Le DOCTYPE

Le DOCTYPE ou DTD (*Document Type Definition*) est l'ensemble des règles et des propriétés que doit suivre le document XML produit. Ces règles définissent généralement le nom et le contenu de chaque balise, ainsi que le contexte dans lequel elles doivent exister. Cette formalisation des éléments est particulièrement utile lorsqu'on utilise de façon récurrente des balises dans un document XML, ou que plusieurs personnes sont appelées à travailler sur le même document XML.

Les DTD sont écrites selon les prescriptions du SGML.

L'étude détaillée des DTD, dépasse de loin le cadre de cet ouvrage, mais un aperçu est cependant utile, surtout pour comprendre le fonctionnement des langages dérivés du XML (comme le XHTML), qui ne manquent pas d'utiliser ces fameux DTD.

Dans certains cas, plusieurs concepteurs peuvent se mettre d'accord pour utiliser un DTD commun pour échanger leurs données. C'est le cas du XHTML, où le DOCTYPE est signalé dans l'en-tête du document (strict, transitional ou frameset).

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

Cependant, dans la plupart des applications XML, le concepteur doit écrire sa propre définition de document.

1. Le DTD interne

On peut inclure le DOCTYPE dans le code du fichier XML. On parlera alors d'un DTD interne.

Un DTD interne suit la syntaxe suivante :

```
<!DOCTYPE élément-racine [
déclaration des composants
]>
```

Du point de vue des DTD, tous les documents XML sont élaborés avec les composants suivants :

- Les éléments, ou plus communément les balises. Exemple : la balise `<body>` en HTML.
- Les attributs qui viennent compléter les balises. Les attributs sont toujours inclus dans la balise ouvrante.
  Exemple : l'attribut **border** de la balise `<table>` en XHTML.
- Les PCDATA (*parsed character data*), chaînes de caractères qui seront affichées par l'interpréteur XML (*Parser*). Exemple : le contenu de la balise `<p>Chapitre 1 : Le XHTML</p>` en XHTML.
- Les CDATA (*character data*), chaînes de caractères qui ne sont pas prises en compte et qui ne sont donc pas affichées par le navigateur XML. Exemple, du code JavaScript dans un document XHTML.
- Les entités, sortes de raccourcis qui permettent de déclarer plusieurs paramètres utilisables en appelant simplement l'entité plus loin dans le document.

Détaillons ci-après quelques règles pour définir ces différents composants.

a. La déclaration d'un élément

Un élément se déclare dans la DTD selon la syntaxe suivante :

```
<!ELEMENT nom_de_l'élément mot-clé>
```

ou

```
<!ELEMENT nom_de_l'élément (contenu)>
```

Exemple

`<!ELEMENT h1>` pour la balise `<h1>` en XHTML.

b. Les éléments vides

Un élément vide (*empty*) se déclare :

```
<!ELEMENT nom_de_l'élément EMPTY>
```

Exemple

`<!ELEMENT img EMPTY>` pour la balise `<img>` en HTML.

c. Les éléments comprenant des caractères à afficher

Les éléments comprenant des caractères à afficher, généralement le contenu des balises, se notent :

```
<!ELEMENT nom_de_l'élément (#PCDATA)>
```

Exemple

`<!ELEMENT p (#PCDATA)>` pour la balise de paragraphe `<p>` en HTML.

d. Les éléments avec des éléments enfant

```
<!ELEMENT nom_de_l'élément (elément_enfant,élément_enfant,...)>
```

Exemple

```
<!ELEMENT <enfants (fils,fille)>
```

Lorsque des éléments enfant sont déclarés, ces éléments enfant doivent apparaître dans le même ordre dans le document XML.

En outre, les éléments enfant doivent être déclarés.

```
<!ELEMENT <enfants (fils,fille)>
<!ELEMENT fils (#PCDATA)>
<!ELEMENT fille (#PCDATA)>
```

e. Les éléments avec une seule occurrence

Les éléments enfant qui doivent apparaître seulement une fois dans le code de l'élément parent se notent :

```
<!ELEMENT nom_de_l'élément (elément_enfant)>
```

Exemple

```
<!ELEMENT adhérent (nom)>
```

L'élément enfant `nom` doit apparaître une fois et seulement une fois dans l'élément parent `adhérent`.

f. Les éléments avec une ou plusieurs occurrences

Les élément qui peuvent apparaître plusieurs fois se déclarent :

```
<!ELEMENT nom_de_l'élément (elément_enfant+)>
```

Exemple

```
<!ELEMENT adhérent (telephone+)>
```

Le signe + déclare que l'élément enfant `telephone` doit apparaître une fois mais peut aussi apparaître plusieurs fois dans l'élément parent `adhérent`.

g. Les éléments avec zéro, une ou plusieurs occurrences

Les éléments facultatifs qui peuvent ne pas apparaître (zéro occurrence) ou apparaître plusieurs fois se notent :

```
<!ELEMENT nom_de_l'élément (elément_enfant*)>
```

Exemple

```
<!ELEMENT adhérent (telephonefixe*)>
```

Le signe * déclare que l'élément enfant `telephonefixe` peut ne pas apparaître, tout comme il peut apparaître plusieurs fois dans l'élément `adhérent`.

h. Les éléments avec zéro ou une occurrence

Les éléments qui peuvent apparaître qu'une fois ou être absents se définissent :

```
<!ELEMENT nom_de_l'élément (elément_enfant?)>
```

Exemple

```
<!ELEMENT adhérent (adresseIP?)>
```

Le signe ? déclare que l'élément enfant `adresseIP` peut être absent ou apparaître une seule fois dans l'élément parent `adhérent`.

i. Les éléments alternatifs

Lorsque des options (le ou logique) sont autorisées dans les éléments enfants, celles-ci se déclarent :

```
<!ELEMENT nom_de_l'élément ((elément_enfant|élément_enfant))>
```

Exemple

```
<!ELEMENT telephone ((national|international))>
```

L'exemple déclare que l'élément parent `telephone` peut contenir l'élément enfant `national` ou l'élément enfant `international`.

Appliquons une DTD interne à notre fichier XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<famille>
<enfants>
<fils>Loïc</fils>
<fille>Marine</fille>
</enfant>
</famille>
```

Celui-ci devient :

```
<?xml version="1.0" "standalone="yes"?>
<!DOCTYPE famille [
<!ELEMENT famille (enfants?)>
<!ELEMENT enfants (fils*, fille*)>
<!ELEMENT fille (#PCDATA)>
<!ELEMENT fils (#PCDATA)>
]>
<famille>
<enfants>
<fils>Loïc</fils>
<fille>Marine</fille>
</enfants>
</famille>
```

Commentaires :

- Lorsqu'un DOCTYPE interne est défini, il faut déclarer que le fichier est indépendant (`standalone="yes"`) dans le prologue XML.
- L'élément racine est déclaré dans le début du DOCTYPE (`<!DOCTYPE famille`).
- Le contenu du DOCTYPE est encodé entre des crochets ouvrants et fermants.
- La ligne `<!ELEMENT famille (enfants?)>` déclare que la balise `<famille>` contient la balise `<enfants>`. Le signe ? prévoit le cas des familles sans enfants.
- La ligne `<!ELEMENT enfants (fils*, fille*)>` déclare que la balise `<enfants>` contient les

balises `<fils>` et `<fille>`. Le signe * est prévu pour le cas où une descendance n'est composée que d'un ou plusieurs fils ainsi que d'une ou plusieurs filles.

- Le DOCTYPE se termine par le signe >.

2. Le DTD externe

Le DTD externe suit la syntaxe suivante :

```
<!DOCTYPE élément-racine SYSTEM "nom_du_fichier.dtd">
```

Le même fichier que ci-dessus est alors :

```
<!DOCTYPE famille SYSTEM "parent.dtd">
<?xml version="1.0" standalone="no"?>
<famille>
<enfants>
<fils>Loïc</fils>
<fille>Marine</fille>
</enfants>
</famille>
```

Le fichier de DTD externe (ici dans le même répertoire) `parent.dtd` contient :

```
<!ELEMENT famille (enfants?)>
<!ELEMENT enfants (fils*, fille*)>
<!ELEMENT fille (#PCDATA)>
<!ELEMENT fils (#PCDATA)>
```

Mais il est aussi possible de faire référence à un DTD externe situé sur un autre site comme pour, par exemple, le XHTML :

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

Un document est dit valide s'il respecte les règles spécifiques de son DOCTYPE.

Il existe une autre méthode pour définir les DTD, non plus en SGML mais en XML, c'est le XML Schema.

Le XML Schema est un langage de description de format de document (*XML Schema Description*), encodé en XML, permettant de définir la structure d'un document XML. Ce fichier de description de structure tient le rôle de DTD et permet également de valider le document XML.