Module

XML

JSON

Introduction

- Communications en réseau
  - Données à sérialiser et à transmettre
    - Format XML
    - Format JSON
    - Et autres
  - Fichier de données de type XML ou JSON
    - Facilement lisible par un non informaticien
    - Sous forme de texte

Part 1 : Fichier .xml

Table des matières

Introduction

- XML
  - e**X**tensible **M**arkup **L**anguage
  - Langage à balise (Markup Language)
- But
  - Transfert, partage et stockage des données
- A ne pas confondre
  - HTML/CSS : présentation des données
- Navigateurs
  - Parser pour XML

Tags

- Pas prédéfinis
  - Le programmeur peut définir les siens
- Suffisamment descriptifs
- Exemple

```
<Teacher>
        <Name>Charlier</Name>
</Teacher>
```

DOM

- Interface
  - permettant à des programmes et à des scripts d'accéder ou de mettre à jour
    - le contenu
    - la structure
    - le style

      de documents XML
  - Le document peut ensuite être traité et les résultats de ces traitements peuvent être réincorporés dans le document tel qu'il sera présenté

SAX

- Simple API pour XML
- Interface de programmation
  - Mêmes rôles que DOM
  - Mais processus différent
    - DOM : tout le document est chargé en mémoire dans une structure
    - SAX : élément par élément

Validité

- Définition d'un document type appelé DTD (*Document Type Definition*)
  - Grammaire permettant de vérifier la conformité du fichier XML
  - Non obligatoire pour un fichier XML
    - La norme XML impose néanmoins le respect des règles de base!
- Fichier XML
  - Valide s'il est accompagné du DTD
  - Bien formé s'il n'a pas de DTD mais respectant les règles de base du XML
- DTD défini
  - Sous forme interne : grammaire incluse au sein même du fichier XML
  - Sous forme externe
    - Appel d'un fichier contenant la grammaire à partir du fichier XML
    - Accès via l'url

Validité

- Exemple

```
<page>
        <title>Bonjour à tous</title>
        <content>
                Vous êtes au bloc 3!
        </content>
        <comment> Ecrit par les professeurs</comment>
</page>
```

- Le DTD serait :

optionnel

```
<!ELEMENT page (title, content, comment?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT content (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
```

chaîne de caractères

Constituants du fichier

- Prologue
  - Contient aussi des déclarations facultatives
- Corps
  - Contenu même du document
- Commentaires et instructions de traitement
  - Dans le prologue
  - Dans le corps

Prologue

- Exemples
  - Sans DTD

```
<?xml version="1.1" encoding="ISO-8859-15"?>
<Teacher>
        <Name>Charlier</Name>
</Teacher>
```

  - Avec DTD interne

```
<?xml version="1.1" encoding="ISO-8859-15"?>
<!DOCTYPE Teacher [
        <!ELEMENT Name (#PCDATA)>
]>
<Teacher>
        <Name>Charlier</Name>
</Teacher>
```

Prologue

- Exemples
  - Avec DTD externe
    - Grammaire personnelle

```
<?xml version="1.1" encoding="ISO-8859-15"?>
<!DOCTYPE Teacher SYSTEM "MyFichier.dtd">
<Teacher>
        <Name>Charlier</Name>
</Teacher>
```

    - Grammaire publique

```
<?xml version="1.1" encoding="ISO-8859-15"?>
<!DOCTYPE rss ….>
<Teacher>
        <Name>Charlier</Name>
</Teacher>
```

Prologue

- Constitué de déclaratives obligatoires et facultatives
- Forme
  - <?xml…?>
- Plusieurs parties
  - Version de XML
  - Jeu de codage de caractères utilisé
  - Un booléen indiquant la présence d'un autre document
    - Valeur par défaut : "yes"
      - Le processeur de l'application n'attend aucune déclaration extérieure de type de document

Prologue

- Déclarative du type du document
  - DTD – Document Type Definition
  - qui définit la structure du document
  - placé en interne ou en externe
  - <!DOCTYPE…>

Corps

- Structuré comme un arbre → Racine unique
- Composé d'éléments
  - Un élément peut être vide
  - Un élément peut contenir du texte
  - Un élément peut comprendre d'autres éléments

Présentation

- Formatage d'un document xml avec un fichier de type
  - CSS (Cascade Style Sheets)
  - XLST (eXtensible Stylesheet Language Transformations)
    - Recommandé par W3C
    - Plus de possibilités
    - Permet la transformation du fichier xml en un autre format (HTML, …)

Présentation

- Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE salut SYSTEM "bonjour.dtd">
<?xml-stylesheet type="text/xsl" href="bonjour.xsl" ?>

<salut>hello monde!</salut>
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
 <html><head></head><body>
 <p align="center" style="font-family:Tahoma; font-size:48pt; color:red">
  <xsl:value-of select="." />
 </p>
 </body></html>
</xsl:template>

</xsl:stylesheet>
```

http://www.validome.org/doc/HTML_fr/xml/representation/exemplesxslt.htm

Part 2 : Fichier .json

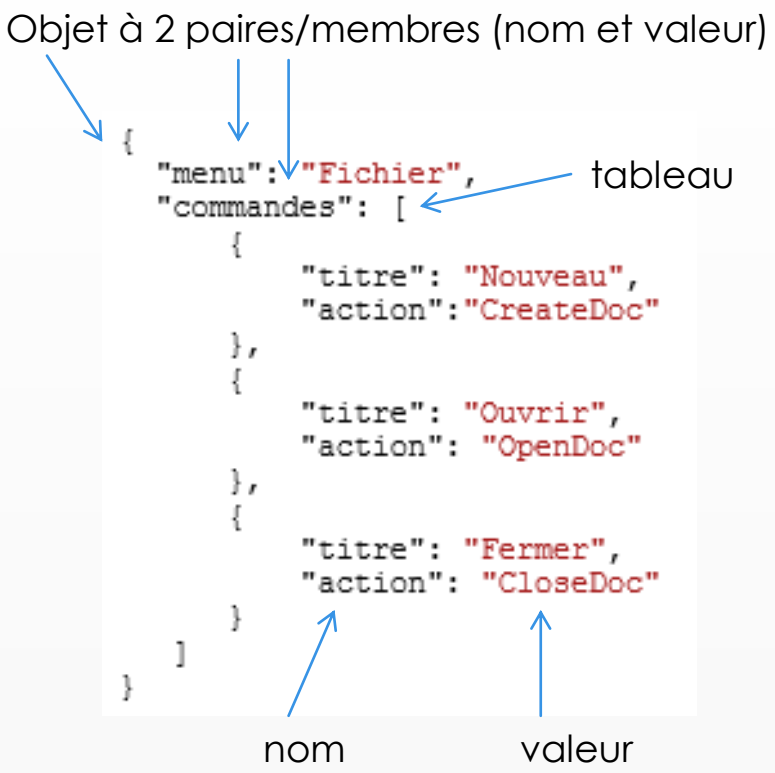
Table des matières

Introduction

- JSON: **J**ava**S**cript **O**bject **N**otation
- Basé sur un sous-ensemble de JavaScript
- Mais malgré tout, indépendant de tout langage de programmation

Exemple

```xml
<?xml version="1.0" ?>
<racine>
  <menu>Fichier</menu>
  <commandes>
     <item>
         <titre>Nouveau</titre>
         <action>CreateDoc</action>
     </item>
     <item>
         <titre>Ouvrir</titre>
         <action>OpenDoc</action>
     </item>
     <item>
         <titre>Fermer</titre>
         <action>CloseDoc</action>
     </item>
  </commandes>
</racine>
```



http://www.xul.fr/ajax-format-json.html

Comparaison

- Points communs avec XML
  - Stocker et échanger des informations sous forme de texte
  - Indépendant de tout langage
  - Forme hiérarchique
  - Lisibilité
- Différences
  - Taille plus petite
  - Pas de tag de fin
  - Plus rapide à lire et à écrire
  - Peut être parsé par la méthode eval() en JavaScript
  - Tableaux possibles
  - Aucun mot réservé