EPAI, Jérôme Frossard (2017)

LES FICHIERS DE TEXTE ET LEURS APPLICATIONS

Caractériser, mettre en forme et analyser des données

Texte et informatique

- En informatique on utilise beaucoup de texte, mais pas seulement du texte dans le langage ordinaire. On utilise du texte également pour :
  - Communiquer avec une machine à travers un terminal avec un interpréteur de commandes (Bash, Cmd, PowerShell, etc.).
  - Écrire le code source d'un programme (Java, JavaScript, etc.).
  - Stocker des données structurées (XML, JSON, etc.).
  - Représenter l'apparence, la structure et la mise en page du texte dans un document de texte (HTML, Open XML, ODF, TeX, etc.).
  - Échanger des données entre des programmes à travers le réseau ou le système de fichiers (HTTP, CSV, XML, JSON, etc.).

Fichier de texte

- Contient une suite de caractères codés sous forme numérique selon une certaine norme (ASCII, ANSI, UTF-8, etc.).
- Ne contient pas d'information sur la norme utilisée pour coder les caractères.
- L'émetteur et le récepteur du fichier doivent s'accorder d'une manière ou d'une autre sur la norme à utiliser.
- Certains éditeurs de texte peuvent détecter la norme d'encodage, mais des erreurs de détection sont possibles.

Éditeur de texte

- Un éditeur de texte est un programme permettant de créer et de modifier des fichiers de textes.
- Ne doit pas être confondu avec un logiciel de traitement de texte (pas de structuration ou de mise en page du texte).
- Peut avoir des fonctions avancées comme l'autocomplétion ou la coloration syntaxique pour faciliter l'écriture de code (Java, HTML, TeX, Bash, etc.).
- Exemples :
  - Atom, Visual Studio Code, Notepade++
  - Vim, Emacs
  - Eclipse, NetBeans, Visual Studio

Extraire des données

Armelle Crevoisier habite à Lurtigen, au numéro 13 de la rue Fontenal. Doriane Chappuis quant à elle, vie à Ulmiz, non loin de Morat. Son domicile se trouve au numéro 22 de la rue du Bourg. Christine Charpié a choisi d'élire domicile, plus au sud, à Semsales, dans la jolie ruelle des Tailleurs, au numéro 11. Enfin, c'est au chemin de Beaulieu 10 à Onnens que se trouve la maison de Sophie Comman.

Extraire des données

Armelle Crevoisier wohnt in Lurtigen in der Schlossstrasse 13. Doriane Chappuis ihrerseits, lebt in Ulmiz, nicht weit von Morat. Ihr Wohnsitz befindet sich in der Burgstrasse 22. Christine Charpié hat ihren Wohnsitz weiter im Süden gewählt, genauer gesagt in Semsales, in der schönen Schneidergasse 11. Zu guter Letzt, am Schönenweg 10 in Onnens befindet sich das Haus von Sophie Comman.

Sémantique et syntaxe

- Le texte de la diapositive précédente contient les nom, prénom et adresse d'un certain nombre de personnes.
- Pour extraire ces informations, il ne suffit pas d'analyser la structure du texte, mais on doit en comprendre le sen.
- La structure du texte est le domaine de la **syntaxe**.
- Le sens d'un texte est le domaine de la **sémantique**.
- D'une manière générale, il est :
  - Facile d'analyser la structure d'un texte
  - Difficile d'en comprendre le sens

Donner du sens à la syntaxe

- Il est souvent utile d'utiliser une syntaxe qui permet de décrire les relations entre les données, par exemple :
  - Carte conceptuelle :
    - Un rectangle représente un concept.
    - Une ligne entre deux rectangles représente un lien entre deux concepts.
  - Table ou **relation** :
    - Une colonne (**champ**) représente une donnée élémentaire (**attribut**) d'un élément d'un ensemble d'éléments du même type, p. ex. des personnes.
    - Une entrée de la table (**enregistrement**) met en relation les valeurs des attributs d'un élément particulier (**entité**), p. ex. le nom, le prénom et l'adresse d'une personne.

Formats et méta-formats

- Un méta-format est format générique qui peut être utilisé pour créer des formats de données spécifiques sans avoir à inventer à chaque fois une nouvelle syntaxe.
- Exemples de méta-format :
  - CSV (*Comma Separated Value*)
  - XML (e*Xtensible Markup Langage*)
  - JSON (*JavaScript Object Notation*)
  - YAML (*YAML An't Markup Language*)
  - EDN (*Extensible Data Notation*)
  - RDF (*Ressource Description Framework*)

CSV: Description

- CSV : Comma Separated Values.
- Un fichier CSV est un fichier de texte qui permet d'échanger des données sous forme de table (relation).
- Le format n'est pas spécifié de manière formelle, le caractère de séparation est en principe la virgule, mais Excel par exemple, utilise le point-virgule.
- Dans un fichier donné, le caractère de séparation doit toujours être le même.
- La forme la plus courante est décrite dans la RFC 4180 qui en définit également le type MIME (texte/csv)

CSV: Format

- Chaque ligne de la table correspond à une ligne de texte terminée par les caractères CR LF ou un caractère LF seul.
- Les valeurs d'une ligne sont séparées par un caractère de séparation (virgule, point-virgule, etc.)
- Si une valeur est une chaîne de caractères, elle peut ou non être mise en guillemets.
- Il n'y a, en principe, pas de métadonnées dans un fichier CSV, mais la première ligne peut contenir le nom des colonnes.

CSV : Avantages et inconvénients

- Avantages
  - Simplicité : le format est simple et facile à comprendre.
- Inconvénients
  - Pas de métadonnée : Les métadonnées d'un CSV se limitent aux noms des colonnes qui sont optionnelles et rien n'est prévu pour indiquer la norme d'encodage des caractères.
  - Standard faible : Il n'y a pas de parseur standard et même si de nombreux outils supportent le format, ils ne sont pas toujours compatibles entre eux.

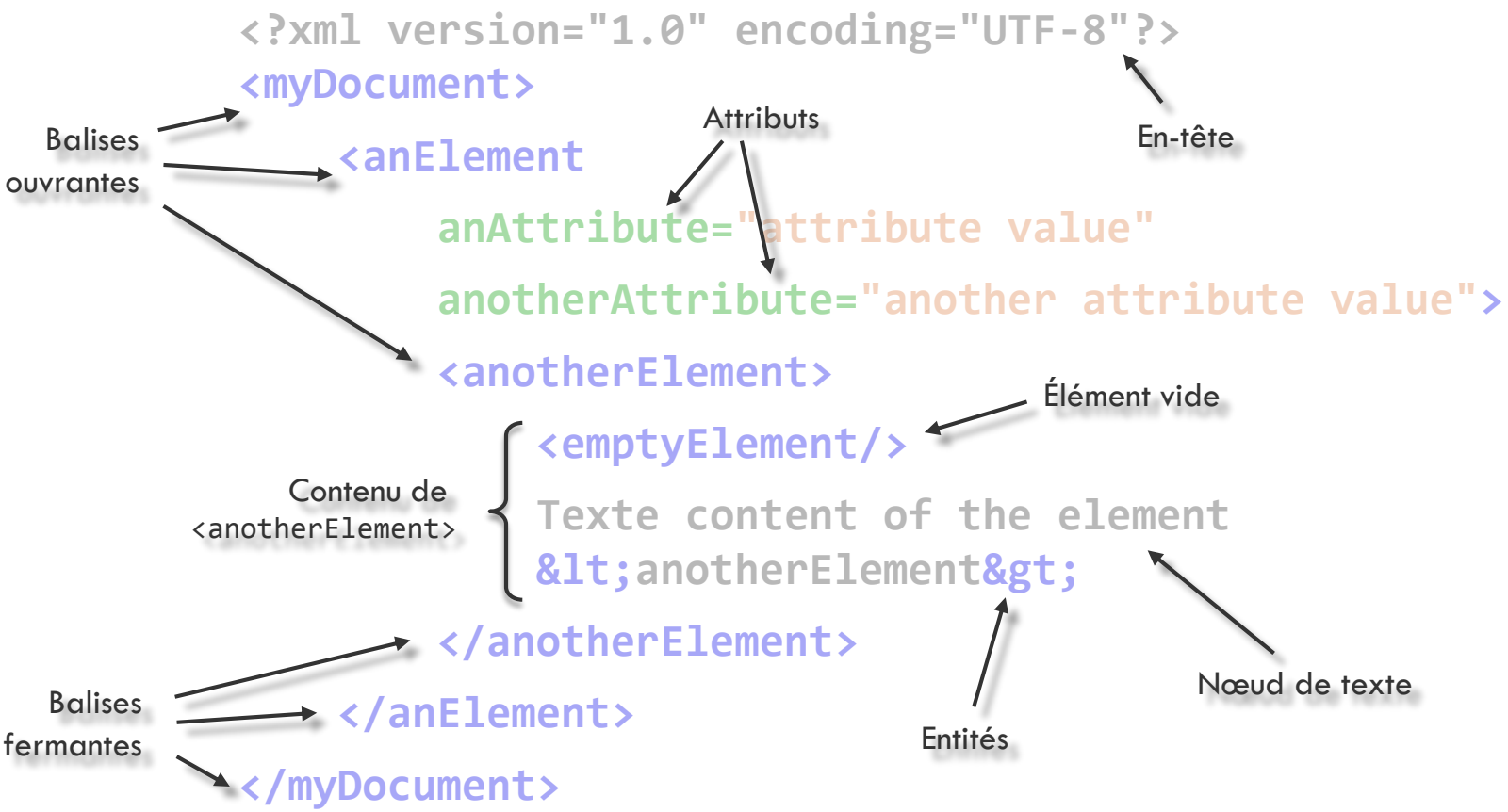
XML : Description

- XML : eXtensible Markup Langage
- **Markup Langage** : Les données sont structurées à l'aide de balises.
- **Extensible** : La spécification décrit une syntaxe générale qui peut être utilisée pour réaliser des langages spécifiques.
- Exemple de langages XML :
  - Web : XHTML, SVG
  - Documents de bureautique : Office Open XML, ODF
  - Syndication de contenus : RSS

XML : Format

- Décrit un arbre dont le tronc est le **document**, les branches sont les **éléments** et les feuille les **nœuds de texte.**
- Un élément est délimité par une **balise ouvrante** et une **balise fermante** qui ont le même nom.
- Les noms des balises doivent respecter certaines règles (commencer par une lettre, pas d'espace, etc.)
- Le contenu d'un élément est tout ce qui se trouve entre les deux balises (pas seulement le nœud de texte).
- Un élément peut avoir des **attributs** qui prennent place à l'intérieur de la balise ouvrante.
- Un en-tête permet de spécifier la version de la spécification et la norme d'encodage des caractères.

XML : Exemple


<?xml version="1.0" encoding="UTF-8"?>
<myDocument>
<anElement
anAttribute="attribute value"
anotherAttribute="another attribute value">
<anotherElement>
<emptyElement/>
Texte content of the element
<anotherElement>
</anotherElement>
</anElement>
</myDocument>
Balises ouvrantes
Attributs
En-tête
Élément vide
Contenu de <anotherElement>
Balises fermantes
Entités
Nœud de texte

XML : Validation

- Par défaut, un parseur XML vérifie qu'un document XML respecte la syntaxe générale du XML. Si c'est le cas, on dit que le document est **bien formé** (*well-formed*).
- Il est possible de fournir au parseur la description d'un langage XML (DTD ou schéma XML), pour lui permettre de vérifier la syntaxe spécifique au langage. Si c'est le cas, on dit que le document est **valide**.
- Le schéma XML ou le DTD sont des métadonnées qui définissent la grammaire d'un langage XML.
- Certaines de ces grammaires sont standardisées (XHTML, SVG, MathML, ODF, Office Open XML, etc.)

XML : Avantages et inconvénients

- Avantages
  - Standard : XML est standard et il existe donc de nombreux outils et de bibliothèques qui permettent de lire et écrire du XML.
  - Polyvalent : XML peut être utilisé dans une grande variété d'applications (échange de données, livres numérique, graphique vectoriel, syndication de contenu, etc.)
  - Auto-descriptif : les données XML se décrivent elles-mêmes (*self-describing data*) si un langage XML standardisé est utilisé.
  - Validation : il est possible de valider la syntaxe d'un langage XML à l'aide d'une grammaire (DTD ou schéma XML).
- Inconvénients
  - Verbeux : le XML ajoute beaucoup de redondance aux données.

Langages XML

- Un grand nombre de format de données standard sont basé sur le méta-format XML.
- La terminologie du XML est un peu différente, on dit :
  - métalangage plutôt que méta-format,
  - langage ou instance plutôt que format.
- Parmi les nombreux langages XML existants, on peut citer :
  - OpenXML et OpenDocument (bureautique)
  - SVG (format de graphique vectoriel)
  - XHTML (HTML avec une syntaxe XML stricte)
  - Atom, RSS (syndication de contenu)
  - MathML

Langages XML : XHTML


```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
    <head>
        <title>Première page XHTML</title>
    </head>

    <body>
        <p class="paragaph">
        Bonjour tout le monde!<br/>Je suis une page XHTML.
        </p>
    </body>
</html>
```

Langages XML : SVG


```
<?xml version="1.0" encoding="UTF-8"?>
<svg
    xmlns="http://www.w3.org/2000/svg">
    <ellipse
        cx="49" cy="49"
        rx="40" ry="40"
        style="fill:#fff066;stroke:#333333;stroke-width:4"/>
    <path
        d="m 27,60 c 1.7,18 33,19 47,2"
        style="opacity:1;fill:none;stroke:#333333;stroke-width:4"/>
    <ellipse
        cx="33" cy="41"
        rx="7" ry="7"
        style="fill:#333333;stroke:#333333"/>
    <ellipse
        cx="67" cy="41"
        rx="7" ry="7"
        style="fill:#333333;stroke:#333333"/>
</svg>
```