1

Créer des documents XML

La spécification XML définit comment écrire un document au format XML. XML n'est pas un langage en lui-même mais, en revanche, un document XML est écrit dans un langage à balises spécifique respectant la spécification XML. Il peut, par exemple, exister des langages spécifiques permettant de décrire des données généalogiques, chimiques ou commerciales, grâce auxquels vous pouvez créer vos propres documents XML.

Chaque langage à balises spécifique créé à partir de la spécification XML doit respecter la grammaire sous-jacente de XML ; c'est donc par là que nous commencerons ce livre. Dans ce chapitre, vous apprendrez les règles d'écriture des documents XML, quel que soit le langage à balises que vous utiliserez ensuite.

Officiellement, les langages à balises spécifiques créés avec XML sont appelés *applications XML*. En d'autres termes, ces langages – XSLT, RSS, SOAP, etc. – sont des applications de XML. Cependant, pour moi, une application est plutôt un programme logiciel complet – comme Photoshop – et je trouve donc ce terme si peu précis que j'éviterai de l'utiliser.

Outils de création des documents XML

Comme pour HTML, vous pouvez utiliser n'importe quel éditeur ou traitement de texte pour écrire du XML. Il existe également de nombreux éditeurs spécialisés, dotés de fonctionnalités spécifiques, comme la validation automatique en cours de frappe (voir Annexe A).

Je supposerai désormais que vous savez créer de nouveaux documents, en ouvrir d'anciens et les sauvegarder. Assurez-vous simplement que tous vos documents XML sont sauvegardés avec l'extension `.xml`.

Exemple de document XML

Les documents XML, comme leurs homologues HTML, sont formés de balises et de données. Une grosse différence entre les deux, cependant, est que les balises utilisées par un document XML sont créées par son auteur. Une autre est qu'un document XML stocke et décrit des données ; il ne fait rien de plus, alors qu'un document HTML décrit comment les afficher.

Comme le montre la Figure 1.1, un document XML devrait être facile à comprendre car les noms choisis pour ses balises devraient décrire les données qu'elles délimitent.

La première ligne du document, `<?xml version="1.0"?>` est la *déclaration XML* qui indique la version de XML utilisée. La suivante, `<merveille>`, débute la partie des données du document et s'appelle l'*élément racine*. Dans un document XML, il ne peut y avoir qu'un seul élément racine.

Les trois lignes suivantes sont appelées *éléments fils* et détaillent l'élément racine.

```
<nom>Colosse de Rhodes</nom>
<lieu>Rhodes en Grèce</lieu>
<hauteur unité="mètres">32</hauteur>
```

Le dernier élément fils, `hauteur`, contient un *attribut* nommé `unité` qui sert à stocker l'unité de mesure. Les attributs sont utilisés pour ajouter des informations supplémentaires à un élément sans ajouter de texte à l'élément lui-même.

Enfin, le document se termine par la balise fermante de l'élément racine, `</merveille>`.

Il s'agit d'un document XML complet et valide.

La Figure 1.2 étend le document de la Figure 1.1 pour pouvoir gérer plusieurs éléments `<merveille>`. Pour ce faire, on crée un nouvel élément racine, `<anciennes_merveilles>`, qui pourra contenir autant d'éléments `<merveille>` que nécessaire. Ce document précis contient des informations sur le colosse de Rhodes et sur la grande pyramide de Kheops, située à Gizeh, en Égypte, et qui mesure 138 mètres de haut.

xml

```xml
<?xml version="1.0"?>
<merveille>
  <nom>Colosse de Rhodes</nom>
  <lieu>Rhodes en Grèce</lieu>
  <hauteur unité="mètre">32</hauteur>
</merveille>
```

Figure 1.1 *Document XML décrivant l'une des sept merveilles du monde antique, le colosse de Rhodes. Ce document contient le nom de la merveille, ainsi que son emplacement et sa hauteur en mètres.*

xml

```xml
<?xml version="1.0"?>
<anciennes_merveilles>
  <merveille>
    <nom>Colosse de Rhodes</nom>
    <lieu>Rhodes en Grèce</lieu>
    <hauteur unité="mètre">32</hauteur>
  </merveille>
  <merveille>
    <nom>Grande pyramide de Kheops</nom>
    <lieu>Gizeh en Égypte</lieu>
    <hauteur unité="mètres">138</hauteur>
  </merveille>
</anciennes_merveilles>
```

Figure 1.2 *Version étendue de la Figure 1.1 pour pouvoir décrire plusieurs merveilles.*

```xml
<?xml version="1.0"?>
<merveille>
  <nom>Colosse de Rhodes</nom>
</merveille>
```

Figure 1.3 *Document XML bien formé : la première ligne n'est pas contenue dans l'élément racine car il s'agit d'une instruction de traitement qui ne fait pas partie des données XML.*

```xml
<?xml version="1.0"?>
<merveille>
  <nom>Colosse de Rhodes</nom>
  <image_principale fichier="colosse.jpg"/>
</merveille>
```

Figure 1.4 *Document XML comprenant un élément vide utilisant une balise combinée, avec une barre de fraction finale. Tous les éléments sont correctement imbriqués : aucun ne chevauche l'autre.*

```xml
<nom>Colosse de Rhodes</nom>
<Nom>Colosse de Rhodes</Nom>
```

```xml
<nom>Colosse de Rhodes</Nom>
```

Figure 1.5 *Le premier exemple est un document XML valide, bien qu'il soit source de confusion. Les deux éléments (`nom` et `Nom`) sont, en fait, totalement différents et indépendants l'un de l'autre. Le second exemple est incorrect car les balises ouvrantes et fermantes ne sont pas appariées.*

```xml
<image_principale fichier="colosse.jpg"/>
```

Figure 1.6 *Les apostrophes autour de la valeur sont obligatoires. Elles peuvent être simples ou doubles pourvu qu'elles se correspondent. Ici, la valeur de l'attribut `fichier` ne désigne pas nécessairement un fichier image : elle pourrait simplement indiquer "Photos de nos dernières vacances".*

Règles d'écriture des documents XML

La structure de XML est très régulière et prévisible. Elle est définie par un ensemble de règles dont les plus importantes sont décrites ci-dessous. Un document qui respecte ces règles est dit *bien formé* et il peut alors être utilisé de très nombreuses façons.

Il faut un élément racine

Tout document XML doit contenir un et un seul élément racine, qui contient tous les autres éléments du document. Les seules parties XML autorisées en dehors de l'élément racine (et placées avant lui) sont les commentaires et les instructions de traitement.

Toutes les balises doivent être fermées

Tout élément doit avoir une balise fermante. Les éléments vides peuvent utiliser une balise fermante séparée ou une balise combinant l'ouverture et la fermeture, avec une barre de fraction avant le chevron final (voir Figure 1.4).

Les éléments doivent être correctement imbriqués

Si l'on ouvre l'élément A, puis l'élément B, il faut d'abord fermer l'élément B avant de fermer l'élément A (voir Figure 1.4).

La casse a une importance

XML est sensible à la casse, ce qui signifie que les éléments `merveille`, `MERVEILLE` et `Merveille` sont considérés comme totalement différents.

Les valeurs doivent être entre apostrophes

La valeur d'un attribut doit toujours être placée entre des apostrophes simples ou doubles.

Éléments, attributs et valeurs

XML utilise les mêmes briques de base que HTML : les balises qui définissent les éléments, les valeurs de ces éléments et les attributs. Un *élément* XML est l'unité de base d'un document : il peut contenir du texte, des attributs et d'autres éléments. Il possède une balise ouvrante ayant un nom entouré des signes inférieur (<) et supérieur (>). Le nom, que vous pouvez choisir vous-même, doit décrire le but de l'élément et, notamment, son contenu. Un élément se termine généralement par une balise fermante, formée du même nom précédé d'une barre de fraction. L'exception à cette règle est constituée des éléments vides, qui peuvent être "autofermants" et qui seront présentés plus loin.

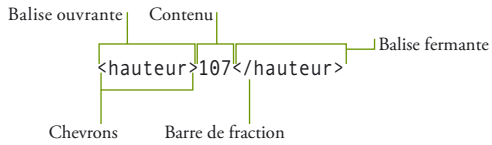

Figure 1.7 *Un élément typique est formé d'une balise ouvrante, d'un contenu et d'une balise fermante. Cet élément* `hauteur` *contient du texte.*

Les éléments peuvent avoir des *attributs*, qui sont contenus dans la balise ouvrante de l'élément et sont associés à des valeurs entre apostrophes qui décrivent leur but et leur contenu éventuel. Les informations contenues dans un attribut sont généralement considérées comme des *métadonnées*, c'est-à-dire comme des informations sur les données de l'élément, par opposition aux données elles-mêmes. Un élément peut avoir un nombre quelconque d'attributs, du moment que chacun d'eux a un nom unique.

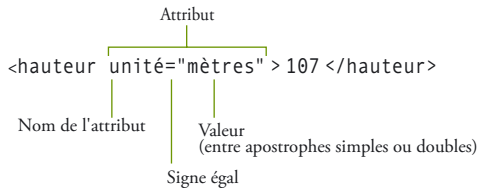

Figure 1.8 *Ici, l'élément* `hauteur` *possède un attribut* `unité` *dont la valeur est* `mètres`*. Le mot* `mètres` *ne fait pas partie du contenu de l'élément* `hauteur` *et cet attribut n'affecte pas la valeur 32 mètres à* `hauteur`*. L'attribut* `unité` *ne fait que décrire le contenu de l'élément* `hauteur`*.*

Le reste de ce chapitre est consacré à l'écriture des éléments, des attributs et des valeurs.

Espaces

Vous pouvez placer des espaces, ce qui comprend les sauts de ligne, autour des éléments XML afin de faciliter la lecture et la modification des documents. Ces espaces seront ignorées par le processeur XML, exactement comme les espaces des pages HTML sont ignorées par les navigateurs. À la Figure 1.9, par exemple, l'élément `merveille` contient trois autres éléments (`nom`, `lieu` et `hauteur`) mais n'est associé à aucun texte. Les éléments `nom`, `lieu` et `hauteur`, au contraire, contiennent du texte mais aucun autre élément. L'élément `hauteur` est le seul élément ayant un attribut. Vous pouvez remarquer que l'on a ajouté des espaces supplémentaires (en vert dans la figure) pour faciliter la lecture du code.

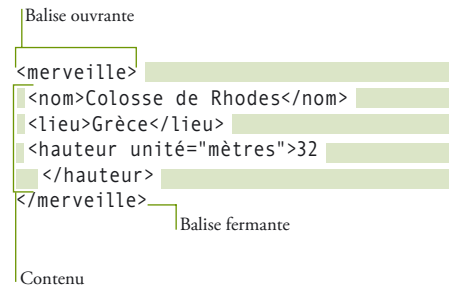

Figure 1.9 *Document XML contenant des espaces pour faciliter sa relecture.*

xml
`<?xml version="1.0"?>`

Figure 1.10 *La déclaration XML étant une instruction de traitement, pas un élément, elle n'a pas de balise fermante.*

Commencer un document XML

En principe, un document XML doit commencer par une déclaration indiquant la version de XML utilisée. Cette ligne s'appelle la *déclaration XML*.

CONSEILS

- *Le W3C a publié en 2006 une recommandation pour XML version 1.1 mais elle apporte peu de choses par rapport à la version 1.0 et elle n'est quasiment pas supportée par les outils existants.*
- *Assurez-vous de placer le numéro de version entre apostrophes doubles ou simples (peu importe, du moment qu'elles sont appariées).*
- *Les balises commençant par `<?` et se terminant par `?>` sont appelées instructions de traitement. Outre la déclaration de la version de XML, ces instructions permettent également de préciser la feuille de style à utiliser, par exemple. Les feuilles de style seront étudiées dans la Partie 2 de ce livre.*
- *Cette instruction de traitement XML peut également préciser l'encodage des caractères du document (UTF-8, ISO-8859-1, etc.). Les encodages des caractères sont présentés à l'Annexe B.*

L'élément racine

Tout document XML doit posséder un et un seul élément qui contient tous les autres. Cet élément est appelé l'*élément racine*. Dans un document HTML, l'élément racine est toujours `<HTML>` alors qu'il peut s'agir de n'importe quel nom en XML, comme `<anciennes_merveilles>` à la Figure 1.11. Aucun contenu ou élément ne peut apparaître avant la balise ouvrante de l'élément racine ni après sa balise fermante.

```xml
<?xml version="1.0"?>
<anciennes_merveilles>
</anciennes_merveilles>
```

Figure 1.11 *Document XML contenant uniquement une déclaration XML et un élément racine, ici* `<anciennes_merveilles>`.

CONSEILS

- *Attention à la casse :* `<MERVEILLE>` *est différent de* `<Merveille>` *ou* `<merveille>`.
- *Les noms des éléments et des attributs doivent être courts et évocateurs.*
- *Les noms des éléments et des attributs doivent débuter par une lettre, un blanc souligné ou par le symbole deux-points. Les noms qui commencent par les lettres* `xml` *(quelle que soit la casse) sont réservés et ne peuvent pas être utilisés.*
- *Les noms des éléments et des attributs peuvent contenir un nombre quelconque de lettres, de chiffres et de blancs soulignés, ainsi que certains caractères de ponctuation.*
- *Attention : bien que les deux-points, les tirets et les points soient autorisés dans les noms d'éléments et d'attributs, je vous déconseille de les utiliser car ils sont souvent employés dans des circonstances particulières (par exemple pour identifier respectivement les espaces de noms, les soustractions et les propriétés d'objets).*
- *Rien ne peut apparaître à l'extérieur des balises ouvrante et fermante de l'élément racine. Les seules exceptions sont les instructions de traitement.*

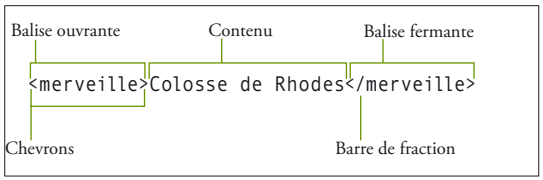

Figure 1.12 *Élément XML simple.*

```
x m l
<?xml version="1.0"?>
<anciennes_merveilles>
  <merveille>Colosse de Rhodes</merveille>
</anciennes_merveilles >
```

Figure 1.13 *Document XML ne comprenant qu'un élément fils sous la racine.*

Les éléments fils

Après avoir créé l'élément racine, on peut lui ajouter autant d'éléments fils qu'on le souhaite. L'idée est qu'il existe une relation entre la racine, ou élément père, et son élément fils. Pour les éléments fils, utilisez des noms qui identifient clairement leur contenu, afin qu'il soit plus facile ensuite de traiter l'information.

La Figure 1.12 représente un simple élément XML composé d'une balise ouvrante, d'un contenu (qui peut être du texte, d'autres éléments, voire être vide) et d'une balise fermante.

La Figure 1.13 montre comment l'élément fils est contenu dans l'élément racine.

CONSEILS

- *La balise fermante n'est jamais facultative (comme c'est parfois le cas en HTML). En XML, les éléments doivent obligatoirement avoir une balise fermante.*
- *Les règles de nommage des éléments fils sont les mêmes que pour l'élément racine. La casse a son importance, les noms doivent commencer par une lettre, un blanc souligné ou un symbole deux-points et peuvent contenir un nombre quelconque de lettres, de chiffres et de blancs soulignés. Cependant, bien que ce soit autorisé, je déconseille l'utilisation des deux-points, des tirets et des points dans les noms. En outre, vous ne pouvez pas utiliser de nom commençant par* `xml`*, quelle que soit la casse.*
- *Les noms ne sont pas obligés d'être écrits dans l'alphabet français ou latin. Cependant, si votre logiciel ne reconnaît pas certains caractères, il ne saura pas les afficher ni les traiter correctement.*
- *Si vous utilisez des noms évocateurs pour les éléments, le document XML sera plus simple à comprendre et plus facile à utiliser.*

Imbrication des éléments

Lorsque l'on crée un document XML, on souhaite parfois découper les données en parties plus petites ; pour ce faire, vous pouvez créer des éléments fils d'éléments fils, et ainsi de suite. La possibilité d'imbriquer plusieurs niveaux d'éléments fils permet d'identifier et de manipuler les différentes parties de vos données et d'établir une relation hiérarchique entre ces différentes sections.

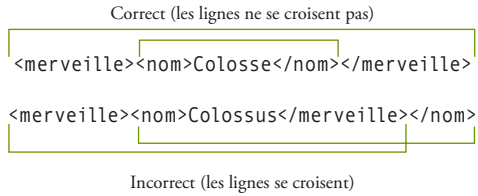

Figure 1.14 *Pour vérifier que les balises soient correctement imbriquées, il suffit de raccorder par un trait la balise ouvrante à la balise fermante correspondante. Ces traits ne doivent pas se croiser.*

CONSEILS

- *Il est essentiel que chaque élément soit totalement inclus dans un autre. En d'autres termes, vous ne pouvez pas placer la balise fermante de l'élément externe avant d'avoir fermé l'élément interne. Dans le cas contraire, le document ne sera pas bien formé et provoquera une erreur lorsqu'il sera traité par le processeur XML. La Figure 1.14 contient un exemple d'éléments mal imbriqués.*
- *Le nombre de niveaux d'imbrication n'est pas limité. La Figure 1.15 est un exemple d'imbrication à trois niveaux (racine, élément fils, éléments fils du fils).*
- *Lorsque l'on imbrique des éléments, il est fortement conseillé de les indenter afin de mieux mettre en évidence les relations entre père, fils et frères. La plupart des éditeurs XML le feront automatiquement pour vous.*

xml

```
<?xml version="1.0"?>
<anciennes_merveilles>
  <merveille>
    <nom>Colosse de Rhodes</nom>
    <lieu>Rhodes en Grèce</lieu>
    <hauteur unité="mètres">32</hauteur>
  </merveille>
</anciennes_merveilles >
```

Figure 1.15 *L'élément `merveille` est imbriqué dans `anciennes_merveilles` car c'est son fils. Les éléments `nom`, `lieu` et `hauteur` sont imbriqués dans `merveille` car ce sont ses fils.*

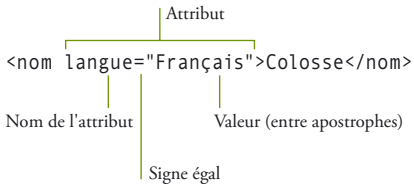

Figure 1.16 *La valeur d'un attribut doit être entourée d'apostrophes simples ou doubles.*

xml
```
<?xml version="1.0"?>
<anciennes_merveilles>
  <merveille>
    <nom langue="Français">Colosse de
    Rhodes</nom>
    <nom langue="Grec">Κολοσσός της Ρόδου</nom>
    <lieu>Rhodes en Grèce</lieu>
    <hauteur unité="mètres">32</hauteur>
  </merveille>
</anciennes_merveilles>
```

Figure 1.17 *Attributs permettant d'ajouter des informations sur le contenu d'un élément.*

Attributs

Un attribut permet de stocker des informations supplémentaires sur un élément sans ajouter de texte au contenu de l'élément lui-même. Il est formé d'une paire nom/valeur et est ajouté à la balise ouvrante d'un élément, comme le montre la Figure 1.16.

CONSEILS

- *Les noms des attributs suivent les mêmes règles que celles des noms d'éléments.*
- *Pour un même élément, tous les attributs doivent porter un nom différent.*
- *À la différence de HTML, les valeurs des attributs doivent impérativement être placées entre apostrophes simples ou doubles.*
- *Si la valeur d'un attribut contient des apostrophes doubles, il faut entourer cette valeur d'apostrophes simples (et* vice versa*).*
  Par exemple : `commentaire= 'Elle a dit "Le Colosse est tombé !"'`.
- *Comme en programmation, une bonne pratique consiste à considérer les attributs comme des métadonnées, c'est-à-dire des données sur les données. En d'autres termes, les attributs devraient servir à stocker des informations sur le contenu de l'élément, pas le contenu lui-même. C'est ce que l'on fait à la Figure 1.17.*
- *Un autre moyen de marquer et d'identifier des informations distinctes consiste à utiliser des éléments imbriqués, comme nous l'avons vu précédemment.*