

# Enregistrement des données

Christian Stettler

- **Enregistrement des données**
- **Formats de fichier**
- **Manipulation des données**
- **Limitation des fichiers**

- Le type des données étant défini, il faut choisir le format de stockage: « Comment seront stockées les informations? »
- Les données peuvent être enregistrées soit sous forme de **fichiers** soit dans **une base de données**
- Les fichiers et bases de données sont de différents types, selon les **objectifs de stockage** (transfert, archivage, données en production, etc.)

- Le format des fichiers peut être
  - Binaire
  - Plat/ASCII (avec champs de largeur fixe ou délimités)
  - XML

- Le **format binaire** correspond à l'enregistrement dans un fichier des **octets** représentant les informations.
- L'**organisation** des octets doit être connue pour la **lecture** ou **modification** des informations! Il faut donc disposer d'un logiciel capable **d'interpréter** le contenu du fichier.
- Ces fichiers sont souvent appelés « fichiers au format propriétaire ».
- L'organisation des octets pouvant être la même que celle utilisée en RAM, **l'écriture** et la **lecture** sont **optimisées** : il est possible de copier directement un grand nombre d'octets de la RAM vers le fichier ou inversement.
- Les fichiers de format binaire peuvent contenir des caractères non affichables (les 32 premiers ASCII par exemple)

Š', r/-9džBøihwllBcų3ēh<sup>a</sup>EE  
1ÚÉY\* μ?K%9'tìsRÛŔŸ<sup>a</sup>-jìqĀ  
ý]lAaí€,, hIä'y9%2Ñ\*`zèÜ]„ç  
y ÑèÔ!+øĀEöłò¼üfMyÎkĀ£ú♠«K  
%tœ[İßd²2'¶B\Cwp½ŮW7BŮRld.

## Format de fichiers : plat ou ASCII

- Un fichier au format « plat » ou « ASCII » ne contient pas de caractères non affichables.
- L'écriture et la lecture se font généralement par ligne
- Deux possibilités existent pour séparer les données (champs) de chaque ligne:
  - Attribution d'une largeur fixe
  - Utilisation d'un délimiteur



## Format de fichiers : plat ou ASCII

- Exemple avec largeur fixe:

Nom	Prénom	Cours	Module	Début	NbHeures
Stettler	Christian	BDD	621-2	16.09.2019	2
Teodoro	Douglas	Algo	631-1	20.09.2019	3

- Une **largeur max** est définie pour chaque champ  
(attention en cas de dépassement !)

## Format de fichiers : plat ou ASCII

- Exemple avec délimiteur:

Nom;Prénom;Cours;Module;Début;NbHeures
Stettler;Christian;BDD;621-2;16.09.2019;2
Teodoro;Douglas;Algo;631-1;20.09.2019;3

- Le caractère ; est utilisé comme délimiteur.
- Les fichiers de type **CSV** utilisent le ; ou la **tabulation** pour séparer les champs.
- Dans MS Excel (associé aux CSV), le menu outil-converter permet d'indiquer le délimiteur ou de spécifier qu'il s'agit d'un fichier à largeur fixe.



## Format de fichiers : plat ou ASCII

- Le format plat ou ASCII ne permet que difficilement de représenter une **hiérarchie des données**, par exemple la liste des téléphones de chaque client.
- La définition de champs dans les champs, donc de **nouveaux délimiteurs** à l'intérieur d'un champ reste possible mais cela ralentit les opérations de lecture/écriture et rend la **maintenance du modèle difficile** (ajouter des informations pour chaque numéro,...)
- Le format XML permet la **hiérarchisation/l'arborescence** des données

# Format de fichiers : XML

**XML** (*Extensible Markup Language* ([en](#)) <sup>note 1</sup> « langage extensible de balisage ») est un [langage informatique de balisage générique](#). Il sert essentiellement à stocker/transférer des données de type texte [Unicode](#) structurées en champs arborescents. Ce langage est qualifié d'*extensible* car il permet à l'utilisateur de définir les balises des éléments. <sup>note 2</sup> L'utilisateur peut multiplier les [espaces de nommage](#) des balises et emprunter les définitions d'autres utilisateurs. <sup>1, note 3</sup>

Le [World Wide Web Consortium](#) (W3C), promoteur de standards favorisant l'échange d'informations sur [Internet](#), recommande la syntaxe XML pour exprimer des langages de balisages *spécifiques*.

De nombreux langages respectent la syntaxe XML : [SVG](#) pour des données de graphique vectorisé, [XHTML](#) pour des données de page web, [RSS](#) pour des données de [syndication de contenu](#), [XSLT](#) pour des données de [transformation d'un document XML](#), etc. Cette syntaxe est reconnaissable par son usage des *chevrons* (< >) pour les balises d'éléments et de la chaîne <!-- --> pour les commentaires du code.

L'objectif initial est de faciliter l'échange automatisé de contenus entre [systèmes d'informations](#) hétérogènes ([interopérabilité](#)). XML est une simplification du [Standard Generalized Markup Language](#) (SGML) dont il retient les principes essentiels comme :



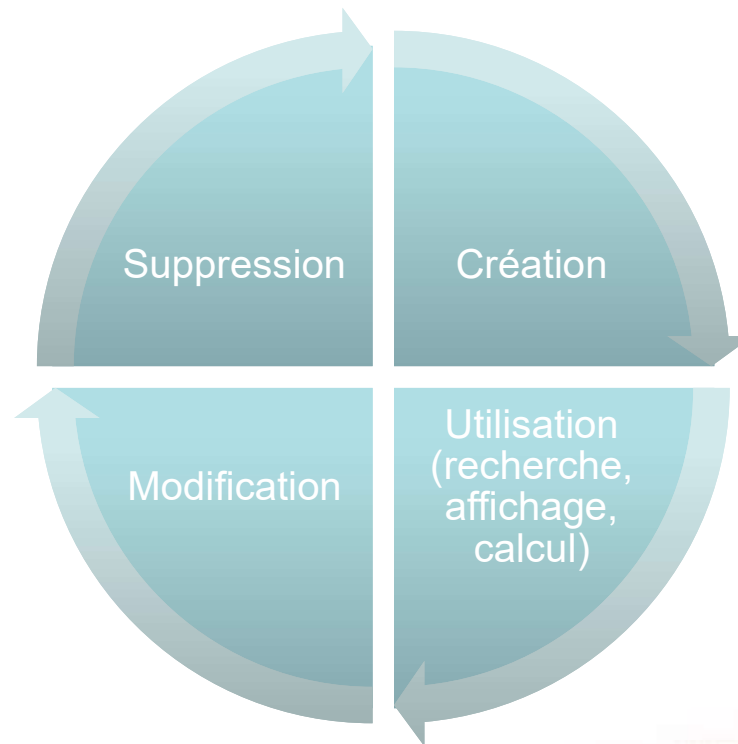
## Format de fichiers : XML

- Exemple de fichier XML
- Une liste de téléphones peut être ajoutée (et pas forcément pour tout le monde)

```
<listeProfs>
- <prof>
  <nom>Stettler</nom>
  <prenom>Christian</prenom>
  <cours>BDD</cours>
  <module>621-2</module>
  <debut>16.09.2019</debut>
  <nbheures>2</nbheures>
</prof>
- <prof>
  <nom>Teodoro</nom>
  <prenom>Douglas</prenom>
  <cours>Algo</cours>
  <module>631-1</module>
  <debut>20.09.2019</debut>
  <nbheures>3</nbheures>
  - <telephones>
    - <telephone>
      <type>Bureau</type>
      <numero>022 1234567</numero>
    </telephone>
    - <telephone>
      <type>Mobile</type>
      <numero>077 7654321</numero>
    </telephone>
  </telephones>
</prof>
</listeProfs>
```

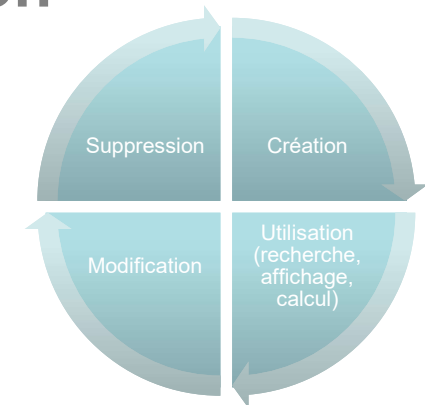


- Les données ont un cycle de vie:





- Le **CRUD** (Create, Retrieve, Update, Delete) représente « la **gestion des informations** » : les opérations de base de toute information



- Comment implémenter le **CRUD** avec le format des fichiers?



- Si le **CRUD** peut être implémenté avec le format des fichiers, qu'en est-il des performances?
- Que se passe-t-il pour un grand nombre de données?
- Comment donner un accès à plusieurs processus/utilisateur en parallèle?  
Comment gérer la sécurité des accès?