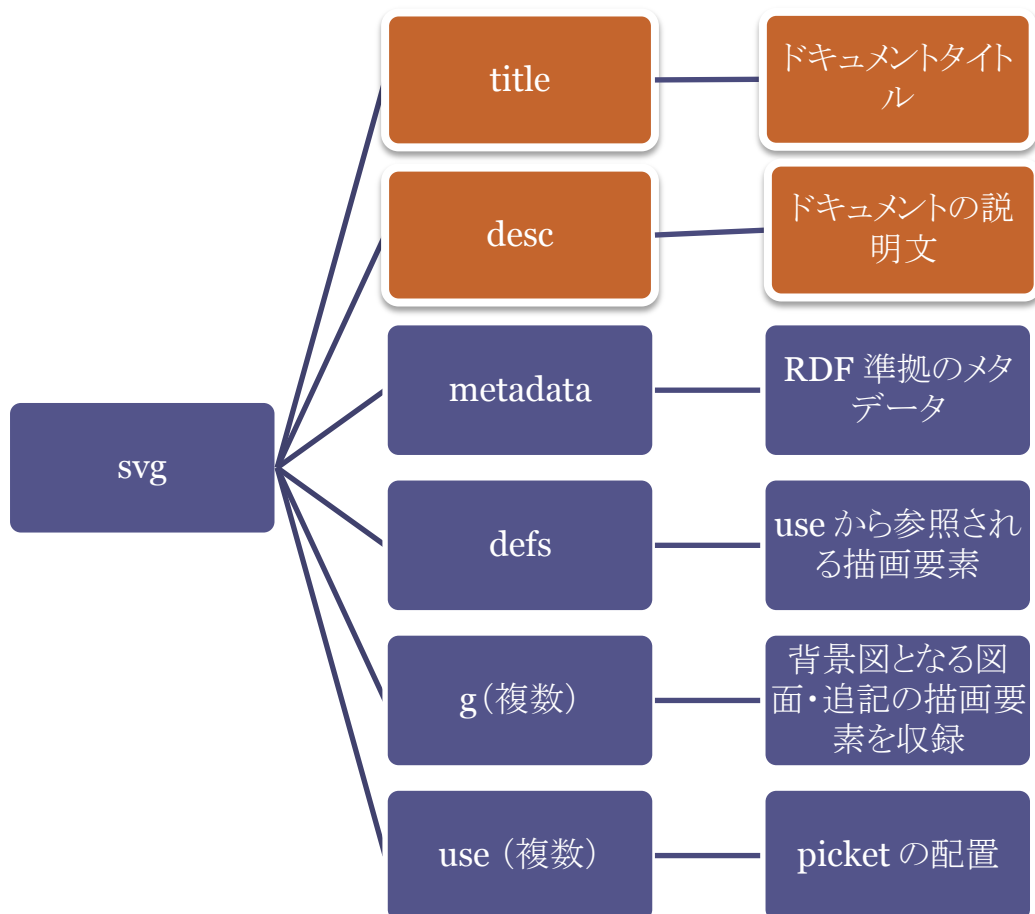
### 5.3 The 'defs' element – SVG Tiny 1.2

(主に 'use' 要素からの参照の目的で、)'defs' 要素を配置して、その配下に参照される図形要素を記述することが推奨されている。また、'defs' 要素は文書のコンテンツ本体部分の記述よりも前に配置することが推奨されている。

図形要素の記述に関しては、特に SVG Tiny 1.2 で規定されているものではないが、picket がそれ以外の描画要素の背後に隠れてしまうのは、視認性の点から問題がある。そのため、picket の描画が最前面になされるように、順序を設定している。

## ② SVG ドキュメントの標準メタデータ

SVG ドキュメントのルート直下に配置された `desc` および `title` 要素に関してガイドラインを設定する。



### ドキュメントのタイトル 【recommended】

要素・属性

記述内容

`/svg/title/text()`

建物および階を認識するための文字列。

```
<svg>
  <title>丸の内ビルディング 地下 1 階フロア図</title>
  <desc>このデータは東京都千代田区にある丸の内ビルディング【通称:丸ビル】の地下 1 階フロア図を SVG 化したものです。丸の内ビルディングの詳細な情報は関連ドキュメント http://building.org/maru/index.html をご覧ください</desc>
  ...
</svg>
```

この情報は、ドキュメントをブラウザで表示したり、ブックマークした際にタイトルとして使用されたりすることを想定したものである。機械処理を目的としたものではなく、利用者にとってドキュメントの概要を容易に把握できることが重要である。

## ドキュメントの説明【recommended】

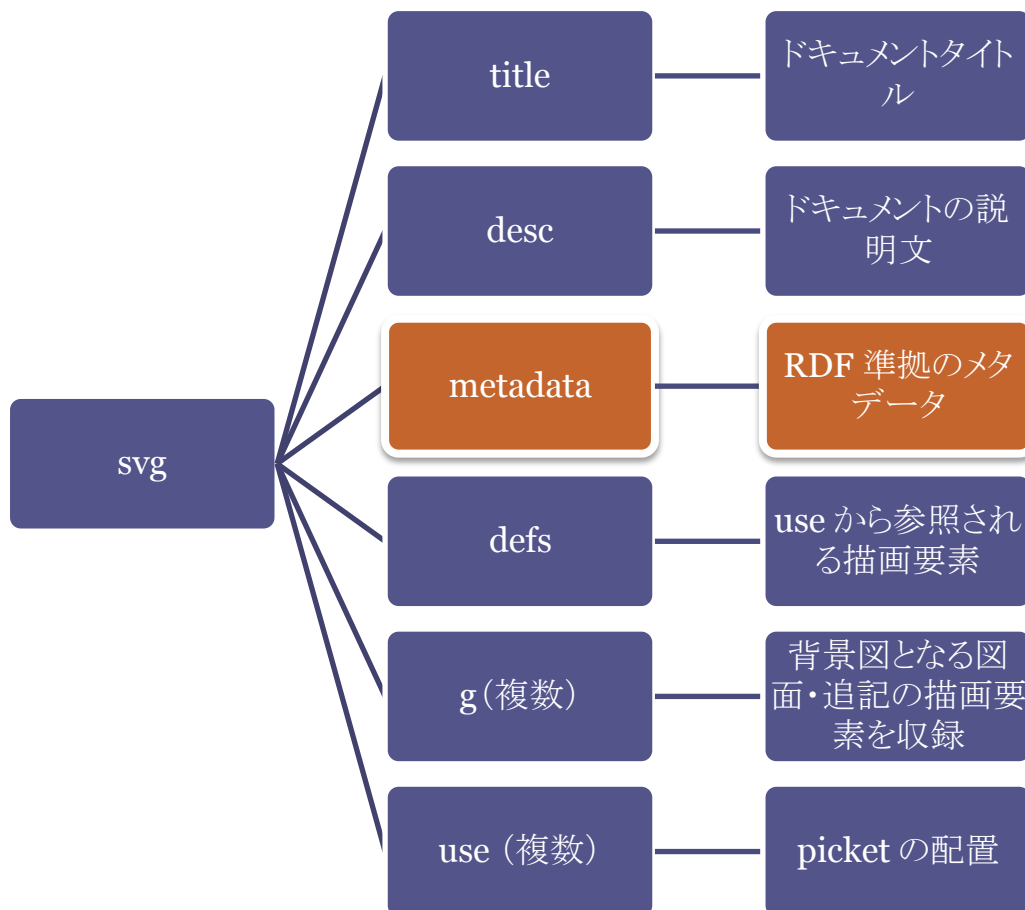
要素・属性	記述内容
<code>/svg/desc/text()</code>	このドキュメントに関する説明文。このフロアおよびフロアに関してさらに詳しい情報が外部で提供されている場合には、URL を平文で記述する。

```
<svg>
  <title>丸の内ビルディング 地下 1 階フロア図</title>
  <desc>このデータは東京都千代田区にある丸の内ビルディング【通称:丸ビル】の地下 1 階フロア図を SVG 化したものです。丸の内ビルディングの詳細な情報は関連ドキュメント http://building.org/maru/index.html をご覧ください</desc>
  ...
</svg>
```

この情報は、ファイル入手した利用者がドキュメントの概要を把握することを目的としたもので、特に機械での処理を想定したものではない。

### ③ SVG ドキュメントの拡張メタデータ

SVG ドキュメントの直下に配置された **metadata** 要素についてガイドラインを設定する。



#### メタデータのフレームワーク【required】

**Metadata** 要素の運用は、SVG ドキュメントの記述者に一任されているが、相互運用性を担保するためには、広く使用されているフレームワークを選択・採用することが必要である。

*Metadata which is included with SVG content should be specified within 'metadata' elements. The contents of the 'metadata' should be elements from other XML namespaces, with these elements from these namespaces expressed in a manner conforming with either the Namespaces in XML 1.0 or Namespaces in XML 1.1 Recommendations*

[18.2 The 'metadata' element – SVG Tiny 1.2](#)

SVG Tiny 1.2 仕様では、‘metadata’ 要素の中には、他の XML 名前空間に属する要素を記述すること、と定義されている。どのような要素を導入するかは記述者の判断に任されているが、SVG Tiny 1.2 仕様全体を通じて、RDF の使用が示唆されている。

*A popular usage of RDF metadata in the SVG community is the inclusion of license terms using the Creative Commons [CC] license framework. The popular open-source SVG authoring tool Inkscape can automatically include this license information, and inclusion of such a license is required by many collaborative Web sites. This indicates the potential and impact of metadata.*

#### 18.1 Metadata - Introduction – SVG Tiny 1.2

このような仕様上の背景、また、Inkscape などのメジャーな SVG オーサリングツールが RDF/XML を採用しているといった「デファクト」を尊重し、‘metadata’ 要素の中では RDF/XML 形式でのメタデータを記述することを前提とする。

## RDF/XML の記述制約【required】

一般的に RDF/XML 記法では同じ内容の情報を記述するのにさまざまなバリエーションがあり得る。

```
<rdf:RDF>
  <rdf:Description rdf:about="" dc:title="丸ビル地下一階フロア図"/>
</rdf:RDF>
```

```
<rdf:RDF>
  <rdf:Description rdf:about="">
    <dc:title>丸ビル地下一階フロア図</dc:title>
  </rdf:Description>
</rdf:RDF>
```

上記の二例は、あるドキュメントに関するタイトル (**dc:title**) を記述したもののだが、同じ情報を属性として記述したり、要素として記述したり、といったように柔軟性がある。さらに複雑なメタデータ構造になると、要素の省略やオントロジーによる主語・目的語の逆転などによってバリエーションはさらに多様になる。

このような柔軟性は、メタデータ記述の柔軟性を高めるものの、一般的な XML 処理系では扱いづらいというデメリットがある。特に、XPath や XSLT のような、XML の構造を前提とした方法でメタデータを利用したい場合には問題となる。

このような事情を配慮して、本ガイドラインでは RDF/XML による任意の記述ではなく、RDF/XML による記述方法がある程度固定化・定型化することを推奨し、メタデータに対するアクセス性を担保する。

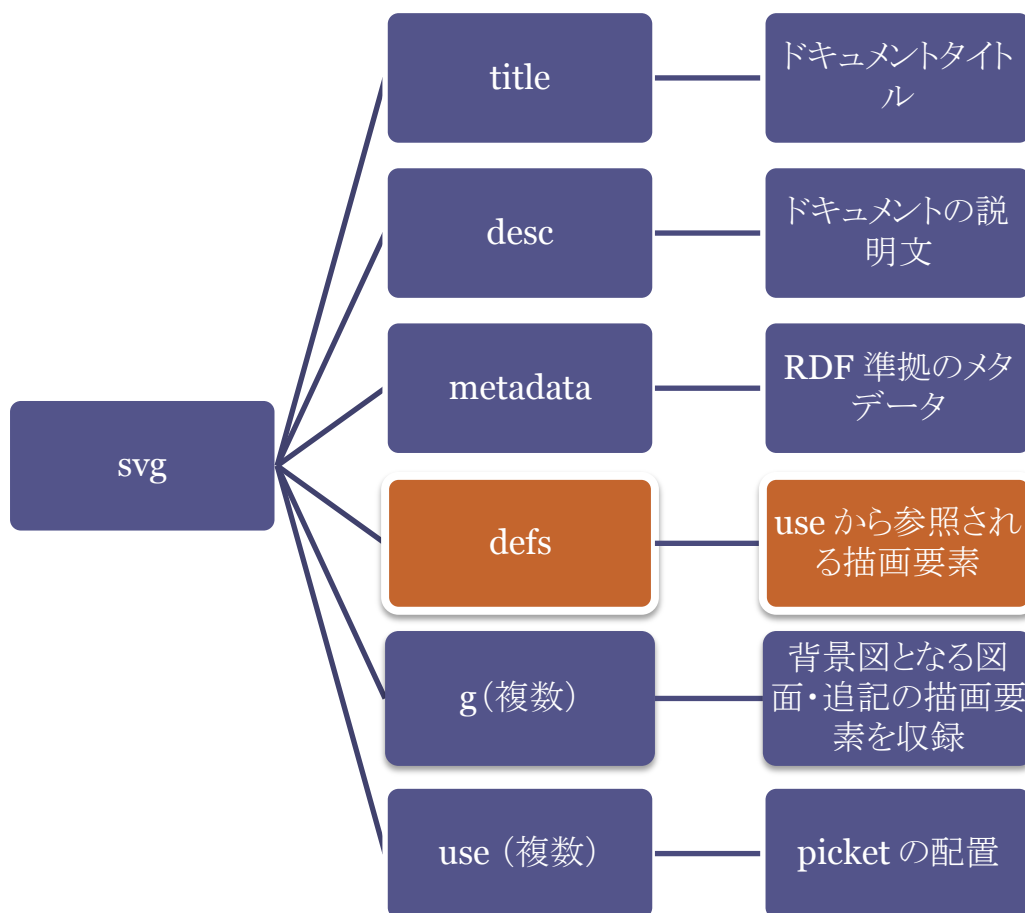
本項以降でのメタデータの記法はこの「固定化・定型化」の方針に沿って記述方法を定義したものである。なお、本ガイドラインに含まれない RDF メタデータについては、作成者が任意に記述してかまわない。

## RDF/XML の具体化

次章にて定義する。

#### ④ SVG の一般描画要素 (use から参照される描画要素)

SVG ドキュメントの直下に配置された 'defs' 要素に関しては、'use' から参照される描画要素を格納するものとし、描画要素として、Picket 及び区画ポリゴンについてのガイドラインを設ける。



## Picket 及び区画ポリゴンについて 【required】

Picketとして定義されるg要素に対しては、必ずIDを付与するものとする。後述するPicketとして定義される、use要素のclass属性でこのIDを指定することで、Picketの描画要素と配置の対応が可能となる。

区画ポリゴンとして定義されるg要素に対しては、必ずIDを付与し、class属性に区画ポリゴンの識別子を付与するものとする。また、対応するPicketをuse要素として定義し、後述するPicketとして定義されるuse要素のxlinkでこのIDを指定することで、Picketと区画ポリゴンの対応が可能となる。

区画ポリゴンは必ずPicketと対応付ける必要があり、対応関係は1:1になるようにしなければならない。

要素・属性	記述内容
<code>/svg/defs/g/@xml:id</code>	Picket 及び区画ポリゴンの ID を指定する。 (SVG Tiny 1.2) この ID は文書内でユニークでなければならず、また、xml:id 仕様に準拠した文字構成でなければならない。
<code>/svg/defs/g/@id</code>	Picket 及び区画ポリゴンの ID を指定する。 (SVG 1.1 / SVG Tiny 1.2) 過去の SVG 仕様では、xml:id 属性の代わりに id 属性が使用されている。いずれかを指定する。
<code>/svg/defs/g/use/@xlink</code>	'/svg/defs/g/'で定義した、Picket の id を指定する。
<code>/svg/defs/g/use/@x</code> <code>/svg/defs/g/use/@y</code>	区画ポリゴンの x,y 座標は“o”とする。

PicketのIDに関しては、Null及び空文字列の使用は不可とする。SVG定義は以下の通りとする。

g要素の種類	
名称	SVG定義
フロア代表点	floor

出入口点	entrance
施設代表点	institution
ノード	node

区画ポリゴンの ID に関しては規定しないが、class 属性の SVG 定義は以下の通りとする。

g 要素の種類	
名称	SVG 定義
区画ポリゴン	area_polygon

```

<defs>
  <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
  <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_0000000001" class="area_polygon">
    <path
      fill="none" stroke="#00ff00" stroke-width="200" stroke-linecap="butt"
      stroke-linejoin="miter" stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>
<!-- 輪郭 -->
<g class="frame_polygon">
  <path
    fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
  <path
    fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
  <path
    fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

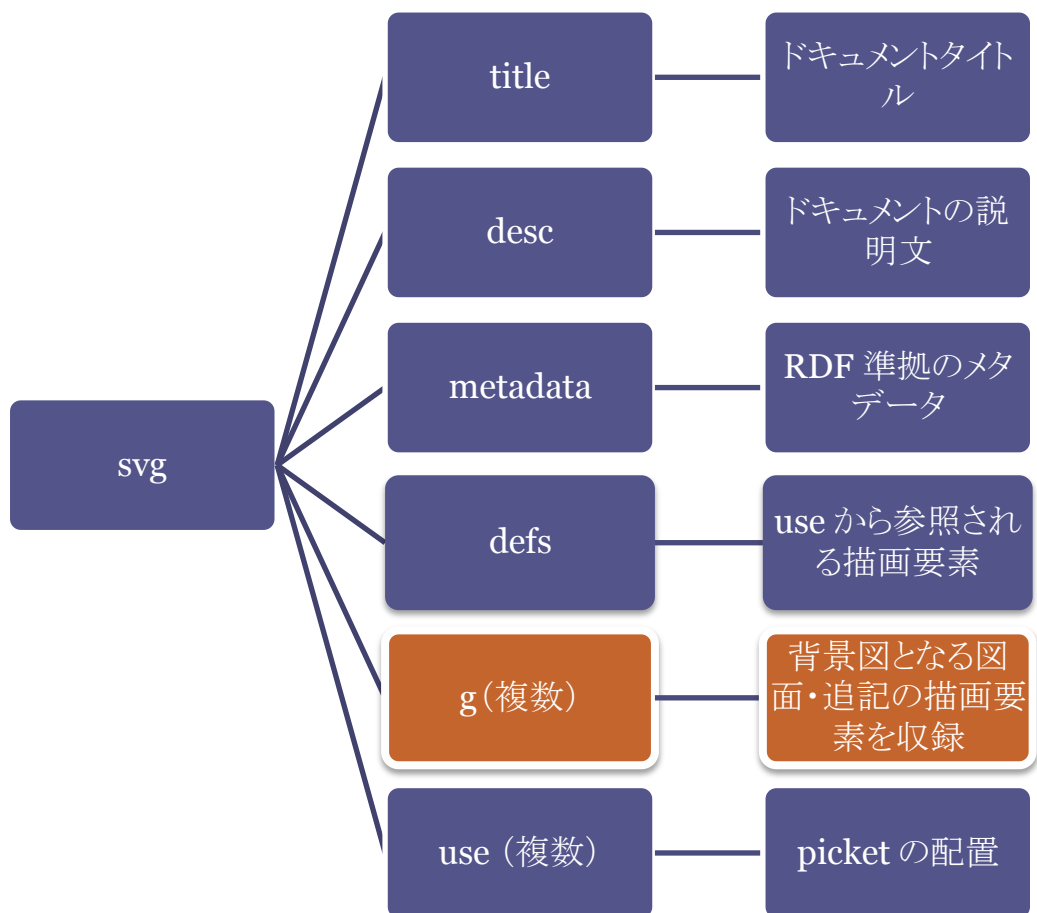
<!-- 施設代表点 -->

```

```
<use x="0" y="0" xlink:href="#AP_000000001" xml:id="id_000000020" class="institution_point D-1 section 飲食店エリア"/>
```

## ⑤ SVG の一般描画要素(背景図となる図面・追記の描画要素を収録)

SVG ドキュメントの直下に配置された `g` 要素に関しては、背景図となる図面・追記の描画要素を格納するものとし、描画要素として、区画ポリゴン及び区画ポリラインについてのガイドラインを設ける。



## 背景図について

背景図の描画要素としては、CAD 図面のようなベクトルフォーマットに由来する描画要素を挿入してもよいし、JPEG や PNG 形式のラスタ画像を挿入してもよい。ラスタ画像を使用する場合には、以下のような形式でルート直下の `g` 要素の配下に `image` 要素を配置することに対応する。なお、ラスタ画像の種類、大きさ、個数については特に規定しない。

要素・属性	記述内容
<code>/svg/g/image/@xlink:href</code>	ラスタ画像への URI を指定する。

```
<?xml version="1.0" encoding="utf-8"?>
<svg>
  <title>...</title>
  <desc>...</desc>
  <metadata>...</metadata>
  <defs>...</defs>
  <g>
    <image xlink:href="{uri-to-raster-image}" x="0" y="0" width="640" height="480"/>
  </g>
  <g class="..."></g>
  <use .../>
</svg>
```

## 区画ポリゴン及び区画ポリラインについて【required】

区画ポリゴン及び区画ポリラインには、タイトル・説明・属性といった付加情報をオプションとして設定することもできる。このような情報は SVG Tiny 1.2 の仕様の中でもさまざまな方法で記述することが可能である。

要素に対しては、'class' 属性を用いることで、ひとつ、あるいは複数の属性情報を与えることができる。

要素・属性	記述内容
<code>/svg/defs/g/@class</code>	区画ポリゴン及び区画ポリラインの属性情報を指定する。 半角スペース区切りで、複数の属性情報を指定することができる。

class 属性情報については、あらかじめリストを作成してそれらを使用するような取り決めが必要である。本流通フォーマットでは、class 属性の基本ルールとして、Null 及び空文字列の使用は不可とし、SVG 定義は以下の通りとする。

g 要素の種類	
名称	SVG 定義
フロア輪郭	frame_polygon
柱	pillar_polygon
間仕切り	partition_line
背景	back_line

```
<defs>
  <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
  <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_000000001" class="area_polygon">
    <path
      fill="none" stroke="#0ff00" stroke-width="200" stroke-linecap="butt"
      stroke-linejoin="miter" stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>
```

```

</defs>
<!-- 輪郭 -->
<g class="frame_polygon">
  <path
    fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
  <path
    fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
  <path
    fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

```

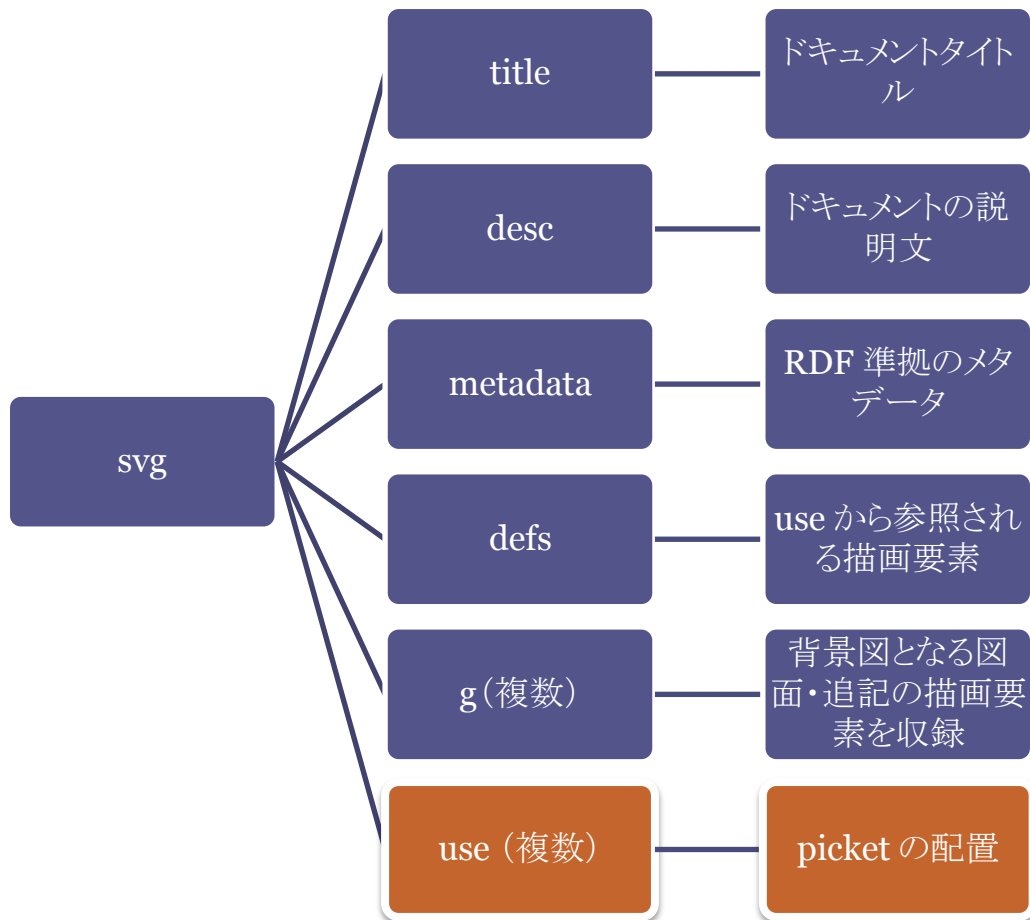
なお、吹き抜けのあるフロア形状は、フロア輪郭の描画要素に SVG の `fill-rule` プロパティを適切に設定したくり抜きのある図形を用いることで表現する。または、複数の図形を組み合わせることで表現する。

背景図としてラスタ画像を利用する場合には、たとえば吹き抜けに対応する部分を透過に設定しておくなどの処理を併用してもよいが、特に規定は設けない。透過をサポートしない画像フォーマットでの運用を考慮し、フロア輪郭として整備された図形を正式な情報とする。

## ⑥ Picket のための描画要素

---

SVG ドキュメントの直下に配置された `use` 要素は、「`picket`」に相当するものとし、以下のようなガイドラインを設ける。



### Picket の ID 【required】

Picket として定義される use 要素に対しては、必ず ID を付与するものとする。この ID を付与することで、picket に対して URI による一意の識別が可能となる。

要素・属性	記述内容
<code>/svg/use/@xml:id</code>	picket の ID を指定する。(SVG Tiny 1.2) この ID は文書内でユニークでなければならず、また、xml:id 仕様に準拠した文字構成でなければならない。
<code>/svg/use/@id</code>	picket の ID を指定する。(SVG 1.1/SVG Tiny 1.2) 過去の SVG 仕様では、xml:id 属性の代わりに id 属性が使用されている。いずれかを指定する。

```

<defs>
  <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
  <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_0000000001" class="area_polygon">
    <path
      fill="none" stroke="#00ff00" stroke-width="200" stroke-linecap="butt"
      stroke-linejoin="miter" stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>

<!-- 輪郭 -->
<g class="frame_polygon">
  <path
    fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
  <path
    fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
  <path
    fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

<!-- フロア代表点 -->
<use x="0" y="0" xlink:href="#floor" xml:id="id_0000000001" class="floor_point"/>

<!-- 出入口点 -->
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000002" class="entrance_point
id_0000000003 id_0000000004 indoor none normal normal"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000005" class="entrance_point
id_0000000006 id_0000000007 indoor none normal emergency-gate"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000009" class="entrance_point none
id_0000000010 outdoor none normal emergency-gate"/>

<!-- 施設代表点 -->
<use x="0" y="0" xlink:href="#institution" xml:id="id_0000000003" class="institution_point A-1
tenant 飲食店"/>
<use x="0" y="0" xlink:href="#institution" xml:id="id_0000000008" class="institution_point A-2
equipment 喫煙所"/>
<use x="0" y="0" xlink:href="#AP_0000000001" xml:id="id_0000000020" class="institution_point D-1
section 飲食店エリア"/>

<!-- ノード -->
<use x="0" y="0" xlink:href="#node" xml:id="id_0000000007" class="node"/>

```

## Picket のリンク【required】

Picket として定義される use 要素に対しては、必ずリンク(xlink)を定義するものとする。このリンクを定義することで、picket 及び区画ポリゴンに対しての対応が可能となる。

picket のリンク先に関しては、‘defs’で定義した ID を指定する。

```
<defs>
  <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
  <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_0000000001" class="area_polygon">
    <path
      fill="none" stroke="#00ff00" stroke-width="200" stroke-linecap="butt"
      stroke-linejoin="miter" stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>

<!-- 輪郭 -->
<g class="frame_polygon">
  <path
    fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
  <path
    fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
  <path
    fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

<!-- フロア代表点 -->
<use x="0" y="0" xlink:href="#floor" xml:id="id_0000000001" class="floor_point"/>

<!-- 出入口点 -->
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000002" class="entrance_point
id_0000000003 id_0000000004 indoor none normal normal"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000005" class="entrance_point
id_0000000006 id_0000000007 indoor none normal emergency-gate"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000009" class="entrance_point none
id_0000000010 outdoor none normal emergency-gate"/>

<!-- 施設代表点 -->
<use x="0" y="0" xlink:href="#institution" xml:id="id_0000000003" class="institution_point A-1
tenant 飲食店"/>
<use x="0" y="0" xlink:href="#institution" xml:id="id_0000000008" class="institution_point A-2
equipment 喫煙所"/>
```

```

<use x="0" y="0" xlink:href="#AP_000000001" xml:id="id_000000020" class="institution_point D-1
section 飲食店エリア"/>

<!-- ノード -->
<use x="0" y="0" xlink:href="#node" xml:id="id_000000007" class="node"/>

```

## Picket の メタ情報 【optional】

**picket** には、タイトル・説明・属性といった付加情報をオプションとして設定することもできる。このような情報は **SVG Tiny 1.2** の仕様の中でもさまざまな方法で記述することが可能だが、処理の簡便性のために、一応のガイドラインを導入する。

*The 'class' attribute assigns one or more class names to an element. The element may be said to belong to these classes. A class name may be shared by several element instances. The class attribute has several roles:*

*\* As a style sheet selector (when an author wishes to assign style information to a set of elements). Note: SVG Tiny 1.2 does not mandate the support of style sheets.*

*\* For general purpose processing by user agents.*

*The attribute value indicates membership in one or more sets. Any number of elements may be assigned to the same set. Multiple set names must be separated by white space characters.*

### 5.10.1 Attributes common to all elements - SVG Tiny 1.2

要素に対しては、'class' 属性を用いることで、ひとつ、あるいは複数の属性情報を与えることができる。

タイトル・説明に関しては、'svg' 要素の場合と同様に'title' および 'desc' 要素を利用することができる。

要素・属性	記述内容
<b>/svg/use/@class</b>	<b>picket</b> の属性情報を指定する。 半角スペース区切りで、複数の属性情報を指定することができる。
<b>/svg/use/title/text()</b>	<b>picket</b> のタイトルを指定する。

/svg/use/desc/text()

picket の説明を指定する。

```
<defs>
  <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
  <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_0000000001" class="area_polygon">
    <path
      fill="none" stroke="#00ff00" stroke-width="200" stroke-linecap="butt"
      stroke-linejoin="miter" stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>

<!-- 輪郭 -->
<g class="frame_polygon">
  <path
    fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
  <path
    fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
  <path
    fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

<!-- フロア代表点 -->
<use x="0" y="0" xlink:href="#floor" xml:id="id_0000000001" class="floor_point"/>

<!-- 出入口点 -->
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000002" class="entrance_point
id_0000000003 id_0000000004 indoor none normal normal"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000005" class="entrance_point
id_0000000006 id_0000000007 indoor none normal emergency-gate"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_0000000009" class="entrance_point none
id_0000000010 outdoor none normal emergency-gate"/>

<!-- 施設代表点 -->
<use x="0" y="0" xlink:href="#institution" xml:id="id_0000000003" class="institution_point A-1
tenant 飲食店"/>
<use x="0" y="0" xlink:href="#institution" xml:id="id_0000000008" class="institution_point A-2
equipment 喫煙所"/>
<use x="0" y="0" xlink:href="#AP_0000000001" xml:id="id_0000000020" class="institution_point D-1
section 飲食店エリア"/>
```

```
<!-- ノード -->
<use x="0" y="0" xlink:href="#node" xml:id="id_0000000007" class="node"/>
```

これらの情報は、「出入口点だけを参照したい」「特定の名前のついた点を参照したい」といったニーズに対して用意されるもので、**picket** を検索するような仕組みや、**picket** を強調表示するような仕組みと組み合わせることを想定したものである。

**class** 属性情報については、あらかじめリストを作成してそれらを使用するような取り決めが必要である。本流通フォーマットでは、**class** 属性の基本ルールとして、**<Picket** の種類,属性 (空間参照系モデルで定義したもの) **>**とし、以下の4パターンに分けることとする。

Picket の種類	属性
フロア代表点	-
出入口点	施設代表点 ID ノード ID 種別 設備 改札フラグ 非常口フラグ
施設代表点	種別情報 種別 用途
ノード	-

**Picket** の種類及び属性項目をスペースで区切ることで **class** 属性として定義する。なお、各項目では **Null** 及び空文字列の使用は不可とする。**Picket** 及び各属性項目の **SVG** 定義は以下の通りとする。

Picket の種類		属性		
名称	SVG 定義	名称	種類	SVG 定義
フロア代表点	floor_point	-	-	-
出入口点	entrance_point	施設代表点 ID	(文字列)	(施設代表点 ID)
			なし (※1)	none
		ノード ID	(文字列)	(ノード ID)
		種別	屋内出入口	indoor

			屋外出入口	outdoor
		設備	なし	none
			階段	stairs
			エレベータ	elevator
			エスカレータ	escalator
		改札フラグ	改札あり	ticket-gate
			改札なし	normal
		非常口フラグ	非常口あり	emergency-gate
			非常口なし	normal
施設代表点	institution_point	識別情報	(文字列)	(識別情報)
		種別	テナント	tenant
			付帯設備	equipment
			区画(※2)	section
		用途	(文字列)	(用途)
ノード	node	-	-	-

※1) 建物の屋外出入口など、施設とは独立した出入口の場合。

※2) 販売エリアや物販エリアなど、広い範囲を表す場合。

SVG の一般描画要素(defs と use で定義した描画要素)の関係【参考】

ポリゴンと代表点をセットで取り扱う方法について、**use** 要素から参照される図形をポリゴンではなく **g** 要素とし、**g** 要素の中にポリゴンと代表点を記述する。また、**defs** の外側で定義する **use** 要素の中で、**xlink** にポリゴンの ID を指定することによって、代表点とポリゴンをセットで扱う。

代表点とポリゴンを独立させる場合は、**defs** の外側で定義する **use** 要素の中で、**xlink** に代表点の ID を指定する。

```
<defs>
  <!-- 施設代表点の描画要素-->
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_000000001" class="area_polygon">
    <path
      fill="none" stroke="#00ff00"
      stroke-width="200"
      stroke-linecap="butt"
      stroke-linejoin="miter"
      stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>

<!-- 施設代表点 (ポリゴンとの関連づけなし)-->
<use x="0" y="0" xlink:href="#institution" xml:id="id_000000003" class="institution_point A-1 tenant 飲食店"/>

<!-- 施設代表点 (ポリゴンとの関連づけあり)-->
<use x="0" y="0" xlink:href="#AP_000000001" xml:id="id_000000020" class="institution_point D-1 section 飲食店エリア"/>
```

施設代表点の描画要素とポリゴンの描画要素の関連付け  
(g 要素の中にポリゴンと施設代表点を記述する)

施設代表点とポリゴンの関連付け  
(xlink に g 要素のポリゴン ID を設定する)

## アイコンの利用【参考】

PNG や JPEG 等の画像を屋内地図上のアイコンとして表現する場合は、次の例で示すとおり記述するとよい。

### ○アイコンファイルの保存場所

アイコン表示はアイコンファイルへのリンクとして指定することを前提するため、対象のアイコンファイルを流通フォーマットの一部として管理する。

下記の例では「**xlink:href="ticon.png"**」としているが、この場合はフロア SVG ファイルと同フォルダに「**ticon.png**」が必要である。但し、アイコンファイルの指定は URL 指定またはパス指定が可能であり、もし運用面でアイコンの存在する共有フォルダ等が規定できれば流通フォーマットの一部としての管理は不要である。

### ○施設用途毎のアイコン変更

アイコン表示の利用には施設代表点の「用途毎」にそれぞれ別のアイコン表示が必要なため、流通フォーマット上の定義としてはアイコン表示が必要な施設の用途種別毎に<defs>での代表点形状の定義を分ける。

```
-----
DEFS 要素
-----
<defs>
  <g xml:id="floor"><circle cx="0" cy="0" fill="#fff000"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" fill="#ffff00"/></g>
  <g id="institution_飲食店">
    <image xlink:href="icon1.png" width="64" height="64" x="-32" y="-32" />
  </g>
  <g id="institution_トイレ">
    <image xlink:href="icon2.png" width="64" height="64" x="-32" y="-32" />
  </g>
  <g xml:id="node"><circle cx="0" cy="0" fill="#00ffff"/></g>
</defs>

-----
USE 要素
-----
<use
  x="192833.2"
  y="484364.56"
  xlink:href="#institution_飲食店"
  xml:id="use41657"
  class="institution_point A-1 tenant 飲食店" />
<use
  x="193428.38"
  y="505592.09"
  xlink:href="institution_トイレ"
  xml:id="use41707"
  class="institution_point A-2 tenant トイレ" />
```

use 要素で指定した座標にアイコンの中心が配置される様、width,height,x,yの座標を調整する。

#### ○地図を拡大・縮小・回転してもアイコンのサイズ・方向を維持する場合

SVGを地図用途へ適用する場合、地図を拡大・縮小・回転してもアイコンのサイズ・方向が維持する場合がある。この場合は、SVG Tiny1.2の仕様(7.7.5 The TransformRef value)に基づき次ぎの例のように記述することができる。

```
-----
DEFS 要素
-----
<defs>
  <g xml:id="floor"><circle cx="0" cy="0" fill="#fff000"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" fill="#ffff00"/></g>
  <g id="institution_飲食店">
    <image xlink:href="icon1.png" width="64" height="64" x="-32" y="-32" />
  </g>
  <g id="institution_トイレ">
    <image xlink:href="icon2.png" width="64" height="64" x="-32" y="-32" />
  </g>
  <g xml:id="node"><circle cx="0" cy="0" fill="#00ffff"/></g>
</defs>

-----
USE 要素
```

```
-----  
<use transform="ref(svg,192833.2,484364.56)"  
  x="0"  
  y="0"  
  xlink:href="#institution_飲食店"  
  xml:id="use41657"  
  class="institution_point A-1 tenant 飲食店" />  
<use transform="ref(svg,193428.38,505592.09)"  
  x="0"  
  y="0"  
  xlink:href="#institution_トイレ"  
  xml:id="use41707"  
  class="institution_point A-2 tenant トイレ" />
```

### ○アイコンと背景図の区別

背景図としてラスタ画像を利用する場合には、文書中に背景画像を引用するための **image** タグと、アイコン画像に対応する **image** タグとが共存することとなる。これらの **image** 要素は、前者がルート直下の **g** 要素配下に、後者がルート直下の **defs** 要素の配下に挿入されるため、文書構造によって明確に区別することが可能である。本ガイドラインに準拠してエンコードされた文書を再利用する場合には、特定の属性やプロパティによって背景図・アイコンを区別するのではなく、文書構造に依拠した分類を行うことを前提とする。

## 2. RDF/XML in SVG【フロアとアンカーポイントの情報】

SVG ドキュメントの中で使用される RDF/XML メタデータについて、以下のような実例に沿ってガイドラインの定義を行う。

```
<?xml version="1.0" encoding="utf-8"?>
<svg
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:cc="http://creativecommons.org/ns#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:svg="http://www.w3.org/2000/svg"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:crs="http://www.ogc.org/crs"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
version="1.2">
  <title>丸の内ビルディング 地下 1 階フロア図</title>
  <desc>このデータは東京都千代田区にある丸の内ビルディング【通称:丸ビル】の地下 1 階フロア図を SVG 化したものです。丸の内ビルディングの詳細な情報は関連ドキュメントをご覧ください</desc>
<!--
この文書に関する補足／改定履歴等の情報
2009-09-18 初版作成
2009-09-27 改訂。文書に関するメタデータ記述例を追加
-->
  <metadata>
    <rdf:RDF>
      <cc:Work rdf:about="">
        <dc:format>image/svg+xml</dc:format>
        <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
        <dc:creator>
          <foaf:Agent>
            <foaf:name>作成者</foaf:name>
            <foaf:mbox rdf:resource="mailto:somebody@somewhere.org"/>
            <foaf:phone rdf:resource="tel:+81-03-1234-5678"/>
            <foaf:homepage rdf:resource="http://somewhere.org/~somebody"/>
          </foaf:Agent>
        </dc:creator>
        <dc:rights>
          <foaf:Agent>
            <foaf:name>権利者</foaf:name>
            <foaf:mbox rdf:resource="mailto:somebody@somewhere.org"/>
            <foaf:phone rdf:resource="tel:+81-03-1234-5678"/>
            <foaf:homepage rdf:resource="http://somewhere.org/~somebody"/>
          </foaf:Agent>
        </dc:rights>
        <dcterms:created>2009-09-18</dcterms:created>
        <dcterms:modified>2009-09-29</dcterms:modified>
        <dcterms:isReferencedBy rdf:resource="index.html"/>
        <dcterms:conformsTo rdf:resource="{uri-to-specify-standard}"/>
        <cc:license rdf:resource="http://creativecommons.org/licenses/by-nc/3.0"/>
        <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
svg:transform="matrix(1,0,0,-1,0,0)"/>
        <foaf:topic>
          <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
        </foaf:topic>
      </cc:Work>

      <!-- アンカーポイントと Picket の対応関係のエンコード -->
      <!-- 基本的な対応関係の構造:スタンドアロン型 -->
      <geo:Point geo:lat="35.00" geo:long="139">
        <foaf:depiction rdf:resource="#anchor1"/>
      </geo:Point>
      <!-- (参考) スタンドアロン形式に対して、さらに詳細な情報を付随させる応用例 -->
```

```

    <geo:Point geo:lat="35.00" geo:long="139" foaf:name="東側出入り口前基準点">
      <foaf:depiction rdf:resource="#anchor2"/>
      <dcterms:conformsTo rdf:resource="{uri-to-specify-standard}"/>
    </geo:Point>
    <!--(参考) 外部のDB・リポジトリ・文書などで定義され、URIによって参照できる基準点を対応付ける例 -->
    <geo:Point rdf:resource="{uri-to-specify-anchorpoint}">
      <foaf:depiction rdf:resource="#anchor3"/>
    </geo:Point>
  </rdf:RDF>
</metadata>

<defs>
  <g xml:id="floor"><circle cx="0" cy="0" r="300" fill="#fff000" stroke-width="100"/></g>
  <g xml:id="entrance"><circle cx="0" cy="0" r="300" fill="#ffff00" stroke-width="100"/></g>
  <g xml:id="institution"><circle cx="0" cy="0" r="300" fill="#000fff" stroke-width="100"/></g>
  <g xml:id="node"><circle cx="0" cy="0" r="300" fill="#00ffff" stroke-width="100"/></g>

  <!-- エリアポリゴン形状 -->
  <g xml:id="AP_000000001" class="area_polygon">
    <path
      fill="none" stroke="#00ff00" stroke-width="200" stroke-linecap="butt"
      stroke-linejoin="miter" stroke-miterlimit="4"
      d="m 10,92 65,33 -144,-16 -21,-33 145,-65 z"
      transform="translate(-10,-20)" />
    <use xlink:href="#institution" x="0" y="0"/>
  </g>
</defs>

<g><!-- 図面に対応する描画要素を記述 --></g>

<!-- 輪郭 -->
<g class="frame_polygon">
  <path
    fill="none" stroke="#ff0050" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m -65,26 33,10 -88,-158 -176 z" />
</g>

<!-- 柱 -->
<g class="piller_polygon">
  <path
    fill="none" stroke="#000050" stroke-width="100" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 115900.66,77392.404 1125.51,0 0,-1500 -1125.51,0 0,1500 z" />
</g>

<!-- 間仕切り -->
<g class="partition_line">
  <path
    fill="none" stroke="#000000" stroke-width="150" stroke-miterlimit="4" stroke-opacity="1"
    stroke-dasharray="none"
    d="m 106,204 0,477 -465,0 0,4450 -59,0 -334,322 -334,27" />
</g>

<!-- フロア代表点 -->
<use x="0" y="0" xlink:href="#floor" xml:id="id_000000001" class="floor_point"/>

<!-- 出入口点 -->
<use x="0" y="0" xlink:href="#entrance" xml:id="id_000000002" class="entrance_point
id_000000003 id_000000004 indoor none normal normal"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_000000005" class="entrance_point
id_000000006 id_000000007 indoor none normal emergency-gate"/>
<use x="0" y="0" xlink:href="#entrance" xml:id="id_000000009" class="entrance_point none
id_000000010 outdoor none normal emergency-gate"/>

<!-- 施設代表点 -->
<use x="0" y="0" xlink:href="#institution" xml:id="id_000000003" class="institution_point A-1
tenant 飲食店"/>

```

```
<use x="0" y="0" xlink:href="#institution" xml:id="id_000000008" class="institution_point A-2
equipment 喫煙所"/>
<use x="0" y="0" xlink:href="#AP_000000001" xml:id="id_000000020" class="institution_point D-1
section 飲食店エリア"/>

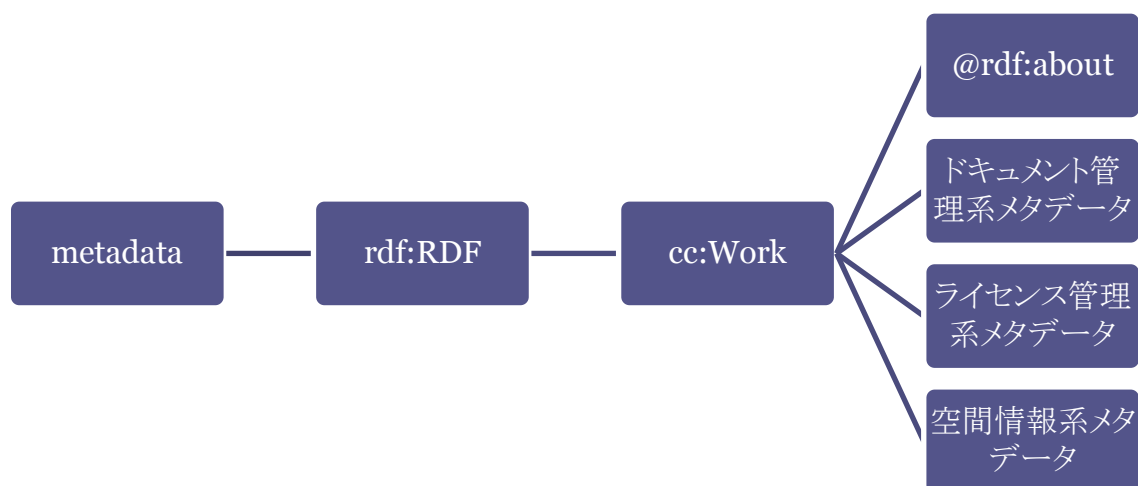
<!-- ノード -->
<use x="0" y="0" xlink:href="#node" xml:id="id_000000007" class="node"/>
</svg>
```

## ① RDF/XML の基本構造

Metadata 要素配下の基本的な構造についてガイドラインを設定する。

### 基本構造【recommended】

RDF/XML のメタデータは、SVG ドキュメントのルート直下の `metadata` 要素の配下に記述され、以下のような基本構造を持つものとする。



(以下、`rdf:RDF` 要素を起点とした表現とする)

要素・属性	記述内容
<b><code>rdf:RDF</code></b>	RDF/XML のための基本構造
<b><code>rdf:RDF/cc:Work</code></b>	RDF/XML のための基本構造。
<b><code>rdf:RDF/cc:Work/@rdf:about</code></b>	RDF/XML のための基本構造。 <code>rdf:about=""</code> という固定値を設定する。

**rdf:RDF/cc:Work/\***

RDF/XML のための基本構造。

ここに要素を挿入することで、各種のメタデータを表現する。

この定義に沿った RDF/XML は以下のような基本構造をとる。

```

<metadata>
  <rdf:RDF>
    <cc:Work rdf:about="">
      ...
    </cc:Work>
  </rdf:RDF>
</metadata>

```

**ライセンスを考慮しない基本構造【optional】**

前項の **cc:Work** は、Creative Commons の定義するライセンス管理を意図した要素で、特にライセンス管理のためのメタデータを記述する場合には **cc:Work** の使用が要求される。Inkscape のようなオーサリングツールでも、デフォルトでは **cc:Work** を使用した SVG ドキュメントを出力する。

ただ、特にライセンス管理を必要としないコンテンツについては、より広範な意味を持つ **rdf:Description** 要素を使用しても問題ない。この場合、基本構造は以下のような構成となる。

要素・属性	記述内容
<b>rdf:RDF</b>	RDF/XML のための基本構造
<b>rdf:RDF/rdf:Description</b>	RDF/XML のための基本構造。
<b>rdf:RDF/rdf:Description/@rdf:about</b>	RDF/XML のための基本構造。 rdf:about="" という固定値を設定する。
<b>rdf:RDF/rdf:Description/*</b>	RDF/XML のための基本構造。 ここに要素を挿入することで、各種のメタデータを表現する。

この定義に沿った RDF/XML は以下のような構造をとる。

```
<metadata>
  <rdf:RDF>
    <rdf:Description rdf:about="">
      ...
    </rdf:Description>
  </rdf:RDF>
</metadata>
```

なお、ライセンス宣言を必要としない場合であっても、**cc:Work** を使用することは問題がないため、新規に作成するドキュメントについては **cc:Work** を使用することを推奨する。

## ② ドキュメント管理系メタデータ

ドキュメント管理系のメタデータについて、以下のようなガイドラインを導入する。

### ドキュメントの種別【optional】

ドキュメントがある種の標準やガイドラインに準拠しており、それを明示的に宣言したい場合には、**cc:Work** 要素の配下に **dcterms:conformsTo** 要素を配置し、その **rdf:resource** 属性に標準やガイドラインを指す **URI** を指定することで記述する。

要素・属性	記述内容
<b>dcterms:conformsTo</b>	ドキュメント種別を宣言するための基本構造
<b>dcterms:conformsTo/@rdf:resource</b>	ドキュメント種別に対応する URI

```
<dcterms:conformsTo rdf:resource="{uri-to-specify-standard}"/>
```

なお、ドキュメントが複数の標準やガイドラインに準拠する場合には、同要素を複数記述することで宣言を行う。

標準やガイドラインについては、**URI**によって参照できるものであれば特に制限を設けないが、本ガイドラインでは高精度屋内地図／略地図／案内図といった、屋内図の種別を利用者に提示する用途を想定している。

### ドキュメントの作成者【optional】

ドキュメントの作成者に関するメタデータを記述する場合には、**cc:Work** 要素の配下に **dc:creator** 要素を配置、その配下に作成者情報を記述する。

要素・属性	記述内容
<b>dc:creator</b>	作成者情報のための基本構造
<b>dc:creator/foaf:Agent</b>	作成者情報のための基本構造

<b>dc:creator/foaf:Agent/foaf:name</b>	作成者名称のための基本構造(任意)
<b>dc:creator/foaf:Agent/foaf:name/text()</b>	作成者の名称を記述する
<b>dc:creator/foaf:Agent/foaf:mbox</b>	作成者メールアドレスのための基本構造(任意)
<b>dc:creator/foaf:Agent/foaf:mbox/@rdf:resource</b>	作成者のメールアドレスを URI 形式で記述する
<b>dc:creator/foaf:Agent/foaf:phone</b>	作成者電話番号のための基本構造(任意)
<b>dc:creator/foaf:Agent/foaf:phone/@rdf:resource</b>	作成者の電話番号を URI 形式で記述する
<b>dc:creator/foaf:Agent/foaf:homepage</b>	作成者ホームページのための基本構造(任意)
<b>dc:creator/foaf:Agent/foaf:homepage/@rdf:resource</b>	作成者のホームページの URI を記述する

この定義に沿った RDF/XML は以下のような構造をとる。

```

<dc:creator>
  <foaf:Agent>
    <foaf:name>作成者</foaf:name>
    <foaf:mbox rdf:resource="mailto:somebody@somewhere.org"/>
    <foaf:phone rdf:resource="tel:+81-03-1234-5678"/>
    <foaf:homepage rdf:resource="http://somewhere.org/~somebody"/>
  </foaf:Agent>
</dc:creator>

```

作成者は「人物」に限らず、グループや団体であってもかまわない。

なお、phone、homepage、mbox、name の各要素の使用は任意であり、複数使用してもかまわない。ただし、要素を使用した場合には対応するテキストあるいは rdf:resource 属性を設定する必要がある。

## ドキュメントの権利者【optional】

ドキュメントの著作権を保持する権利者に関するメタデータを記述する場合には、cc:Work 要素の配下に dc:rights 要素を配置、その配下に作成者情報を記述する。

要素・属性	記述内容
<b>dc:rights</b>	権利者情報のための基本構造
<b>dc:rights/foaf:Agent</b>	権利者情報のための基本構造
<b>dc:rights/foaf:Agent/foaf:name</b>	権利者名称のための基本構造(任意)
<b>dc:rights/foaf:Agent/foaf:name/text()</b>	権利者の名称を記述する
<b>dc:rights/foaf:Agent/foaf:mbox</b>	権利者メールアドレスのための基本構造(任意)
<b>dc:rights/foaf:Agent/foaf:mbox/@rdf:resource</b>	権利者のメールアドレスを URI 形式で記述する
<b>dc:rights/foaf:Agent/foaf:phone</b>	権利者電話番号のための基本構造(任意)
<b>dc:rights/foaf:Agent/foaf:phone/@rdf:resource</b>	権利者の電話番号を URI 形式で記述する
<b>dc:rights/foaf:Agent/foaf:homepage</b>	権利者ホームページのための基本構造(任意)
<b>dc:rights/foaf:Agent/foaf:homepage/@rdf:resource</b>	権利者のホームページの URI を記述する

この定義に沿った RDF/XML は以下のような構造をとる。

```

<dc:rights>
  <foaf:Agent>
    <foaf:name>権利者</foaf:name>
    <foaf:mbox rdf:resource="mailto:somebody@somewhere.org"/>
    <foaf:phone rdf:resource="tel:+81-03-1234-5678"/>
    <foaf:homepage rdf:resource="http://somewhere.org/~somebody"/>
  </foaf:Agent>
</dc:rights>

```

権利者は「人物」に限らず、グループや団体であってもかまわない。

なお、phone、homepage、mbox、name の各要素の使用は任意であり、複数使用してもかまわない。ただし、要素を使用した場合には対応するテキストあるいは rdf:resource 属性を設定する必要がある。

### ドキュメントの作成日時【optional】

ドキュメントの作成日時に関するメタデータを記述する場合には、**cc:Work** 要素の配下に **dcterms:created** 要素を配置、そのテキストに日付情報を記述する。

要素・属性	記述内容
<b>dcterms:created</b>	作成日時のための基本構造
<b>dcterms:created/text()</b>	作成日時を ISO8601 に準拠した文字列として記述する

```
<dcterms:created>2009-09-18</dcterms:created>
```

### ドキュメントの最終更新日時【optional】

ドキュメントの最終更新日時に関するメタデータを記述する場合には、**cc:Work** 要素の配下に **dcterms:modified** 要素を配置、そのテキストに日付情報を記述する。

要素・属性	記述内容
<b>dcterms:modified</b>	最終更新日時のための基本構造
<b>dcterms:modified/text()</b>	最終更新日時を ISO8601 に準拠した文字列として記述する

```
<dcterms:modified>2009-09-18</dcterms:modified>
```

### ③ ライセンス管理系メタデータ

---

ライセンス管理系のメタデータについて、以下のようなガイドラインを導入する。

#### ライセンスの宣言【recommended】

ドキュメントのライセンスを宣言するには、**cc:Work** 要素の配下に **cc:license** 要素を配置する。なお、ここでは **Creative Commons** ライセンスの定義を利用しているが、**Creative Commons** に該当しないライセンスについても、ライセンスを **URI** によって識別できる限りは宣言が可能である。

要素・属性	記述内容
<b>cc:license</b>	ライセンス宣言のための基本構造
<b>cc:license/@rdf:resource</b>	このドキュメントが採用するライセンスのURI。

```
<cc:license rdf:resource="http://creativecommons.org/licenses/by-nc/3.0/" />
```

#### ④ 空間情報系メタデータ

SVG ドキュメントの座標系と地理空間座標系の対応付け、SVG ドキュメントと実空間の地点情報の対応付け、SVG ドキュメントと関連するドキュメントの対応付け、といった、空間情報系に特有のメタデータの記述についてガイドラインを導入する。

#### 関連ドキュメントへのリンク【recommended】

SVG ドキュメントとは別に、SVG ドキュメントへの参照を保持する建物に関する Web ページが存在する場合には、その情報への URL を記述することでドキュメント間の関連付けを行う。このような場合には、cc:Work 要素の配下に dcterms:isReferencedBy 要素を配置する。

要素・属性	記述内容
<b>dcterms:isReferencedBy</b>	関連ドキュメントのための基本構造
<b>dcterms:isReferencedBy/@rdf:resource</b>	(この SVG ドキュメントへのハイパーリンクを設置しているであろう)建物情報への URL を指定する。

```
<rdf:RDF>
  <cc:Work rdf:about="">
    <dc:format>image/svg+xml</dc:format>
    <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
    <dcterms:isReferencedBy rdf:resource="index.html"/>
    <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
svg:transform="matrix(1,0,0,-1,0,0)"/>
    <foaf:topic>
      <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
    </foaf:topic>
  </cc:Work>
</rdf:RDF>
```

#### 入れ子構造を持つドキュメントの表現【optional】

ある SVG ドキュメントがあるフロアの全体図を表現しているものとする。同時に、フロアの部分を表現した SVG ドキュメント(部分図)が複数存在するような場合を考える。

このような場合に、個々のドキュメント間が「全体図」と「部分図」の関係にあることを以下の2つの手法のいずれかでエンコードするものとする。

(全体図を起点としたエンコーディング)

全体図と部分図(複数)が存在している場合に、全体図の **cc:Work** 要素の配下に **dcterms:hasPart** 要素を配置する。なお、参照する部分図の数に合わせて、**dcterms:hasPart** 要素は複数設置する。

要素・属性	記述内容
<b>dcterms:hasPart</b>	部分図への参照を保持するための基本構造。複数設置してもよい。
<b>dcterms:hasPart/@rdf:resource</b>	部分図となる SVG ドキュメントの URL

```
<rdf:RDF>
  <cc:Work rdf:about="">
    <dc:format>image/svg+xml</dc:format>
    <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
    <dcterms:hasPart rdf:resource="f1-part1.svg"/>
    <dcterms:hasPart rdf:resource="f1-part2.svg"/>
    <dcterms:hasPart rdf:resource="f1-part3.svg"/>
    <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
    svg:transform="matrix(1,0,0,-1,0,0)"/>
    <foaf:topic>
      <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
    </foaf:topic>
  </cc:Work>
</rdf:RDF>
```

全体図でのエンコード例 (**dcterms:hasPart** を使用)

(部分図を起点としたエンコーディング)

全体図と部分図(複数)が存在している場合に、個々の部分図の **cc:Work** 要素の配下に **dcterms:isPartOf** 要素を配置する。

要素・属性	記述内容
<b>dcterms:isPartOf</b>	全体図への参照を保持するための基本構造。
<b>dcterms:isPartOf/@rdf:resource</b>	全体図となる SVG ドキュメントの URL

```

<rdf:RDF>
  <cc:Work rdf:about="">
    <dc:format>image/svg+xml</dc:format>
    <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
    <dcterms:isPartOf rdf:resource="f1.svg"/>
    <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
svg:transform="matrix(1,0,0,-1,0,0)"/>
    <foaf:topic>
      <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
    </foaf:topic>
  </cc:Work>
</rdf:RDF>

```

部分図でのエンコード例（dcterms:isPartOf を使用）

なお、それぞれの手法は排他的なものではなく、併用も可能である。エンコード手法の選択に当たっては、フロア図の管理や流通の形態に応じて、適切な方法を選択することが望ましい。

## SVG ドキュメントの代表点【recommended】

この SVG ドキュメントのおおよその場所を特定するために、代表点の緯度・経度・地上高・階層の情報を記述する方法をガイドライン化する。

地球上のある点を表現するための RDF ボキャブラリとして、Geo Vocabulary という語彙を利用する。また、「ドキュメント」という仮想空間のオブジェクトと「地球上のポイント」という現実空間のオブジェクトの関連付けを行うために、FOAF の topic ボキャブラリを使用する。使用にあたっては、cc:Work 要素の配下に以下のような構造を持った foaf:topic 要素を挿入する。

要素・属性	記述内容
<b>foaf:topic</b>	代表点情報のための基本構造
<b>foaf:topic/geo:Point</b>	代表点情報のための基本構造
<b>foaf:topic/geo:Point/@geo:lat</b>	代表点の WGS84 緯度を度単位の数値で指定する

<b>foaf:topic/geo:Point/@geo:long</b>	代表点の WGS84 経度を度単位の数値で指定する
<b>foaf:topic/geo:Point/@geo:alt</b>	代表点の地上高をメートル単位の数値で指定する
<b>foaf:topic/geo:Point/@foaf:name</b>	フロアの名称を文字列として記述する。

```

<rdf:RDF>
  <cc:Work rdf:about="">
    <dc:format>image/svg+xml</dc:format>
    <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
    <dc:relation rdf:resource="index.html"/>
    <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
svg:transform="matrix(1,0,0,-1,0,0)"/>
    <foaf:topic>
      <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
    </foaf:topic>
  </cc:Work>
</rdf:RDF>

```

緯度・経度については、このフロアの所属する建物の代表点緯度経度と共通のものでかまわない。ここで定義する緯度経度は、地図上でのおよその位置を特定するような用途で使用することを想定している。

地上高については、厳密な定義に基づく数値を指定するのではなく、同一の建物内でのフロアの上下関係を数値化することを目的としたものである。より高い階に対しては大きな地上高、より低い階に対しては小さな地上高を指定するものとする。主に、建物内でのフロア順序を「ソート」するような目的で、機械処理用に用意したものである。そのため、外部の建物のフロア図と地上高が一致していたからといって、それぞれのフロアの高さが同一であることを意図するものではない。

フロア名称には「地下一階」「1F」「M2F」など、建物の中でフロアを識別するための任意の名称を指定する。この名称は主にデータ利用者に向けて提供されるもので、機械処理を想定したものではない。

**地理座標との関連付け【required】**

SVGドキュメントの座標系がアフィン変換によって地理座標系に(近似的にでも)変換できるような形式の場合、アフィン行列を介して SVG 座標系と地理座標系との関連付けを行うことができる。

このようなドキュメントの場合には、以下①②の方法のいずれかを使用することで、SVGドキュメントと地理座標との関連付けを行うことを推奨する。特に後方互換性を考慮して、①②双方の記述を併記することが望ましい。

他方、略地図や精度が不詳なデータのように、単純なアフィン変換による関連付けが難しい場合には、①や②の手法では地理座標系・SVG座標系変換の信頼性を担保することが難しい。このようなケースへの適用を考慮し、SVGドキュメント上の **Picket** と地理座標が既知の点(以下、アンカーポイント)を一対一で対応付けることで、部分的な地理座標・SVG座標の対応を確保する手法(③)を導入する。

なお、①②の手法と③の手法は相互補完的なものであり、ひとつの文書内で共存させても構わないものとする。特に図面の種別に依存せず、最低でもひとつの手法を導入することで、SVGドキュメントと地理座標との関係を維持することが求められる。

#### 地理座標系との関連付け① ~ SVG Tiny 1.2 【optional】

SVGドキュメントの座標系がアフィン変換によって地理座標系に(近似的にでも)変換できるような形式の場合に、RDF/XMLメタデータボキャブラリを使用して SVG座標系と地理座標系との関連付けを行う。

なお、本ガイドラインは SVG Tiny 1.2 で規定される以下の文書に準拠したものである。

*In order to allow interoperability between SVG content generators and SVG user agents dealing with maps encoded in SVG, the use of a common metadata definition for describing the coordinate system used to generate SVG documents is encouraged.*

*Such metadata must be added under the 'metadata' element of the topmost 'svg' element describing the map, consisting of an RDF description of the Coordinate Reference System definition used to generate the SVG map [RDF]. Note that the presence of this metadata does not affect the rendering of the SVG in any way; it merely provides added semantic value for applications that make use of combined maps.*

#### 7.15 Geographic coordinate systems - SVG Tiny 1.2

この定義を導入するには、cc:Work 要素の配下に crs:CoordinateReferenceSystem 要素を挿入する。

要素・属性	記述内容
<b>crs:CoordinateReferenceSystem</b>	座標系変換パラメータのための基本構造
<b>crs:CoordinateReferenceSystem/ @rdf:resource</b>	WGS84 を指す URI を指定する。通常は以下の URI を指定する <a href="http://purl.org/crs/84">http://purl.org/crs/84</a>
<b>crs:CoordinateReferenceSystem/ @svg:transform</b>	座標系変換のためのパラメータを指定する。

```
<rdf:RDF>
  <cc:Work rdf:about="">
    <dc:format>image/svg+xml</dc:format>
    <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
    <dc:relation rdf:resource="index.html"/>
    <crs:CoordinateReferenceSystem rdf:resource="http://purl.org/crs/84"
    svg:transform="matrix(1,0,0,-1,0,0)"/>
    <foaf:topic>
      <geo:Point geo:lat="35" geo:long="139" geo:alt="25" foaf:name="地下一階"/>
    </foaf:topic>
  </cc:Work>
</rdf:RDF>
```

なお、crs:CoordinateReferenceSystem 要素を複数記述することで、複数の地理座標系への変換ルールを記述することも問題がない。ただ、処理の簡略化の観点から、「文書中で最初に登場する crs:CoordinateReferenceSystem 要素は WGS84 座標系に対する変換ルールを記述する」こととする。

### 地理座標系との関連付け② ～ 地理情報 SVG 仕様【optional】

SVGドキュメントが「地理情報-SVGにおける地図の表現及びサービス」に準拠している場合には、①の記法のデータを以下のような構造で記述することもできる。この記法は、メタデータとしてではなく、SVGの要素として記述する。

要素・属性	記述内容
-------	------

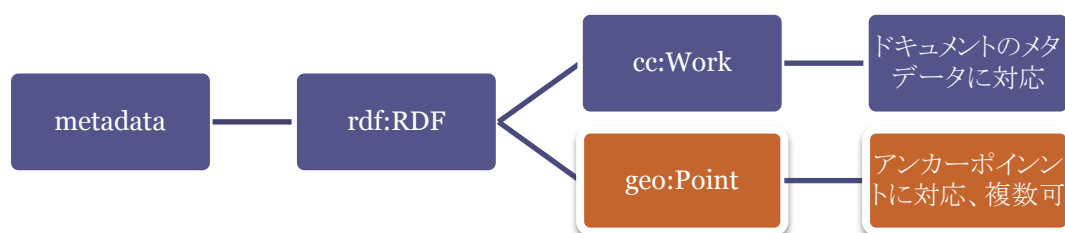
<b>/svg/globalCoordinateSystem</b>	座標系変換パラメータのための基本構造
<b>/svg/globalCoordinateSystem/@srsName</b>	WGS84 を指す URI を指定する。 通常は以下の URI を指定する <a href="http://purl.org/crs/84">http://purl.org/crs/84</a>
<b>/svg/globalCoordinateSystem/@transform</b>	座標系変換のためのパラメータを指定する。

```
<svg>
  <globalCoordinateSystem srsName="http://purl.org/crs/84" transform="matrix(1,0,0,-1,0,0)"/>
  <title>...</title>
  <desc>...</desc>
  <metadata>...</metadata>
  ...
</svg>
```

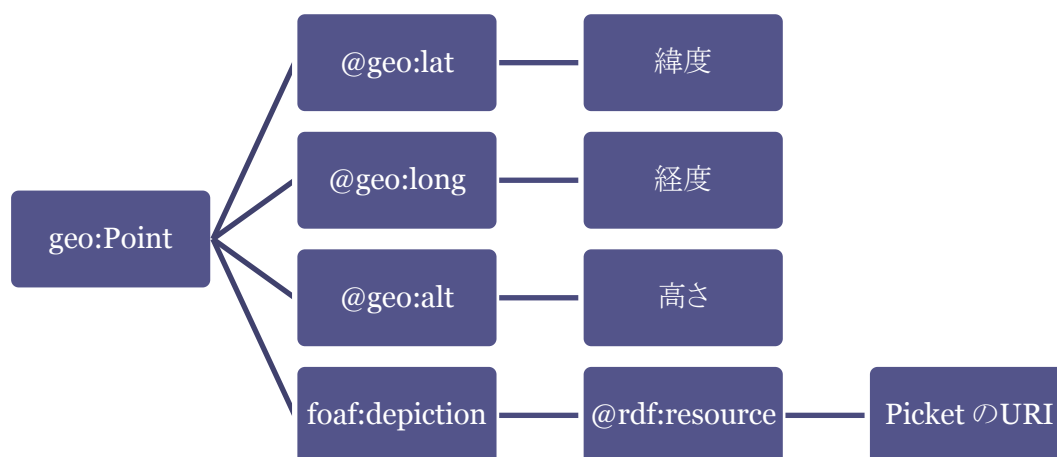
### 地理座標系との関連付け③ ~ アンカーポイントの埋め込み【optional】

SVGドキュメントに存在する Picket と、既知の緯度経度情報を持つアンカーポイントの対応関係が分かっている場合には、Picket とアンカーポイントの対応関係を宣言することによって、地理座標系と SVGドキュメント座標系との部分的な対応付けを行うことができる。

このような対応関係は①②とは異なり、ドキュメント座標系と地理座標系の関係付けではなく、複数のドキュメント座標と複数の地理座標の関連付けを表現する必要がある。そのため、RDF/XML 形式のエンコードに以下のような構造を導入することで表現するものとする。



ここで、**geo:Point** 要素は、緯度経度を持つアンカーポイントに対応する要素であり、複数の要素を挿入することが可能である。さらに、**geo:Point** 要素は以下のような構造を持ち、詳細な情報を保持する。



各要素・属性の詳細は以下の通りである。

(スタンドアローン形式の基本構造)

要素・属性	記述内容
<b>geo:Point</b>	アンカーポイントに対応する基本構造
<b>geo:Point/@geo:lat</b>	アンカーポイントの WGS84 緯度を度単位の数値で指定する
<b>geo:Point/@geo:long</b>	アンカーポイントの WGS84 経度を度単位の数値で指定する

<b>geo:Point/@geo:alt</b>	アンカーポイントの地上高をメートル単位の数値で指定する ※省略時は文書に設定された高さ情報を代理の値とする
<b>geo:Point/foaf:depiction</b>	アンカーポイントに対応する Picket に対応する基本構造
<b>geo:Point/foaf:depiction/@rdf:resource</b>	Picket の描画要素を指す URI を指定する。

アンカーポイントの種類及び属性項目を class 属性として定義する。

use 要素の種類	
名称	SVG 定義
変換に用いたアンカーポイント	calc

(参考)アンカーポイントの付加的情報

アンカーポイントに対しては、上記の基本構造だけでは表現しきれない、さまざまな情報が付随することが予想される。このような情報を保持するために、**geo:Point** 要素に対しては、任意の属性や要素を追加しても問題ないものとする。以下の表では、アンカーポイントに名称・説明・種別があると仮定した場合のエンコードの一例を、参考情報として提示する。

(参考:スタンドアロン形式の付加的情報)

要素・属性	記述内容
<b>geo:Point/@foaf:name</b>	アンカーポイントの名称を文字列として記述する。
<b>geo:Point/@dc:description</b>	アンカーポイントに関する注釈を記述する

<b>geo:Point/dcterms:conformsTo/@rdf:resource</b>	アンカーポイントが何らかの URI で規定される標準やガイドラインに準拠している場合、その URI を記述する
---------------------------------------------------	---------------------------------------------------------

(参考)アンカーポイントを外部参照する場合の構造

上記の基本構造では、アンカーポイントに関する情報をすべて文書内にエンコードする手法を定義している。一方で、外部で何らかの方法で定義されるアンカーポイントの情報を参照するような情報の提供形態も考えられる。

このような情報の提供形態はアンカーポイントの運用と併せて最適解を検討する必要があるものの、ここでは参考情報として、アンカーポイントの情報を外部参照する場合を想定したエンコード例を提示する。

(参考:外部参照形式の基本構造)

要素・属性	記述内容
<b>geo:Point</b>	アンカーポイントに対応する基本構造
<b>geo:Point/@rdf:resource</b>	URI によって一意に特定される、あるいは、外部ドキュメントの内部で規定されるようなアンカーポイントを参照する場合、当該アンカーポイントの URI を記述する
<b>geo:Point/foaf:depiction</b>	アンカーポイントに対応する Picket に対応する基本構造
<b>geo:Point/foaf:depiction/@rdf:resource</b>	Picket の描画要素を指す URI を指定する。

表に示すように、geo:Point および foaf:depiction による基本構造は変わらない。具体的な地理座標を表現する geo:lat や geo:long といった属性の代わりに、@rdf:resource 属

性を使用することで、外部で定義されるアンカーポイントを参照する、というのがこの方式の骨子である。

```
<svg>
  <metadata>
    <rdf:RDF>
      <!-- スタンドアロン形式 -->
      <geo:Point geo:lat="35.00" geo:long="139">
        <foaf:depiction rdf:resource="#anchor1"/>
      </geo:Point>
      <!-- スタンドアロン形式と付加情報 -->
      <geo:Point geo:lat="35.00" geo:long="139" foaf:name="東側出入り口前基準点">
        <foaf:depiction rdf:resource="#anchor2"/>
        <dcterms:conformsTo rdf:resource="{uri-to-specify-standard}"/>
      </geo:Point>
      <!-- 外部参照 -->
      <geo:Point rdf:resource="{uri-to-specify-anchorpoint}">
        <foaf:depiction rdf:resource="#anchor3"/>
      </geo:Point>
    </rdf:RDF>
  </metadata>
  <defs>...</defs>
  <use xml:id="anchor1" x="120" y="200" xlink:href="#icon" class="calc"/>
  <use xml:id="anchor2" x="220" y="250" xlink:href="#icon" class="calc"/>
  <use xml:id="anchor3" x="320" y="300" xlink:href="#icon"/>
</svg>
```

### 3. HTML【建物情報】

建物情報は、HTML に準拠した文字情報として記述される。以下に建物情報の一例を示す。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
  <head>
    <title>丸の内ビルディング</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
    <meta name="DC.creator" content="作成者の文字情報" />
    <meta name="DC.rights" content="権利者の文字情報" />
    <meta name="DCTERMS.created" content="2009-09-18" />
    <meta name="DCTERMS.modified" content="2009-09-29" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="geo.position" content="35.68118264026908;139.76376950740814" />
  </head>
  <body>
    <h1>丸の内ビルディング</h1>

    <h2>基本情報</h2>
    <dl>
      <dt>名称</dt>
      <dd><ruby><rb>丸の内ビルディング</rb><rp></rp><rt>マルノウチビルディング
</rt><rp></rp></ruby></dd>

      <dt>通称</dt>
```

```
<dd><ruby><rb>丸ビル</rb><rp>( </rp><rt>マルビル</rt><rp></rp></ruby></dd>

<dt>所在地郵便番号</dt>
<dd>100-6324</dd>

<dt>所在地</dt>
<dd>東京都千代田区丸の内 2-4-1</dd>

<dt>位置</dt>
<dd>緯度:<span>35.68118264026908</span> 経度:<span>139.76376950740814</span></dd>

<dt>総階数</dt>
<dd>地上 37 階・地下 4 階</dd>

<dt>提供フロア数</dt>
<dd>41</dd>
</dl>

<h2>フロア情報へのリンク</h2>
<ul>
<li><a href="b2.svg" rel="フロア図">地下二階</a></li>
<li><a href="b1.svg" rel="フロア図">地下一階</a></li>
<li><a href="f1.svg" rel="フロア図 基準階">一階</a></li>
<li><a href="f2.svg" rel="フロア図">二階</a></li>
<li><a href="network.csv" rel="ネットワーク">ネットワーク</a></li>
</ul>

<h1>この文書についての情報</h1>
<dl>
<dt>ライセンス</dt>
<dd>
<a rel="license" href="http://creativecommons.org/licenses/by/2.1/jp/">

</a>
</dd>
<dt>作成者</dt>
<dd>作成者に関する文字情報。電話・email などの情報。必要に応じて、ここで hCard 等の構造化データを記述
する。</dd>
<dt>権利者</dt>
<dd>権利者に関する文字情報。電話・email などの情報。必要に応じて、ここで hCard 等の構造化データを記述
する。</dd>
<dt>バージョン</dt>
<dd>2009-09-18 初版作成、2009-09-29 改訂。</dd>

</dl>
<h1>更新履歴</h1>
<div>この文書についての更新履歴を列举／追記</div>

</body>
</html>
```

## ① HTML の基本構造

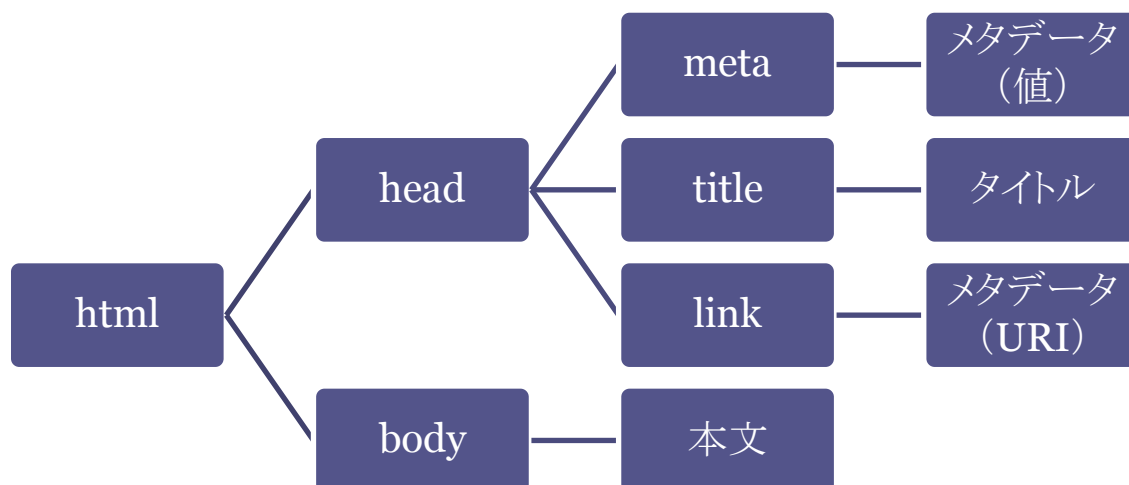
---

HTML の基本的な構造についてのガイドラインを提示する。

### 基本構造【recommended】

一般的な HTML に準拠することを想定し、HTML 仕様で規定される基本構造以上の制約は特に規定しない。レイアウトやデータ作成者による独自情報の付加など状況にあわせて、任意の構造をとって問題ない。

なお、HTML 仕様の想定する一般的な基本構造は以下のようなものである。



## ② HTML のメタデータ

HTML のメタデータとして、以下のような項目を記述することを推奨する。

### HTML ドキュメントのタイトル【required】

要素・属性	記述内容
<b>/html/head/title/text()</b>	文書のタイトル。 建物の名称を含んでいること

```
<head>  
  <title>丸の内ビルディング</title>  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  <meta name="geo.position" content="35.68118264026908;139.76376950740814"/>  
</head>
```

ここで指定する文書タイトルは、一般的な HTML に対して要求されるもので、ブラウザのタイトルバーでの表示、ブックマーク等に利用される汎用のものである。「建物の名称」だけを記述するものでなくともかまわない。

### ドキュメントの作成日時【optional】

要素・属性	記述内容
<b>/html/head/meta</b>	作成日時のための基本構造
<b>/html/head/meta/@name</b>	固定文字列「DCTERMS.created」を設定する
<b>/html/head/meta/@content</b>	作成日時を ISO8601 に準拠した文字列として記述する

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
```

```

<head>
  <title>丸の内ビルディング</title>
  <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
  <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
  <meta name="DC.creator" content="作成者の文字情報" />
  <meta name="DC.rights" content="権利者の文字情報" />
  <meta name="DCTERMS.created" content="2009-09-18" />
  <meta name="DCTERMS.modified" content="2009-09-29" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="geo.position" content="35.68118264026908;139.76376950740814" />
</head>

```

なお、本ガイドラインは Dublin Core Metadata Initiative の提唱するメタデータ記法に準拠したもので、本方式で記述する場合には以下の 1 行を合わせて記述することが推奨されている。

```
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
```

ドキュメントの作成日時は、別途 HTTP ヘッダ等に挿入されていたり、本文に記述されたりする場合もあるが、それらの情報が同内容に維持されることが望ましい。

## ドキュメントの最終更新日時【optional】

要素・属性	記述内容
<b>/html/head/meta</b>	最終更新日時のための基本構造
<b>/html/head/meta/@name</b>	固定文字列「DCTERMS.modified」を設定する
<b>/html/head/meta/@content</b>	最終更新日時を ISO8601 に準拠した文字列として記述する

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
  <head>
    <title>丸の内ビルディング</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
    <meta name="DC.creator" content="作成者の文字情報" />
    <meta name="DC.rights" content="権利者の文字情報" />
    <meta name="DCTERMS.created" content="2009-09-18" />
    <meta name="DCTERMS.modified" content="2009-09-29" />
  </head>

```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="geo.position" content="35.68118264026908;139.76376950740814"/>
</head>
```

なお、本ガイドラインは Dublin Core Metadata Initiative の提唱するメタデータ記法に準拠したもので、本方式で記述する場合には以下の 1 行を合わせて記述することが推奨されている。

```
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
```

ドキュメントの最終更新は、別途 HTTP ヘッダ等に挿入されていたり、本文に記述されたりする場合もあるが、それらの情報が同内容に維持されることが望ましい。

## ドキュメントの作成者情報【optional】

要素・属性	記述内容
<b>/html/head/meta</b>	作成者のための基本構造
<b>/html/head/meta/@name</b>	固定文字列「DC.creator」を設定する
<b>/html/head/meta/@content</b>	作成者に関する情報を自由記述する

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
  <head>
    <title>丸の内ビルディング</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
    <meta name="DC.creator" content="作成者の文字情報" />
    <meta name="DC.rights" content="権利者の文字情報" />
    <meta name="DCTERMS.created" content="2009-09-18" />
    <meta name="DCTERMS.modified" content="2009-09-29" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="geo.position" content="35.68118264026908;139.76376950740814" />
  </head>
```

なお、本ガイドラインは Dublin Core Metadata Initiative の提唱するメタデータ記法に準拠したもので、本方式で記述する場合には以下の 1 行を合わせて記述することが推奨されている。

```
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
```

SVG での RDF/XML による作成者情報と異なり、meta によるデータは構造化されていない。そのため、作成者の電話・メール・ホームページといった情報については、必要に応じて本文でフォローすることが望ましい。

## ドキュメントの権利者情報【optional】

要素・属性	記述内容
<b>/html/head/meta</b>	権利者のための基本構造
<b>/html/head/meta/@name</b>	固定文字列「DC.rights」を設定する
<b>/html/head/meta/@content</b>	著作権保持者に関する情報を自由記述する

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
  <head>
    <title>丸の内ビルディング</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
    <meta name="DC.creator" content="作成者の文字情報" />
    <meta name="DC.rights" content="権利者の文字情報" />
    <meta name="DCTERMS.created" content="2009-09-18" />
    <meta name="DCTERMS.modified" content="2009-09-29" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="geo.position" content="35.68118264026908;139.76376950740814" />
  </head>
```

なお、本ガイドラインは Dublin Core Metadata Initiative の提唱するメタデータ記法に準拠したもので、本方式で記述する場合には以下の 1 行を合わせて記述することが推奨されている。

```
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
```

SVG での RDF/XML による作成者情報と異なり、meta によるデータは構造化されていない。そのため、権利者の電話・メール・ホームページといった情報については、必要に応じて本文でフォローすることが望ましい。

#### ドキュメントの代表点メタデータ【optional】

要素・属性	記述内容
<b>/html/head/meta</b>	権利者のための基本構造
<b>/html/head/meta/@name</b>	固定文字列「geo.position」を設定する
<b>/html/head/meta/@content</b>	緯度経度を記述する。 [経度];[緯度] 形式。

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
  <head>
    <title>丸の内ビルディング</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
    <meta name="DC.creator" content="作成者の文字情報" />
    <meta name="DC.rights" content="権利者の文字情報" />
    <meta name="DCTERMS.created" content="2009-09-18" />
    <meta name="DCTERMS.modified" content="2009-09-29" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="geo.position" content="35.68118264026908;139.76376950740814" />
  </head>
```

建物の属性情報については、本文での言及と並行して、'meta' 要素での記述を併用することで、主に機械処理を容易にすることが可能である。一例として、建物の属性情報のうち、「緯度経度」に関しては 'meta' 要素で記述する方法が普及しており (Geotagging)、上記の例は Geo Tag 仕様にて規定されている記法を導入したものである。

これ以外の属性情報については、対応するメタデータが普及しておらず、特にメタデータとしての記述にガイドラインは設けない。ただし、主にデータ作成者とデータ利用者の間の利便性の面から、以下のようなアドホックなメタデータを提供しても問題ない。

```
<head>
  <title>丸の内ビルディング</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="geo.position" content="35.68118264026908;139.76376950740814"/>
  <meta name="建物正式名" content="丸の内ビルディング"/>
  <meta name="建物正式名カナ" content="マルノウチビルディング"/>
  <meta name="建物通称名" content="丸ビル"/>
  <meta name="住所" content="東京都千代田区丸の内 2-4-1"/>
  <meta name="基準階" content="1階"/>
  <meta name="総階数" content="地上 37 階・地下 4 階"/>
  <meta name="提供フロア数" content="41"/>
</head>
```

### ③ HTML の本文

HTML 本文についてのガイドラインを導入する。

#### ドキュメントの構造化【required】

「抽象モデル」で要求される各項目を記述・提供することが求められる。これは、主に文書の閲覧者に向けて整備するもので、人間にとって理解できる形式であることが重要である。

ただし、完全な自由文で記述してしまうと、将来的な属性情報の再利用・機械処理について技術的な難易度が上がってしまう。自由文と併置する形である程度構造化された「表」「リスト」のような形式で属性情報を記述しておくことで、データの再利用性を確保することをガイドラインとする。

構造化されていない例 → 別途構造化データを併記することを推奨

```
<h2>基本情報</h2>
<div>
  <p>丸の内ビルディング(通称:丸ビル)は東京都千代田区丸の内 2-4-1にあるビルです。</p>
  <p>地上 37 階、地下 4 階、のべ 41 フロアです。</p>
</div>
```

リストによる構造化例

```
<h2>基本情報</h2>
<dl>
  <dt>名称</dt>
  <dd><ruby><rb>丸の内ビルディング</rb><rp>( </rp><rt>マルノウチビルディング</rt><rp></rp></ruby></dd>

  <dt>通称</dt>
  <dd><ruby><rb>丸ビル</rb><rp>( </rp><rt>マルビル</rt><rp></rp></ruby></dd>

  <dt>所在地郵便番号</dt>
  <dd>100-6324</dd>

  <dt>所在地</dt>
  <dd>東京都千代田区丸の内 2-4-1</dd>

  <dt>位置</dt>
  <dd>緯度:<span>35.68118264026908</span> 経度:<span>139.76376950740814</span></dd>

  <dt>基準階</dt>
  <dd>1 階</dd>

  <dt>総階数</dt>
  <dd>地上 37 階・地下 4 階</dd>

  <dt>提供フロア数</dt>
  <dd>41</dd>
</dl>
```

表による構造化例

```
<h2>基本情報</h2>
<table>
  <tr>
    <th>名称</th>
    <td><ruby><rb>丸の内ビルディング</rb><rp>( </rp><rt>マルノウチビルディング</rt><rp></rp></ruby></td>
  </tr>
  <tr>
    <th>通称</th>
    <td><ruby><rb>丸ビル</rb><rp>( </rp><rt>マルビル</rt><rp></rp></ruby></td>
  </tr>
  <tr>
    <th>所在地郵便番号</th>
    <td>100-6324</td>
  </tr>
  <tr>
    <th>所在地</th>
    <td>東京都千代田区丸の内 2-4-1</td>
  </tr>
  <tr>
    <th>位置</th>
    <td>緯度:<span>35.68118264026908</span> 経度:<span>139.76376950740814</span></td>
  </tr>
  <tr>
    <th>基準階</th>
    <td>1 階</td>
  </tr>
  <tr>
    <th>総階数</th>
    <td>地上 37 階・地下 4 階</td>
  </tr>
  <tr>
    <th>提供フロア数</th>
    <td>41</td>
  </tr>
</table>
```

```
</table>
```

構造化の手法によらず、ある属性の値が単独でマークアップされるような形式を推奨する。

### 構造化データへの付加情報【optional】

構造化されたデータに対して、マークアップされた値が何の属性の値なのかを表現するために、`class` 属性を使って区別する方法を推奨する。

*This attribute assigns a class name or set of class names to an element. Any number of elements may be assigned the same class name or names. Multiple class names must be separated by white space characters.*

#### 7.5.2 Element identifiers: the id and class attributes – HTML 4.0.1

```
<tr>
  <th>名称</th>
  <td><ruby><rb class="建物正式名称">丸の内ビルディング</rb><rp></rp><rt class="建物正式名称カナ">マルノウチビルディング</rt><rp></rp></ruby></td>
</tr>
```

このような `class` 属性 で使用する単語については、あらかじめリストを作成してそれらを使用するような取り決めが必要である。ただし、一部の Web ブラウザでは(主に CSS スタイルシートの制約から)、日本語文字列を `class` 属性に使用できない、といった制約がある。たとえば以下のような表形式で「単語」「対応する建物属性」「データ型」の対応付けを管理して、マークアップに利用することが望ましい。

class 属性に指定する単語(例)	対応する建物属性	マークアップ部のデータ型
<b>building_name</b>	建物正式名	文字列
<b>building_name_kana</b>	建物正式名カナ	文字列(カナ)
<b>building_common_name</b>	建物通称名	文字列
<b>building_common_name_kana</b>	建物通称名カナ	文字列(カナ)
<b>building_address</b>	建物住所	文字列
<b>latitude</b>	建物代表点緯度	度単位の数値
<b>longitude</b>	建物代表点経度	度単位の数値
<b>building_floor_base</b>	基準階	文字列
<b>building_floor_desc</b>	総階数	文字列

**building\_floor\_number**

提供フロア数

整数値

### ハイパーリンクへの付加情報【optional】

HTML ドキュメントと外部ドキュメントの関連付けは、一般的にハイパーリンクによって記述される。このリンク関係に **rel** 属性を使って付加情報を与える方法を推奨する。なお、**rel** 属性は **HTML4.0.1** にて以下のように定義されており、複数の付加情報を付与することが可能である。

*This attribute describes the relationship from the current document to the anchor specified by the href attribute. The value of this attribute is a space-separated list of link types.*

#### 12.2 The A Element – HTML 4.0.1

```
<h2>フロア情報へのリンク</h2>
<ul>
  <li><a href="b2.svg" rel="フロア図">地下二階</a></li>
  <li><a href="b1.svg" rel="フロア図">地下一階</a></li>
  <li><a href="f1.svg" rel="フロア図 基準階">一階</a></li>
  <li><a href="f2.svg" rel="フロア図">二階</a></li>
  <li><a href="network.csv" rel="ネットワーク">ネットワーク</a></li>
</ul>

<h1>この文書についての情報</h1>
<dl>
  <dt>ライセンス</dt>
  <dd>
    <a rel="license" href="http://creativecommons.org/licenses/by/2.1/jp/">
      
    </a>
  </dd>
</dl>
```

rel 属性に指定する単語(例)	用途	リンク先
<b>license</b>	リンク先がこのドキュメントの採用したライセンスであることを示す	<b>Creative Commons</b> などのライセンスを定義した <b>URL</b>
<b>フロア図</b>	リンク先が、このドキュメントの建物に関連するフロア図であることを示す	フロア図の <b>SVG</b> ドキュメントの <b>URL</b>

基準階	リンク先が、このドキュメントの建物に関連するフロア図で、かつ、基準階を指すことを示す	フロア図の SVG ドキュメントの URL
ネットワーク	リンク先が、このドキュメントの建物に関連するネットワークであることを示す。	ネットワークの CSV ドキュメントの URL

なお、表中の `license` という値は、**Microformats** にて標準化されており **Creative Commons** のライセンス宣言等に広く利用されている。

## 4. CSV【ネットワーク情報】

ネットワーク情報はフロア情報とは独立し、CSV に準拠した形式で記述される。これにガイドラインを適用した記述例は以下のような形式となる。この実例に沿ってガイドラインの定義を行う。

```
./index.html
link_0000000001,./1f.svg#id_0000000001,1F,./1f.svg#id_0000000002,1F,walk,barrier,both,no-roof,3
link_0000000002,./1f.svg#id_0000000003,1F,./2f.svg#id_0000000001,2F,elevator,barrier-free,both,r
oof,5
```

### ①CSV の基本構造

CSV の基本的な構造についてのガイドラインを提示する。

#### 基本構造【recommended】

一般的な CSV に準拠することを想定し、建物毎に1ファイルを基本とし、カンマあり CSV 形式として保持する。

シリアライズ方法については、あらかじめリストを作成してそれらを使用するような取り決めが必要である。流通フォーマットでは、シリアライズの基本ルールとして、

項目	記述箇所	項目詳細
建物情報	1 行目	(この CSVドキュメントへのハイパーリンクを設置しているであろう)建物情報への URL を指定する。
ネットワーク情報	2 行目以降	リンク ID 起点ノード ID 起点ノードフロア階 終点ノード ID 終点ノードフロア階 種別 バリア 進行方向 屋根 距離

とし、具体的な記述方法を後述する。

## ②CSV のメタデータ

CSV として表現される各要素に関してガイドラインを設定する。

### 建物情報へのリンク【required】

ネットワーク情報と建物情報は双方向で関連づけることとする。記述方法(例)については、以下の通りである。

表現方法	記述例
./建物 HTML ファイル名	./ index.html

### 起点ノード ID 及び終点ノード ID【required】

空間参照系データで定義しているノード ID は、フロア内で一意の値となるが、ネットワークを表現する場合、流通フォーマットとしての連携パターンとして、<同一フロア内のリンク>、<同一建物内で異なるフロア間のリンク>があることを踏まえると、起点ノード ID 及び終点ノード ID は同一建物内でユニーク性を担保する必要がある。記述方法については、以下の通りとする。

表現方法	記述例
./ファイル名#ノード ID	./1f.svg#id_0000000001

# 同一フロア内のリンク

```
link_0000000001, ./1f.svg#id_0000000001,1F, ./1f.svg#id_0000000002,1F,walk,barrier,both,no-roof,3
```

# 同一建物内で異なるフロア間のリンク

```
link_0000000002, ./1f.svg#id_0000000003,1F, ./2f.svg#id_0000000001,2F,elevator,barrier-free,both,r  
oof,5
```

サービス事業者が、屋外道路ネットワークとの接続や、異なる建物間での接続を行う場合、ID がバッティングする可能性があるため、下記のような表現に修正する必要がある。

表現方法	記述例
------	-----

../フォルダ名/ファイル名#ノード ID      ../jr\_tokystation/1f.svg#id\_0000000001

### Link の メタ情報【required】

Link は ID と付随する属性情報によって表現する。Link 及び各属性項目の CSV 定義は以下の通りとする。

項目	種類	CSV 定義
リンク ID	(文字列)	(リンク ID)
起点ノード ID	(文字列)	(起点ノード ID)
起点ノードフロア階	(文字列)	(文字列)
終点ノード ID	(文字列)	(終点ノード ID)
終点ノードフロア階	(文字列)	(文字列)
種別	通常通路	walk
	動く歩道	auto-walk
	階段	stairs
	エスカレータ	escalator
	エレベータ	elevator
	スロープ	slope
バリア	あり	barrier
	なし	barrier-free

進行方向	起点ノード ID→終点ノード ID	fromto
	終点ノード ID→起点ノード ID	tofrom
	両方向	both
屋根	あり	roof
	なし	no-roof
距離	(メートル)	(距離)

```
./index.html
link_000000001,./1f.svg#id_000000001,1F,./1f.svg#id_000000002,1F,walk,barrier,both,no-roof,3
link_000000002,./1f.svg#id_000000003,1F,./2f.svg#id_000000001,2F,elevator,barrier-free,both,r
oof,5
```