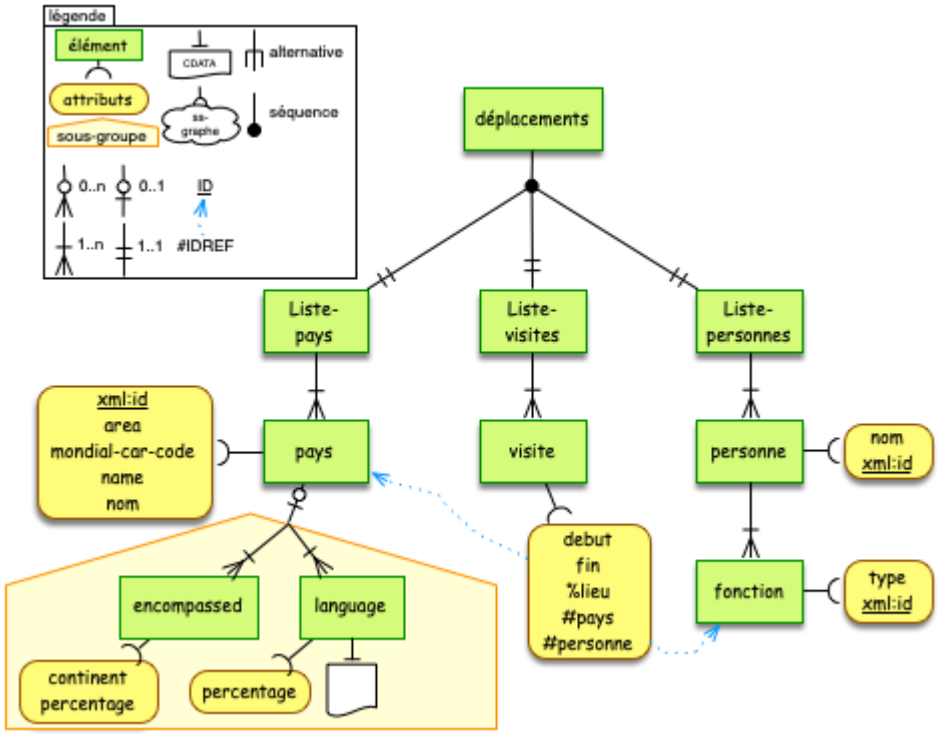
# *Feuille de travaux pratiques n° 1*
# XSLT, SAX & DOM

## Exercice 1 Centres régionaux opérationnels de surveillance et de sauvetage

Le thème du TP est issu de l'Open Data du gouvernement français [1].
On s'intéresse plus particulièrement au jeu de données relatif aux déplacements des présidents de la République et des premiers ministres depuis 1945 [2]. Ces données sont disponibles en format CSV. Je me suis permis de les transformer (en utilisant l'outils csv2xml [3]) et je les ai fusionnés avec des données de la base Mondial [4] en utilisant aussi un "traducteur" anglais-Français des noms de pays (en extrayant un tableau du site `https://english.lingolia.com/fr/vocabulaire/les-pays` puis en le passant en CSV grâce à un tableur et enfin en XML) pour en faire une ressource XML un peu mieux structurée. Il a fallu faire un nettoyage des données assez important aussi bien de manière automatique qu'à la main.
Voici le schéma du document XML ressource :

```
<?xml encoding="UTF-8"?>

<!ELEMENT déplacements (liste-pays,liste-visites,liste-personnes)>

<!ELEMENT liste-pays (pays)+>
<!ELEMENT pays (encompassed+,language+)?>
<!ATTLIST pays
   area CDATA #IMPLIED
   mondial_car_code CDATA #REQUIRED
   name CDATA #REQUIRED
   nom CDATA #REQUIRED
   xml:id ID #REQUIRED>

<!ELEMENT encompassed EMPTY>
<!ATTLIST encompassed
   continent NMTOKEN #REQUIRED
   percentage CDATA #REQUIRED>

<!ELEMENT language (#PCDATA)>
<!ATTLIST language
   percentage CDATA #IMPLIED>

<!ELEMENT liste-visites (visite)+>
<!ELEMENT visite (#PCDATA)>
<!ATTLIST visite
   debut NMTOKEN #REQUIRED
   fin NMTOKEN #REQUIRED
   lieu CDATA #IMPLIED
   pays IDREF #REQUIRED
   personne IDREF #REQUIRED>

<!ELEMENT liste-personnes (personne)+>
<!ELEMENT personne (fonction)+>
<!ATTLIST personne
   nom CDATA #REQUIRED
   xml:id ID #REQUIRED>

<!ELEMENT fonction EMPTY>
<!ATTLIST fonction
   type CDATA #REQUIRED
   xml:id ID #REQUIRED>
```

1. `https://data.gouv.fr`
2. `https://data.culture.gouv.fr/explore/dataset/deplacements-presidents-republique-et-premiers-ministres-depuis-1945/`
3. `https://github.com/edesmontils/csv2xml`
4. `http://dbis.informatik.uni-goettingen.de/Mondial/`




On cherche à écrire un programme permettant d'obtenir pour chaque président de la répubique, le temps passé en visite dans chacun des pays africains. On notera si le pays est francophone, c'est-à-dire si la langue française est officielle ou qu'elle à un pourcentage supérieur à 30% de la population).
Le document à générer doit respecter la DTD ≪tp.dtd≫ suivante :

```
<?xml encoding="UTF-8"?>

<!ELEMENT liste-présidents (président)+>

<!ELEMENT président (pays)+>
<!ATTLIST président
   nom CDATA #REQUIRED>

<!ELEMENT pays EMPTY>
<!ATTLIST pays
   durée NMTOKEN #REQUIRED
   franchophone (En-partie|Officiel) #IMPLIED
   nom CDATA #REQUIRED>
```

Sur le document "tp.xml", cela donnera le document "res.xml" présent sur Madoc.

### Question 1

Donner le programme permettant d'obtenir le résultat précédent en utilisant XSLT, sans utiliser ≪xsl :for-each≫ ou des règles nommées.

### Question 2

Donner les programmes (PHP, Python ou Java) permettant d'obtenir le résultat précédent en utilisant les outils suivants : SAX[5], DOM[6] (sans utiliser XPath ou "getElementsByTagName()"), DOM en utilisant XPath (sans utiliser "//"). Pour chaque version, faites en sorte que le traitement soit optimal en temps de traitement.

### Question 3

Discuter des différentes versions proposées (XSLT, DOM, DOM/XPath, SAX) en termes de performance, d'adéquation au traitement, de clarté du code, de maintenance, etc.

5. Vous pouvez utiliser la bibliothèque Sax4PHP : `https://github.com/edesmontils/Sax4PHP`
6. Vous pouver utiliser la bibliothèque Python "minidom_ext" `https://github.com/edesmontils/Minidom_ext`

**Question 4 Facultative**

Reprendre cette recherche en utilisant :

1. Java et JDOM. Pour cela, vous pourrez vous appuyer sur le site Web de JDOM. Deux versions sont attendues : avec et sans filtres. Là encore, le résultat sera construit en JDOM.
2. les objets PHP : XMLReader pour le traitement et XMLWriter pour la génération. Comparez avec les solutions précédentes.
3. l'API SimpleXML de PHP et la bibliothèque lxml en Python.

## Travail à rendre :

Vous devrez rendre votre travail à l'enseignant sous forme électronique d'une part le rapport (un petit document PDF de 10 pages présentant votre travail, analyses, structures de données, algorithmes, conclusions, etc., les subtilités de vos traitements et vos difficultés) et d'autre part une archive de nom "noms-binome.tar.gz" où vous donnerez :

— votre code (bien présenté et commenté) ;
— un fichier "readme" et un fichier shell permettant à l'enseignant d'exécuter votre code sur sa machine.

Deux zones de dépôt sur Madoc seront disponible à cet effet. Une séance de tests pourra être organisée.
Remarque : une attention toute particulière doit être portée dans votre rapport sur la description de la structure de données utilisée (en SAX et en DOM) et sur la manière dont elle est exploitée.

Quelques références :

— Standard XPath : http ://xmlfr.org/w3c/TR/xpath
— Standard DOM : http ://www.w3.org/DOM/
— Documentation XMLLINT : http ://xmlsoft.org/xmllint.html ;
— Documentation XSLTProc : http ://xmlsoft.org/XSLT/xsltproc.html
— Documentation PHP DOM : http ://www.php.net/manual/fr/ref.dom.php
— Documentation PHP ≪SAX≫ : http ://www.php.net/manual/fr/ref.xml.php
— Documentation PHP SimpleXML : http ://php.net/manual/fr/book.simplexml.php
— Documentation XMLReader/XMLWriter : http ://fr2.php.net/manual/fr/book.xmlreader.php & http ://fr2.php.net/manual/fr/boo
— Site JDOM : http ://www.jdom.org/downloads/docs.html

RAPPEL :

— *Il est aussi possible d'exécuter du PHP dans une fenêtre ≪terminal≫ en faisant : ≪php mon_exo.php≫.*
— *Il est posssible d'exécuter le code XSLT en ligne de commande en utilisant Saxon*[7]*, par exemple : ≪java -jar saxon9he.jar -xsl :mon_exo.xsl -s :ma_source.xml≫. L'ajout du paramètre "-TP :profile.html" permettra d'obtenir un profilage de l'exécution en HTML. XSLT peut aussi être exécuté par PHP ou en Python.*
— *Éventuellement, pour évaluer le temps de traitement, il faut faire précéder la ligne par ≪time≫, par exemple : ≪time php mon_exo.php≫ ou ≪time java MaClasse≫. Le temps de traitement effectif de votre programme est la ligne ≪user≫.*
— *Pour faire des tests de performance (temps d'exécution), il est possible d'utiliser l'outil "hyperfine"* [8]*.*
— *Pour comparer deux documents XML, vous pouvez utiliser XMLDiff*[9]*.*

7. Disponible sur Madoc. Documentation : `http://www.saxonica.com/documentation/`
8. `https://github.com/sharkdp/hyperfine`
9. `https://github.com/Shoobx/xmldiff`