

Institut Supérieur des Etudes Technologiques de Radès		
	Département Technologies de l'Informatique Session: janvier 2013 Enseignants : N.JAAFAR+M.MTIR Durée : 1h30mn Groupe : L2DSI1 + L2DSI2+L2DSI3	

Examen Final **Conception Web Avancée et Programmation** **Asynchrone**

- ❖ Nbre de pages=3 + Annexe composée de 2 pages
- ❖ Documents non autorisés

Exercice 1 (10 points):

1. Le XML est un métalangage de description de données. Expliquer cette affirmation en se basant sur des exemples.
2. Définir ce qu'on appelle une «entité XML», en précisant dans quels cas on en aura recours.
3. Dans une règle DTD de type ATTLIST, définir les rôles joués par les clauses : #REQUIRED, #FIXED et #IMPLIED.
4. Etant donné le fichier DTD, calcul.dtd suivant :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!ELEMENT formules (formule)+ >
```

```
<!ELEMENT formule (valeur|somme|difference|produit|fraction) >
```

```
<!ELEMENT valeur (#PCDATA)>
```

```
<!ELEMENT somme (op1, op2)>
```

```
<!ELEMENT difference (op1, op2)>
```

```
<!ELEMENT produit (op1, op2)>
```

```
<!ELEMENT fraction (op1, op2)>
```

```
<!ELEMENT op1 (valeur|formule)>
```

```
<!ELEMENT op2 (valeur|formule)>
```

Donner le code XML d'un fichier calcul.xml destiné à représenter la formule de calcul suivante :

$$7 + 5 / 2 * 3$$

Le code XML de calcul.xml doit être well formed et valide par rapport à calcul.dtd

On suppose que les deux fichiers calcul.xml et calcul.dtd se trouvent dans le même répertoire.

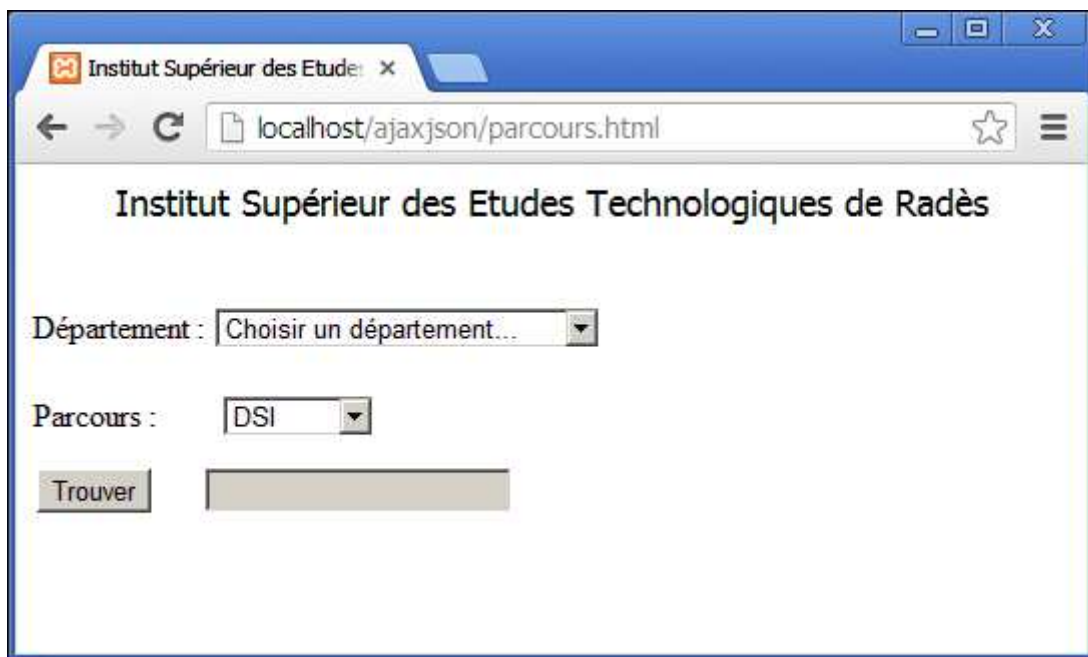
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE formules SYSTEM "calcul.dtd">
  <formules>
    <formule>
      <fraction>
        <op1>
          <formule>
            <somme>
              <op1><valeur>7</valeur></op1>
              <op2><valeur>5</valeur></op2>
            </somme>
          </formule>
        </op1>
        <op2>
          <formule>
            <produit>
              <op1><valeur>2</valeur> </op1>
              <op2><valeur>3</valeur> </op2>
            </produit>
          </formule>
        </op2>
      </fraction>
    </formule>
  </formules>
```

Exercice 2 (10 points):

L'objectif de l'exercice est de développer une interface web permettant de retrouver le nombre d'étudiants par parcours dans un département au niveau de l'ISET de Radès à travers une communication asynchrone Ajax avec un serveur Apache, PHP.

Les listes « Département » et « Parcours » seront alimentées dynamiquement à travers un objet JSON « isetrades », alors que le nombre d'étudiants sera retourné par un fichier PHP « parcours.php » via AJAX.

L'interface utilisée est décrite par l'imprime écran ci-dessous :



Le code source HTML (parcours.html) de cette interface est donné en annexe 1.

Travail à faire :

1. Donner le code source JavaScript de la méthode *dessinerListe1* de l'objet JSON « isetrades » qui permet de dessiner l'objet liste déroulante des départements (*liste1*).

```

enquete.dessinerListe1=function() {
    i=1;
    for(indice in this)
    {
        // if( typeof(this[indice])=="object")
        // document.write("<option value=\""+ indice + "\">" + indice + "</option>");

        if( typeof(this[indice])!="function")
            document.f1.liste1.options[i]=new Option(indice,indice);
        i=i+1;
    }
}

```

- Donner le code source JavaScript de la méthode *dessinerListe2* de l'objet « *isetrades* » qui permet de mettre initialement la liste exhaustive des parcours dans l'objet « *liste2* ».

```

enquete.dessinerListe2=function() {
    i=0;
    for(indice1 in this)
    {
        if( typeof(this[indice1])=="object")    {
            for(indice2 in this[indice1])    {
                //document.write("<option value="+ indice2 + ">" + indice2 + "</option>");
                document.f1.liste2.options[i]=new Option(this[indice1][indice2],this[indice1][indice2]);
                i=i+1;
            }
        }
    }
}

```

- Implémenter la méthode *filtrer* de l'objet « *isetrades* » qui permet au choix d'un département dans la 1^{ère} liste de ne garder dans la 2^{ème} liste que les parcours de ce département.

La méthode *filtrer* doit remettre la totalité des parcours dans la 2^{ème} liste si l'élément choisi dans la 1^{ère} liste à la valeur « *lesdepartements* »

```

enquete.filtrer=function() {

x=document.f1.liste1.value;
n=document.f1.liste2.options.length;
for(i=n - 1; i >= 0; i--)
    document.f1.liste2.options[i]=null;

if(x=="lesdepartements")
    enquete.dessinerListe2();
else
    {
        tab=enquete[x];
        i=0;
        for(indice in tab)      {
            document.f1.liste2.options[i]=new Option(tab[indice],tab[indice]);
            i++;
        }
    }

}

```

4. Donner le code de la méthode *trouver* de l'objet « *isetrades* » qui permet à travers un objet XMLHttpRequest de faire appel d'une façon asynchrone au fichier *parcours.php* (annexe2) en lui envoyant par la méthode GET les deux paramètres *dep* et *parcours* qui correspondent respectivement au département et au parcours choisis.

Un message d'erreur sera affiché si au moment de l'appel de la méthode *trouver* la valeur de la 1^{ère} liste est encore « *lesdepartements* ».

L'image animée (*ajax-loader.gif*) sera visible pour signaler qu'un traitement sur le serveur est encore d'exécution, une fois la transaction est achevée l'image sera masquée.

```
enquete.trouver=function()
{
    var xhr = getXhr();
    x=document.f1.liste1.value;
    y=document.f1.liste2.value;
    url ="parcours.php?dep=" + x +"&par=" + y;
    xhr.open("GET",url,true);
    document.images["chargeur"].style.visibility="visible";
    xhr.send(null);
    xhr.onreadystatechange = function()
    {
        // On ne fait quelque chose que si on a tout reçu et que le serveur est ok
        if(xhr.readyState == 4 && xhr.status == 200)
        {
            document.f1.resultat.value= xhr.responseText;
            document.images["chargeur"].style.visibility="hidden";
        }
    }
}
```

Bonne Chance