



Manipuler les données : XML et CSV

Conditions	Travail en groupe Durée : 2h	Moyens	<ul style="list-style-type: none"> • Poste informatique sous Windows • Connexion internet • Processing 1.5 • Notepad++
Prérequis	Etre initié à Processing	Niveau	Classe de Terminale S spécialité ISN
Compétences	<ul style="list-style-type: none"> • Compétences C2 : Concevoir et réaliser une solution informatique en réponse à un problème 		
Eléments du programme	<ul style="list-style-type: none"> • Représentation de l'information <ul style="list-style-type: none"> ○ Formats ○ Structuration et organisation de l'information • Algorithmes <ul style="list-style-type: none"> ○ Algorithmes simples (Rechercher un élément dans un tableau) • Langages et programmation <ul style="list-style-type: none"> ○ Types de données ○ fonctions 		

La diffusion des données doit respecter un certains nombres de règles.

- Les données doivent être documentées (les données des données = métadonnées)
- Les données doivent respecter une sémantique (exemple de la date : quel calendrier sert de repère).
- Les données doivent respecter une syntaxe.

Pour assurer l'interopérabilité, les données sont généralement accessibles par le téléchargement d'un fichier de type CSV, ou en accès direct via une API délivrant les informations sous format XML.

En quoi consiste le format CSV ? Comment traiter les données d'un fichier CSV ?

En quoi consiste le XML ? Comment traiter les données XML ?

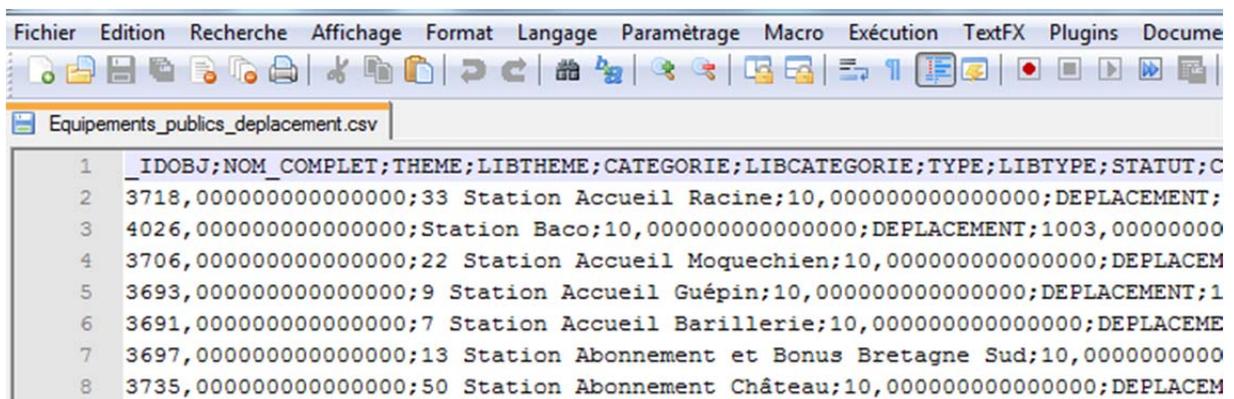
Table des matières :

1. Fichier CSV	2
Définition	2
Créer un fichier CSV	2
2. CSV avec Processing	2
3. Format XML	4
4. XML avec Processing	4
Mettre des données au format XML	4
Exemples XML avec Processing	5

1. Fichier CSV

Définition

Le format CSV (Comma Separated Values) en français, « valeurs séparées par des virgules ». Il s'agit d'un format de fichiers ouvert qui permet de stocker les données d'un tableau. Chaque ligne du fichier correspond à une ligne du tableau ; les colonnes sont en général séparées par des virgules. On peut très bien remplacer les virgules par des tabulations ou tout autre caractère. Ce format est utilisé pour échanger de manière interopérable des données de tableurs, bases de données, annuaires, etc. entre logiciels différents et/ou plateformes différentes.



Créer un fichier CSV

1. Lancer Excel ou Open calc et saisir les données comme ci-contre.
2. Enregistrer au format csv (liste.csv)
3. Ouvrir le fichier créé avec notepad++
4. Reporter et commenter le résultat

	A	B	C	D
1	NOM	PRENOM	CLASSE	SEXE
2	Lepetit	Alexandre	TS1	G
3	Gaumont	Clemence	TS1	F
4	Guerin	Sophie	TS2	F
5	Kerhuel	Alan	TS2	G
6				

2. CSV avec Processing

Le code Processing suivant permet d'extraire les données d'un fichier CSV :

```
String[] lignes = loadStrings("liste.csv");
for (int i=0; i < lignes.length; i++) {
  println("ligne n°"+i+": "+lignes[i]);
}
```

A l'aide de la [documentation](#) processing :

5. Indiquer le fonctionnement de l'instruction « loadstring() »
6. Indiquer la différence entre « String lignes » et « String[] lignes »

Lancer l'exécution de ce programme dans Processing :

7. Reporter et commenter le résultat de l'exécution de ce programme.

A l'aide de la [documentation](#) processing :

8. Indiquer le fonctionnement de l'instruction « split() »

L'objectif est d'extraire chaque élément de chaque ligne séparé par des « ; » :

9. Compléter le programme ci-dessous en utilisant la fonction split() afin d'obtenir le résultat suivant :

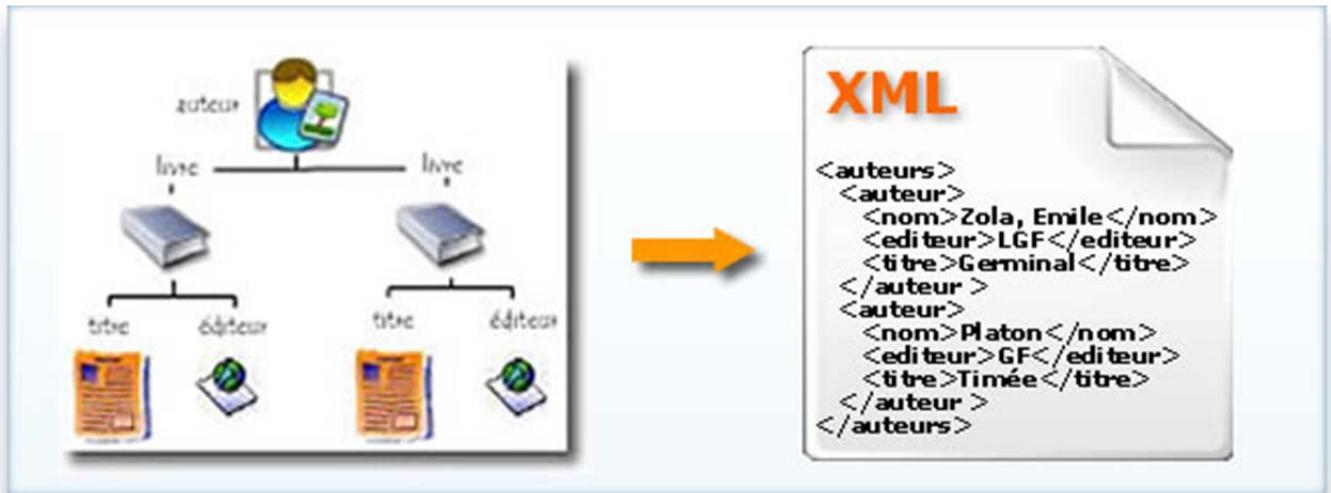
```
String[] lignes = loadStrings("liste.csv");
for (int i=0; i < lignes.length; i++) {
    println("ligne n°"+i+": "+lignes[i]);
    // à compléter avec la fonction split()
    println(tableau) ;
}
```

```
ligne n°0:NOM;PRENOM;CLASSE;SEXE
[0] "NOM"
[1] "PRENOM"
[2] "CLASSE"
[3] "SEXE"
ligne n°1:Lepetit;Alexandre;TS1;G
[0] "Lepetit"
[1] "Alexandre"
[2] "TS1"
[3] "G"
ligne n°2:Gaumont;Clemence;TS1;F
[0] "Gaumont"
[1] "Clemence"
[2] "TS1"
[3] "F"
ligne n°3:Guerin;Sophie;TS2;F
[0] "Guerin"
[1] "Sophie"
[2] "TS2"
[3] "F"
5
```

3. Format XML

XML (entendez *eXtensible Markup Language* et traduisez *Langage à balises étendu*, ou *Langage à balises extensible*) est une méthode permettant de définir un langage (on dit alors qu'il s'agit d'un métalangage), permettant de créer des documents structurés à l'aide de balises (markup). Une balise est une chaîne de caractère du type : <balise>.

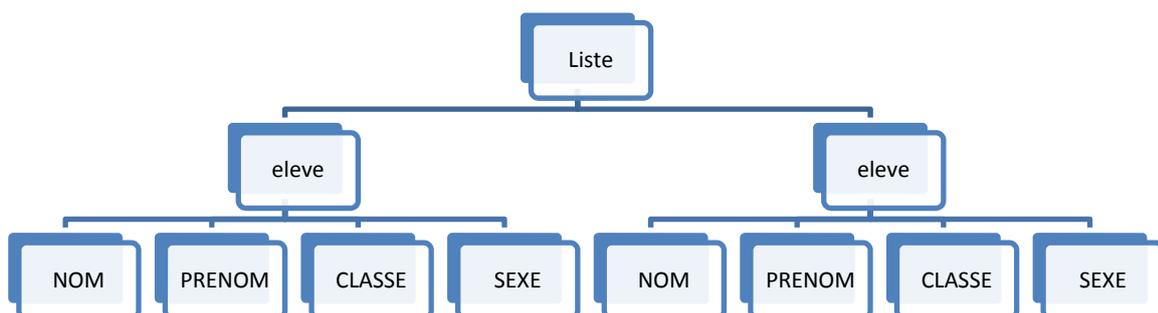
La plupart des composants d'un document XML peuvent être représentés par un arbre possédant diverses branches ou nœuds. Par exemple un fichier d'auteurs, ayant écrits un ou plusieurs livres, diffusés par un éditeur.



4. XML avec Processing

Mettre des données au format XML

L'objectif est de retravailler la liste d'élèves utilisée en début de séance (liste.csv) et de transposer les données au format XML afin d'obtenir l'organisation suivante :



En s'aidant de la syntaxe XML de l'exemple précédent (le fichier d'auteurs) :

10. Ouvrir Notepad++ et sélectionner dans le menu « Langage » l'option XML.
11. Transformer le fichier liste.csv afin de le mettre au format XML et reporter le résultat ci-dessous (attention, il faut ajouter en première ligne le code suivant : <?xml version="1.0" encoding="UTF-8"?>) :

12. Enregistrer le fichier avec l'extension XML

Exemples XML avec Processing

La librairie XML est une librairie native de Processing, c'est-à-dire qu'elle est pré-implantée dans l'environnement de développement.

L'exemple suivant permet d'en comprendre les principes :

```
// The following short XML file called "sites.xml" is parsed
// in the code below. It must be in the project's "data" directory
// <?xml version="1.0"?>
// <websites>
//   <site id="0" url="processing.org">Processing</site>
//   <site id="1" url="mobile.processing.org">Processing Mobile</site>
// </websites>

XMLElement xml;

void setup() {
  size(200, 200);
  xml = new XMLElement(this, "sites.xml");
  int numSites = xml.getChildCount();
  for (int i = 0; i < numSites; i++) {
    XMLElement kid = xml.getChild(i);
    int id = kid.getInt("id");
    String url = kid.getString("url");
    String site = kid.getContent();
    println(id + " : " + url + " : " + site);
  }
}
```

13. Recopier le code dans Processing (version 1.5 uniquement) et enregistrer le fichier sous l'appellation « exemple_xml.pde »
14. Créer le fichier « sites.xml » avec notepad++ en effectuant un copier/coller du code XML en commentaire dans le programme Processing ci-dessus (enlever les //)
15. Ajouter le fichier xml au sketch (menu Sketch/Add file...)
16. Tester le programme et reporter le résultat de la fenêtre d'exécution ci-dessous :

XMLElement est une représentation d'un objet XML. L'objet est capable d'analyser le code XML (attention, dans les versions 2+ de Processing, la librairie XML a été modifiée).

17. Commenter chaque ligne de code du programme Processing.

Astuce : sélectionner avec votre souris un élément de votre programme dans Processing , cliquer droit, puis sélectionner dans le menu contextuel l'option « Find in Reference », cela donne accès directement à la documentation).

Voici une variante du programme précédent :

```
XMLElement xml;

void setup() {
  size(200, 200);
  xml = new XMLElement(this, "sites.xml");
  XMLElement[] kids = xml.getChildren();
  for (int i=0; i < kids.length; i++) {
    String site = kids[i].getContent();
    println(site);
  }
}
```

18. Donner la différence entre l'instruction `getChildren()` et `getChild()`

Le programme suivant montre la possibilité de traiter, directement en ligne, des données XML stockées sur un serveur (ici yahoo):

```
void setup() {
  // télécharge le flux rss portant sur les nouvelles actualités yahoo.com
  String url = "http://rss.news.yahoo.com/rss/topstories";
  XMLElement rss = new XMLElement(this, url);
  println(rss);
  XMLElement[] links = rss.getChildren("channel/item/link");
  for (int i = 0; i < 4; i++) {
    println(links[i].getContent());
  }
}
```

19. Tester le programme

20. Déterminer l'avantage d'utiliser l'instruction `getChildren("channel/item/link")`;

Programmation Processing pour XML

Réaliser sous Processing 1.5, un programme permettant de lire et d'afficher les données contenues dans le fichier `liste.xml`.