

# Requêtes et réponses XML

Nom et prénom : \_\_\_\_\_

## Introduction et objectifs du TP

Lors de cette séance de TP vous allez tout d'abord travailler sur les possibilités offertes par XML pour dialoguer de manière structurée avec un serveur.

## 7.1 Intérêt d'échanges en XML

### 7.1.1 Séance de présentation

XML est un langage informatique de balisage générique. Le W3C, promoteur de standards favorisant l'échange d'informations sur l'Internet, recommande la syntaxe XML pour exprimer des langages de balisages spécifiques. Son objectif initial est de faciliter l'échange automatisé de contenus entre systèmes d'informations hétérogènes (interopérabilité). Cette syntaxe est reconnaissable par son usage des chevrons (< >) et s'applique à de plus en plus de contenus<sup>1</sup>.

### 7.1.2 Echange en mode texte

La personne ayant commandé le site web sur la vente de planches de surf souhaite porter quelques modifications. Au lieu de voir afficher simplement le montant total des ventes, il souhaite avoir le total par type de planche vendues. Ce vendeur propose 3 planches différentes vendues à trois prix différents (Hawaii/520€, Biarritz/450€, Lacanau/500€)

#### Question 1

Reprenez les fichiers du site web du TP1, puis modifiez le code HTML initial afin de pouvoir afficher les informations demandées selon le modèle donnez ci-dessous.

Planches Hawaii vendues	1672
Prix de vente	672 €
Prix de revient	451 €
Planches Lacanau vendues	1672
Prix de vente	547 €
Prix de revient	352 €
Planches Biarritz vendues	1672
Prix de vente	412 €
Prix de revient	240 €

**Bénéfice net : 318936.42 €**

Actualiser

1. Définition wikipédia

## Question 2

Créez un fichier nommé "GetVenteSansXML.php" dans lequel vous écrirez le code suivant

```
1 <?php
2 // Commence avec un nombre arbitraire de ventes
3 $Hawaii= 1672;
4 $Biarritz= 312;
5 $Lacanau= 82;
6 // Reflete les nouvelles ventes
7 srand((double)microtime() * 1000000);
8 $Hawaii= $Hawaii+ rand(0,10);
9 $Biarritz= $Biarritz+ rand(0,5);
10 $Lacanau= $Lacanau+ rand(0,3);
11 echo $Hawaii . "-" . $Biarritz . "-" . $Lacanau . "\n";
12 ?>
```

Commentez la dernière ligne de ce script php. Quelle est la principale différence avec la version du TP1 ?

## Question 3

Il faut maintenant interpréter la réponse du serveur. Complétez le nouveau code de la fonction callback `actualiserPage()` afin que les informations reçues par le serveur s'affichent dans une boîte alert de la manière suivante : "Vente Hawaii=x, Vente Lacanau=y, Vente Biarritz=z" ; Les deux méthodes dont vous avez besoin pour "récupérer" les données sont :

1. `Integer chaine.indexOf(String souschaine, Integer debut)` Retourne la position d'une sous-chaîne souschaine dans une chaîne, à partir de la position debut. Si la sous-chaîne n'est pas trouvée, `indexOf()` retourne -1
2. `String chaine.substring(Integer debut [, Integer fin])` Extrait une sous-chaîne d'une chaîne en partant de l'indice debut et jusqu'à l'indice fin. Si fin n'est pas précisé, la chaîne est extraite depuis le début jusque sa fin.

```
1 if (requete.readyState == 4) {
2     if (requete.status == 200) {
```

```

3   var reponse = _____ ;
4   var posH=reponse.indexOf _____ ;
5   var Hawaii=reponse. _____ ;
6   alert( _____ );
7   var posL=reponse. _____ ;
8   var Lacanau=reponse. _____ ;
9   alert( _____ );
10  var posB=reponse. _____ ;
11  var Biarritz= _____ ;
12  alert( _____ );
13  }
14 }

```

#### Question 4

Revenez maintenant sur la chaîne renvoyée par le script php. L'ordre à l'intérieur de la chaîne a-t'il une importance ? Y a-t'il une erreur dans actualiserPage() ? Si oui alors corrigez l'erreur.

#### Question 5

De manière générale quel reproche pouvez vous faire à ce mode d'échange de données entre un serveur et une application ?

### 7.1.3 Utilisation du XML

#### Question 6

Nous allons maintenant modifier le script php afin que la réponse du serveur soit en XML. Complétez le code suivant afin que la réponse du serveur soit en XML.

```

1  <?php
2  // Commence avec un nombre arbitraire de ventes
3  //..... idem precedemment....
4  //La ligne suivante dit au navigateur que le script
5  //retourne du XML, pas du HTML ou du texte
6  header("Content-Type: text/xml");
7  echo '<?xml version="1.0" encoding="iso-8859-1"?>';
8  ?>
9  <totaux>
10 <Hawaii><?php echo $Hawaii;?></Hawaii>
11 _____
12 _____
13 </totaux>

```

#### Question 7

La réponse du serveur étant au "format" XML, vous allez pouvoir tirer partie de nombreux avantages de ce format de réponse. Il est possible d'accéder à l'arbre DOM représentant la réponse XML du serveur. Vous pouvez donc utiliser toutes les méthodes vues dans le tp précédent et qui permettent d'accéder au contenu des noeuds de l'arbre (getElementsByTagName,...).

Modifiez le code de actualiserPage() afin qu'un message s'affiche dans une boîte alert avec le texte suivant : "Vente Hawaii=x, Vente Lacanau=y, Vente Biarritz=z";

```

1  if (requete.readyState == 4) {
2      if (requete.status == 200) {
3
4      //Acces a l'arbre DOM de la reponse XML
5      var reponse = requete.responseXML;
6      var Hawaii= _____
7      var Lacanau= _____
8      var Biarritz= _____
9      alert("Hawaii=" + _____ +
10     "Lacanau=" + _____
11     + "Biarritz=" + _____ );
12  }
13  }

```

### Question 8

Finalisez le code de la fonction callback afin qu'elle affiche sur la page web les ventes des trois types de planches et le bénéfice total réalisé.

---

#### 7.1.4 Envoyer des données au format XML

Il est tout a fait possible d'envoyer des données au format XML. Pour cela vous devez utiliser une requête POST (cf. cours) et surtout ne pas oublier de positionner le "Content-Type" de la requête à "text/xml" avec "setRequestHeader". Il suffit enfin d'envoyer le tout via la méthode `send()` de l'objet requête. Sauf cas très précis on utilise rarement ce mode d'envoi. Il est beaucoup facile d'envoyer de simples paires nom/valeur (en POST ou GET).

## 7.2 JSON alternative à XML

### 7.2.1 Qu'est-ce-que JSON ?

Tout comme XML, JSON est un format permettant d'envoyer et recevoir des données.

Nous allons réaliser le même exercice que précédemment mais avec JSON. Créez des copies des fichiers que venez d'écrire afin d'y apporter quelques modifications.

### 7.2.2 Envoyer des données au format JSON

#### Question 9

Modifiez le script PHP afin d'envoyer les ventes des planches au format JSON. Vous devez suivre les indications suivantes :

1. Créer un tableau dans lequel sont stockées les ventes des planches.
2. Créer un nouvel objet JSON qui servira à encoder la réponse au format JSON
3. Encoder le tableau au format JSON à l'aide de la méthode `encode` de l'objet JSON créé au point précédent
4. Retournez ces données

En PHP un tableau se construit de la manière suivante :

```
1 \ $datas = array(  
2     "fraise"    => 2,  
3     "cerise"    => 5  
4 );
```

Complétez le code PHP suivant :

```
1  
2 require_once('JSON.php');  
3 _____ = array(  
4     "Hawaii"    => _____  
5     "Lacanau"   => _____  
6     "Biarritz"  => _____  
7 );  
8  
9 $json = new Services\_JSON();  
10 $sortie = \ $json->encode(\ $datas);  
11 echo( _____ );
```

### 7.2.3 Recevoir des données au format JSON

Les données envoyées par le serveur sont prêtes à être décodées en JavaScript. L'exemple ci-dessous permet de récupérer les données.

```
1 //Formatage du texte au format JSON  
2 var donneesJson = JSON.parse(requete.responseText);  
3 alert (donneesJson.cerise); // Affichera 5  
4 alert(donneesJson['fraise']) // Affichera 2  
5 alert(donneesJson[1]);
```

#### Question 10

Modifiez le code de `actualiserPage()` afin que l'affichage des ventes soit effectif

```
1 var donneesJson = JSON.parse(requete.responseText);  
2 _____  
3 _____  
4 _____  
5 _____  
6 _____
```

## 7.3 Aller un peu plus loin (questions bonus)

Avant de passer à la suite veuillez bien à avoir fini ENTIEREMENT tous les Tps.

La suite (et fin de ce TP) se décompose en quelques exercices qui permettent d'aller un peu plus loin dans la gestion d'une interface WEB. Vous pouvez laisser courir votre imagination et essayer un maximum de choses.

### 7.3.1 Gérer plusieurs requêtes XMLHttpRequest

Vous allez apporter une modification à la version actuelle du site web. Plutôt que de devoir appuyer sur un bouton pour avoir la mise à jour des chiffres de vente de chaque planche, celle-ci se fera automatiquement.

Cet objectif nécessite quelques modifications :

- Enlevez le bouton actualiser de la page web
- Ajoutez la fonction `setTimeout("nom de la fonction à appeler", X )` qui permet d'appeler une fonction toutes les *X* milli-secondes
- Utilisez la propriété `onLoad` de la balise `body` pour indiquer que, dès le chargement de la page, il faut exécuter le fonction `getNouveauxTotaux()` ; (vous pouvez chercher de l'aide sur le net)

### 7.3.2 Utilisons une dernière fois le DOM

Faites en sorte que l'ordre d'affichage des ventes sur la page HTML soit classé par ordre croissant. Reprenez les exemples du TP précédent et modifiez le code de `actualiserPage()` en conséquence.



Ce document est publié sous Licence Creative Commons « By-NonCommercial-ShareAlike ». Cette licence vous autorise une utilisation libre de ce document pour un usage non commercial et à condition d'en conserver la paternité. Toute version modifiée de ce document doit être placée sous la même licence pour pouvoir être diffusée.

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>