

Traitement des données (6H00)

Analyse d'une chaîne de caractères au format XML et création d'une liste.

```
<!DOCTYPE html>
<html>
<head>
  <title>XML</title>
  <script type="text/javascript">
    var data = '<?xml version="1.0" encoding="UTF-8" ?> <GENS> <PERSONNE> <NOM>
Flora Bonhomme </NOM> <ADRESSE> <RUE> 5 rue Galliéni </RUE><CODE> 75012 </CODE>
<VILLE> Paris </VILLE> </ADRESSE> <TEL> 01 23 45 67 89 </TEL> </PERSONNE>
<PERSONNE> <NOM> Anne Gentile </NOM> <ADRESSE> <RUE> 2 rue des Dames </RUE> <CODE>
75007 </CODE> <VILLE> Paris </VILLE> </ADRESSE> <TEL> 01 45 23 67 89 </TEL>
</PERSONNE></GENS>';
    function affiche() {
      var parser = new DOMParser();
      var docXML = parser.parseFromString(data, "application/xml");
      var reponses = docXML.getElementsByTagName("NOM");
      var texte = "";
      var liste = document.getElementById('liste');
      for (i = 0; i < reponses.length; i++) {
        var li = document.createElement('li');
        li.appendChild(document.createTextNode(reponses[i].firstChild.nodeValue));
        liste.appendChild(li);
      }
    }
  </script>
</head>
<body onload="affiche();">
  <div><ul id="liste"></ul></div>
  <table id="tb" style="border-style:solid;border-width:1px"></table>
</body>
</html>
```

Exercice : afficher ces mêmes données sous forme d'un tableau HTML.

Même chose avec JSON

```
<!DOCTYPE html>
<html>
<head>
  <title>JSON -> Objets</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <script id="music" type="application/json">[{ "Nom_Musicien": "Bacelli",
"Prénom_Musicien": "Monica" },
      { "Nom_Musicien": "Bach", "Prénom_Musicien": "Carl-Philip-Emmanuel" },
      { "Nom_Musicien": "Bayrakdarian", "Prénom_Musicien":
"Isabel" }]</script>
  <script type="text/javascript">
    var musiciens;
    function Affiche() {
      ChargeTableau();
      CréeeListe();
    }
    function ChargeTableau() {
      musiciens = JSON.parse(document.getElementById('music').text);
    }
    function CréeeListe() {
      var list = document.getElementById("liste");
      for (var i = 0; i < musiciens.length; ++i) {
        var li = document.createElement("p");
        var noeud = document.createTextNode(musiciens[i].Nom_Musicien + ", " +
musiciens[i].Prénom_Musicien);
        li.appendChild(noeud);
        list.appendChild(li);
      }
    }
  </script>
</head>
<body>
  <div><ul id="liste"></ul></div>
  <table id="tb" style="border-style:solid;border-width:1px"></table>
</body>
</html>
```

```

    }
    </script>
  </head>
  <body onload="Affiche()">
    <div id="liste"></div>
  </body>
</html>

```

Exercice : Afficher les données dans une liste numérotée

Avec ces mêmes données, création d'un *objet*, stockage des données dans un tableau d'objets.

```

var musiciens = [];
function Musicien(n, p) {
  this.nom = n;
  this.prenom = p;
}
function ChargeTableau() {
  var Musiciens = JSON.parse(document.getElementById('music').text);
  Musiciens.forEach(function (mus) {
    musiciens.push(new Musicien(mus.Nom_Musicien, mus.Prénom_Musicien));
  });
}
function Affiche() {
  ChargeTableau();
  CréeeListe();
}
function CréeeListe() {
  var list = document.getElementById("liste");
  musiciens.forEach(function (mus) {
    var li = document.createElement("li");
    var noeud = document.createTextNode(mus.nom + ", " + mus.prenom);
    li.appendChild(noeud);
    list.appendChild(li);
  });
}

```

Exercice : afficher les données sous forme d'un tableau HTML avec colonne pour les noms et une pour les prénoms.

Afficher la liste des musiciens une

Lecture et création dynamique de données

Prototypes, surcharge de méthodes : nous allons dans ce qui suit utiliser un vocabulaire de type *programmation objet*, même s'il n'est pas parfaitement adapté.

Ajout d'une classe *Personne* avec une méthode d'affichage de l'identité, dont dérivent les classes *Musicien* et *Abonné*.

La classe *Musicien* rajoute un attribut *instrument* à la classe *personne* surcharge l'identité. On notera la différence dans la création d'*afficheIdentité* sur *Personne* et la surcharge dans *Musicien*.

On reprend l'exercice précédent, on remplace la réponse à *onload* par chargement des données sans affichage, puis on rajoute un bouton affichant la liste des musiciens (et stockant les abonnés dans une liste).

Une nouvelle classe *Abonné* possède une *Adresse* avec les attributs *Rue*, *Ville*, et surcharge *identité* en affichant l'adresse de l'*Abonné*.

Puis on rajoute un formulaire destiné à ajouter des *Abonné*. Le bouton *un abonné* permet d'insérer un *Abonné* dans la liste et de l'afficher au-dessous.

Enfin un nouveau bouton permet d'afficher toutes les personnes (« polymorphisme ») dans une liste.

Liste des musiciens

- Bacelli Monica piano
- Bach Johann Christoph piano
- Bayrakdarian Isabel piano

Nom Dupont
Prénom Pierre
Rue rue Naudet
Ville Talence et
Ajouter un abonné

Liste des abonnés un simple

- Dupont Pierre rue Naudet Talence

Afficher la liste de toutes les personnes

Liste des personnes

Ajouter

- Bacelli Monica piano
- Bach Johann Christoph piano
- Bayrakdarian Isabel piano
- Dupont Pierre rue Naudet Talence

Réalisation *guidée* de l'exercice, en introduisant au fur et à mesure les notions qui viennent d'être présentées (et en ajoutant quelques curiosités syntaxiques).

```
<script type="text/javascript">
//Classe Personne
function Personne(n, p) {
    this.nom = n;
    this.prenom = p;
}
Personne.afficheIdentité = function () {
    return this.nom + " " + this.prenom;
}
//Classe Musicien
function Musicien(n, p, i) {
    Musicien.prototype = Personne;
    Personne.apply(this, arguments);
    this.instrument = "piano";
    this.afficheIdentité = function () {
        return Personne.afficheIdentité.call(this) + " " + this.instrument;
    }
}
//Classe Adresse
function Adresse(v, r) {
    this.ville = v;
    this.rue = r;
    this.adresse = function () {
        return this.rue + " " + this.ville;
    }
}
//Classe Abonné
function Abonne(n, p, r, v, c) {
    Abonne.prototype = Personne;
    Personne.apply(this, arguments);
    this.code = c;
    this.adresse = new Adresse(v, r);
    this.afficheIdentité = function () {
        return this.code + " " + Personne.afficheIdentité.call(this) + " " +
this.adresse.adresse();
    }
}
// Tableaux de données
var artistes = [];
var personnes = [];
var abonnees = [];
// Affichage des éléments du tableau
Array.prototype.Affiche = function (liste) {
    var list = document.getElementById(liste);
    list.style.visibility = "visible";
    while (list.lastChild)
        list.removeChild(list.lastChild);
    this.forEach(function (elem) {
        var li = document.createElement("li");
        li.appendChild(document.createTextNode(elem.afficheIdentité()));
        list.appendChild(li);
    });
};
// Chargement des données
function ChargeMusiciens() {
    var musiciens = JSON.parse(document.getElementById('music').text);
    musiciens.forEach(function (e) {
        var mus = new Musicien(e.Nom, e.Prenom);
        artistes.push(mus);
        personnes.push(mus);
    });
}
function InsereAbonné() {
```

```

var nom = document.getElementById("nom").value;
var prenom = document.getElementById("prenom").value;
var rue = document.getElementById("rue").value;
var ville = document.getElementById("ville").value;
var code = parseInt(document.getElementById("code").value);
document.getElementById("code").value = code + 1;
var ab = new Abonne(nom, prenom, rue, ville, code);
abonnes.push(ab);
personnes.push(ab);
abonnes.Affiche('Abonnes');
}
</script>

```

Création d'un "panier"

Exercice : à partir d'un tableau de données (albums), permettre la mise à jour dynamique du panier par glisser-déplacer (ajout et suppressions), calcul de coûts et totaux.

Exercice : On pourra utiliser les objets sessionStorage ou localStorage pour assurer un suivi du panier entre plusieurs pages ou même entre deux connexions (tout ça sans l'utilisation de cookies ou de mécanisme de suivi de session et donc sans accès à un quelconque serveur).

A achever pour la rentrée.

Les données utilisées dans l'exemple ci-contre sont extraites du tableau suivant
JSON, qui pourra être utilisé pour réaliser ces exercices :

Achat d'albums

Déplacez la pochette dans votre caddie pour ajouter l'album à votre liste d'achat. Pour supprimer l'album de votre liste, il suffit de tirer sa pochette hors du caddie ou de mettre à 0 la quantité dans le tableau de droite.



Titre de l'album	Quantité	Prix
Mozart : Le Nozze di Figaro	1 ▼	10 €
Mozart : Intégrale des trios avec piano	1 ▼	10 €
Coltrane : A love supreme	2 ▼	20 €
Prix total		40 €

- [Mozart : Le Nozze di Figaro](#). 2011
- [Mozart : Intégrale des trios avec piano](#). 2012
- [Coltrane : A love supreme](#). 1978
- [Atom Heart Mother](#). 1973
- [Wallflower](#). 2003
- [Chaleur Humaine](#). 2014

au format

```

var Albums = [{ "Code": "10", "Titre": "Mozart : Le Nozze di Figaro", "An": "2011",
"URL": "http://ecx.images-amazon.com/images/I/51LIHDbHUQL.jpg" },
  { "Code": "15", "Titre": "Mozart : Intégrale des trios avec piano", "An":
"2012", "URL": "http://ecx.images-amazon.com/images/I/41ZTHHNBRKL.jpg" },
  { "Code": "16", "Titre": "Coltrane : A love supreme", "An": "1978", "URL":
"http://ecx.images-amazon.com/images/I/71vq8ZUAmxL_SL1281.jpg" },
  { "Code": "11", "Titre": "Atom Heart Mother", "An": "1973", "URL":
"http://ecx.images-amazon.com/images/I/81UBy5DIxNL_SL1300.jpg" },
  { "Code": "21", "Titre": "Wallflower", "An": "2003", "URL": "http://ecx.images-
amazon.com/images/I/510rzLFTI2L.jpg" },
  { "Code": "29", "Titre": "Chaleur Humaine", "An": "2014", "URL":
"http://ecx.images-amazon.com/images/I/51wcEQzibdL.jpg" },
  { "Code": "53", "Titre": "Rameau : Le grand théâtre de l'amour", "An": "2013",
"URL": "http://ecx.images-amazon.com/images/I/71SXjNkclGL_SL1500.jpg" },
  { "Code": "57", "Titre": "Enfant de la Ville", "An": "2009", "URL":
"http://ecx.images-amazon.com/images/I/51J7UYqvXUL.jpg" },
  { "Code": "82", "Titre": "Unapologetic", "An": "2012", "URL":
"http://ecx.images-amazon.com/images/I/71202hkagIL_SL1000.jpg" }
];

```