

Série TP n°4**Chapitre 2: XML**

Module

MTI Méthd. et tech d'implement.

Filière

Master GSI

1^{ère} Année**XML DOM**

Soit le script python suivant

```
#!/usr/bin/env python
#- * -coding: utf - 8 - * -##testdom2.py##
DATA_FILE = 'TP/annuaire-v0.2.xml'
import xml.dom.minidom as minidom
import sys
def main():
    try:
        xmldoc = minidom.parse(DATA_FILE)
    except:
        print "Can't Open the file",
        sys.exit()
    print xmldoc.toxml()
    return 0
if __name__ == '__main__':
    main()
```

Que fait ce script ?

Créer la fonction suivante, exécuter là dans la fonction main :

```
def treat_doc(xmldoc):
    # get the annuaire list
    annuaire = xmldoc.getElementsByTagName('annuaire')[0]
    print annuaire
    cpt = 0
    # display personne by personne
    for personne in annuaire.childNodes:
        print "-"*40
        print "Personne n°", cpt
        print personne.toxml()
        cpt += 1
```

Expliquer les résultats

On veut afficher tous les numéros de téléphones dans notre fichier, on utilise la fonction

```
def display_tel(xml doc):  
    # display only telephones  
    # get the tel list  
    telephones = xml doc.getElementsByTagName('telephone')  
    print telephones  
    cpt = 0  
    # display tel by tel  
    for tel in telephones:  
        print "-"*40  
        print "Tel n°", cpt  
        print tel.toxml()  
        print "N°:", tel.firstChild.data  
        print "Type:", tel.getAttribute("type")  
        cpt += 1
```

Afficher les numéros de téléphone par personne

```
def display_tel_personne(xml doc):  
    #get the personnes list  
    personnes = xml doc.getElementsByTagName('personne')  
    print personnes  
    cpt = 0  
    # display telephone by personne  
    for personne in personnes:  
        print "-"*40  
        print "Personne n°", cpt  
        nom = personne.getElementsByTagName('nom')[0]  
        prenom = personne.getElementsByTagName('prenom')[0]  
        tels = personne.getElementsByTagName('telephone')  
        print "*"*20  
        print "Nom:\t", nom.firstChild.data  
        print "Prénom:\t", prenom.firstChild.data  
        for tel in tels:  
            print "-"*20  
            print "N°:", tel.firstChild.data  
            print "Type:", tel.getAttribute("type")  
        cpt += 1
```

Ajouter un nouvel attribut à l'élément Personne

```
def add_id_personne(xml doc):  
    #get the personnes list  
    personnes = xml doc.getElementsByTagName('personne')  
    print personnes  
    cpt = 0  
    # display personne by personne  
    for personne in personnes:  
        print "-"*40  
        print "Personne n°", cpt, personne.nodeValue, personne.nodeType,
```

```
personne.setAttribute('id', str(cpt))  
cpt += 1  
print personne.toxml()
```

Partie 2 : Manipulation avec javaScript

Donner l'arbre DOM de code suivant

```
<html>
  <body>
    <div>
      <b>Nom:</b> <span id="FirstName"></span><br/>
      <b>Prénom:</b> <span id="LastName"></span><br/>
      <b>Tel:</b> <span id="phone"></span><br>
      <b>ville:</b> <span id="town"></span>
    </div>
  </body>
</html>
```

Ajouter le code javascript suivant, exécuter le

```
<script>
  //modify the HTML
  document.getElementById("FirstName").innerHTML= "Taha";
  document.getElementById("LastName").innerHTML= "Zerrouki";
</script>
```

Modifier le code javascript pour afficher le tel et la ville

Modifier le code pour créer des Element HTML

```
//Create a <p> element

var para = document.createElement("H1");
// Create a text node
var t = document.createTextNode("This is a paragraph");
// Append the text to <p>
para.appendChild(t);
para.setAttribute("align", "center");
document.body.appendChild(para);
```

Le code qui permet de charger un fichier XML

```
<script>
  if (window.XMLHttpRequest) //if browser supports XMLHttpRequest
  { // Create an instance of XMLHttpRequest object. code for IE7+, Firefox,
Chrome, Opera, Safari
    xmlhttp = new XMLHttpRequest();
  }
  else
  { // code for IE6, IE5
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
  }
  // sets and sends the request for calling "node.xml"
  xmlhttp.open("GET", "annuaire-v0.2.xml", async = false);
  xmlhttp.send();
  // sets and returns the content as XML DOM
  xmlDoc=xmlhttp.responseXML;
  // parsing dom
</script>
```

Récupérer les données

```
//parsing the DOM object
document.getElementById("FirstName").innerHTML =
    xmlDoc.getElementsByTagName("nom")[0].childNodes[0].nodeValue;
</script>
```

Afficher le prénom, tel, ville

Affichage des infos dans une table

```
<table id="table_personnes" border=1>
<tr> <th>Nom</th><th>Prénom</th><th>Phone</th><th>Ville</th></tr>
</table>
```

Affichage des infos dans une table

```
var identities = xmlDoc.getElementsByTagName("identite")
var table = document.getElementById("table_personnes")
for(i in identities)
{
    var tr = document.createElement("tr");           // create a row
    var td = document.createElement("td");           // create a row
    td.innerHTML = identities[i].getElementsByTagName("nom")[0].childNodes[0].nodeValue;
    tr.appendChild(td);           // add td to tr
    // redo for another data
    tr.appendChild(td);
    table.appendChild(tr);
}
```

Afficher le prénom, téléphone et ville

Travail à Domicile

- Utiliser le fichier des étudiants résultant de TP n°1.
- Créer un programme en utilisant XML DOM API (python ou Java) pour :
 - Classer les étudiants par la moyenne générale.
 - Calculer le résultat de l'année à partir de la moyenne générale.
 - Mettre en évidence le résultat de l'année.
 - Extraire les statistiques suivantes : La meilleure moyenne, la mauvaise ; Le nombre des étudiants. ; La moyenne de la classe ; Le nombre par résultat.

2ème partie :

On veut apprendre comme utiliser des API distantes offerts par des sites web en XML ou JSON comme :

- Météo : <https://openweathermap.org/>
- Carte géographique : <https://www.openstreetmap.org>
- FootBall: <https://rapidapi.com/api-sports/api/api-football/>
- Plus d'API sur <https://rapidapi.com>

On veut générer des fiches techniques à partir des résultats des API, choisir un thème :

- Une brochure touristique (description du lieu, Carte, Météo, Heures de prières, etc.)
- Equipe de football (description du lieu, images, résultats....)
- Comparateur de prix entre les sites
- Index des films.
- Etc...