Etude de cas : PARSER XML/HTML

A) Qu'est-ce que le XML/HTML?

a. Définition du XML

XML signifie eXtensible Markup Language : en français, c'est un langage de balisage extensible (markup=balise)
XML est devenu un format d'échange de données incontournable aussi bien que le format CSV.

Le **XML** permet la mise en forme de documents via l'utilisation de balises. Développé et standardisé par le World Wide Web Consortium (W3C) à la fin des années 1990, il répondait à l'objectif de définition d'un langage simple, facile à mettre en application.
Le **XML** se classe dans la catégorie des langages de description (il n'est ni un langage de programmation, ni un langage de requêtes). Il est donc naturellement utilisé pour décrire des données en s'appuyant sur des balises et des règles personnalisables.

- C'est un langage de balisage :
  - Cela signifie qu'on accole aux données des « étiquettes » qui qualifient leur contenu (qu'on appelle une balise. Une balise se repère avec des délimiteurs : **<** et **>**
- C'est un langage extensible, cela signifie deux choses en fait :
  - selon ses besoins, on peut définir et utiliser les « étiquettes » que l'on souhaite ;
  - une fois qu'un jeu d'étiquettes est défini, même par quelqu'un d'autre, on peut l'étendre en y ajoutant ses propres créations.

Il ne faut en fait pas parler de langage XML au singulier, mais bien de langages au pluriel. En effet, XML désigne un certain type de fichier texte, respectant des règles d'écriture.

b. Structure d'un fichier XML

Fichier exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<savants>
  <!-- Quelques physiciens importants -->
  <savant id="phys0">
    <identité prénom="Galileo" nom="Galilei"/>
    <dates naissance="1564-02-15" décès="1642-01-08"/>
    <contributions>Relativité du mouvement &amp; système
héliocentrique</contributions>
  </savant>
  <savant id="phys1">
    <identité prénom="Isaac" nom="Newton"/>
    <dates naissance="1643-01-04" décès="1727-03-31"/>
    <contributions>Gravitation universelle, principe d'inertie, décomposition de la
lumière &amp; calcul infinitésimal</contributions>
  </savant>
  <savant id="phys2">
    <identité nom="Einstein" prénom="Albert"/>
    <dates naissance="1879-03-14" décès="1955-04-18"/>
    <contributions><![CDATA[Relativités restreinte & générale, nature corpusculaire de
la lumière & effet photoélectrique]]></contributions>
  </savant>
</savants>
```

- **Le prologue**
  `<?xml version="1.0" encoding="UTF-8"?>`

Cette première ligne est le *prologue* du fichier. Il indique la version de la recommandation XML qui sert de base à l'écriture du fichier (il n'y a eu à cette date qu'une seule version, la version 1.0), ainsi que l'encodage de caractères utilisé. Cette ligne est facultative.

➢ **Les commentaires**

```
<!-- Quelques physiciens importants -->
```

Un commentaire, comme en HTML, commence par la chaîne de caractères <!-- et se termine par -->. Tout ce qui est entre ces deux chaînes de caractères n'est pas une donnée du document, mais un commentaire laissé par le développeur

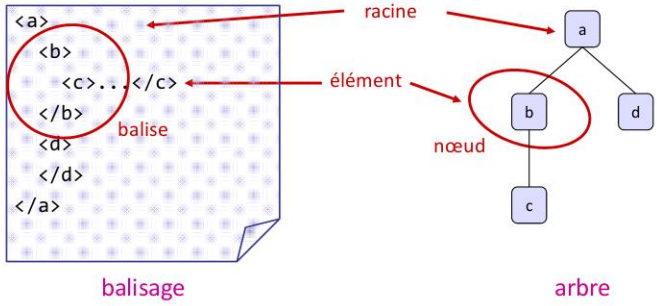
➢ **Les éléments et attributs**

Nous abordons là les étiquettes dont nous avons parlé en guise d'introduction. Ce sont les éléments et attributs qui permettent de qualifier les données, et de leur conférer un sens.

✓ *Les éléments*

Il existe deux types d'éléments : les éléments vides... et ceux qui ne le sont pas.

- Un élément non vide commence par une balise ouvrante et se termine par une balise fermante. Dans le document précédent, des exemples de balises ouvrantes sont <savants> ou <contributions> ; les balises fermantes correspondantes sont respectivement </savants> et </contributions>.
- Un élément vide ne comporte qu'une seule balise, dite « auto-fermante ». C'est le cas de l'élément identité : il est caractérisé par la balise <identité... : vous noterez qu'elle se termine par />.



L'inclusion des balises les unes dans les autres crée un lien de parenté (les balises parents ont des balises enfant dans leur intérieur). On peut représenter cette hiérarchie de parenté avec un arbre. Cette structure hiérarchique avec des nœuds et des contenus d'appelle le DOM (Document Object Model : modèle de document objet)

✓ *Les attributs*

- Quand on veut préciser ce que contient un élément, on peut placer un attribut sur la balise ouvrante. Par exemple, l'élément date comporte deux attributs, nommés naissance et décès. Un attribut possède un nom, ainsi qu'une valeur. Par exemple, l'attribut prénom du deuxième savant a pour valeur Isaac.
- Les attributs *ne sont jamais repris* dans la balise fermante : l'attribut id présent dans la balise ouvrante de l'élément savant est absent de la balise fermante </savant>.
- Enfin, l'ordre dans lequel les attributs sont écrits dans une balise ouvrante n'a aucune importance : les attributs du dernier élément savant sont inversés par rapport aux deux premiers, mais les deux formes sont strictement équivalentes.

➢ **Les entités**

Vous aurez noté que certains caractères ont une signification en XML : c'est le cas par exemple des caractères < ou >. Quand on doit les utiliser, il faut donc trouver une parade pour qu'ils ne soient pas interprétés. Pour cela, on utilise des *entités*, qui codent ces caractères par des chaînes de caractères les représentant ou des nombres.

Une entité commence par le caractère & et se termine par un point-virgule ;. Par exemple, le caractère < doit être écrit < ou <. En particulier, le caractère & doit lui-même être codé comme une entité sous la forme &

➢ **Les sections CDATA**

Lorsqu'un contenu est susceptible de contenir plusieurs entités, il peut être fastidieux de les écrire. On utilise alors ce que l'on appelle une « *section CDATA* », qui est destinée à contenir des informations affichées telles quelles, sans traitement par l'outil de consultation (navigateur Internet ou tout autre type d'application chargée de traiter le document XML).

C'est le cas dans notre exemple où l'on a écrit <![CDATA[Relativités restreinte & générale, nature corpusculaire de la lumière & effet photoélectrique]]> : il n'a pas été nécessaire de remplacer chaque occurrence de & par l'entité correspondante &

b. Définition du HTML

Le HTML (Hyper Text Markup Language) est une norme de codage XML particulière où la liste des balises et des attributs possibles par balise a été décidée par le W3C de façon fixe. Le HTML sert de langage de codage pour écrire des pages web. Il est destiné à décrire des documents Hypertexte (= texte amélioré avec mise en forme, gras, souligné, italique, couleur, images, liens, etc : ce qu'on appelle une page web)

Ici la liste complète des balises de la norme HTML : https://www.w3schools.com/tags/default.asp
Ici la liste des attributs de balises HTML autorisés selon les balises : https://www.w3schools.com/tags/ref_attributes.asp

Donc la façon dont on écrit un document HTML est identique à la façon dont on écrit un document XML, sauf que les balises ne peuvent pas être inventées comme on le souhaite, mais selon une norme précise. La norme actuelle est le HTML 5. La structure d'un document HTML précise aussi des balises obligatoires dans un document et leur ordre de parenté.

Pour indiquer que le document est de type HTML plutôt que XML, on conseille une 1ère ligne de prologue (toujours facultative) pour les documents HTML : <!DOCTYPE html>

B) Qu'est-ce qu'un parser XML/HTML?

Définition :

parser = lire et analyser syntaxiquement un flux de donnée pour le découper en unité syntaxique connexes.

C'est l'action d'exploiter un fichier sous une forme brute afin d'en tirer les informations utiles. Cette action est généralement réalisée à partir d'un fichier XML ou CSV. Afin de faciliter le traitement de l'information, on a tendance à parser le fichier afin d'en soutirer les informations importantes et les insérer dans une base de données ou une structure de données de type fichier .

Un parser est donc un analyseur du contenu d'une source de données pour chercher à en comprendre le contenu et le trier et l'organiser d'une façon ordonnée.

Concrètement, un parser est un ensemble de sous-programmes spécifiques à développer (gestionnaires ou handler), qui s'appuient sur une bibliothèque que l'on trouve dans les différents environnements de programmation (Visual studio, PHP, Python, …).

Le but ici dans notre problème est de parcourir un fichier texte au format XML ou HTML contenant des balises, des données et des attributs ayant des valeurs, et donc de pouvoir récupérer les données et les valeurs dans leur contexte (balise). Dans le cadre de l'étude de cas, il s'agit d'alimenter les traits (vecteurs de traits) à partir de fichiers contenant le code source de pages web. Le format de sortie est fixé par un format XML épuré, les valeurs binaires (positive, negative) seront déduites du nombre d'étoiles (3 minimum pour positive)

Ici on fait un parser XML/HTML car on va récupérer des données à analyser provenant d'un fichier HTML (site web), qui est un cas particulier de format XML. Les données analysées et récupérées sont ensuite stockées et organisées dans un fichier avec un certain format choisi. Ici on choisit de représenter les données stockées aussi au format XML. Donc on analyse un HTML (cas de norme XML particulière) contenant plein de données et on produit un XML contenant des données analysées, classées, triées de façon ciblée.

C) Principe d'un parser XML et HTML

a. Où peuvent être les données qui nous intéressent pour le parser?

Certaines balises permettent de structurer et ne contiennent aucune donnée.
Dans l'exemple suivant issu d'un relevé GPS :
Nous pouvons constater qu'un Trackpoint est composé d'une date/heure, d'une position, d'une altitude…
Une position est composée d'une latitude et d'une longitude.
Les valeurs effectives que nous devons rechercher peuvent se retrouver à 2 endroits.

```
<Trackpoint>
  <Time>2006-05-14T06:10:38Z</Time>
  <Position>
      <LatitudeDegrees>43.129576</LatitudeDegrees>
      <LongitudeDegrees>2.558775</LongitudeDegrees>
  </Position>
  <AltitudeMeters>201.052856</AltitudeMeters>
  <DistanceMeters>0.000000</DistanceMeters>
  <HeartRateBpm>110</HeartRateBpm>
  <Cadence>21</Cadence>
  <SensorState>Absent</SensorState>
</Trackpoint>
```

b. Valeur d'attribut

Dans l'exemple suivant :

```
<Lap StartTime="2006-05-14T06:10:38Z">
      ...
</Lap>
```

Nous avons la balise Lap qui contient l'attribut StartTime qui possède la valeur :
2006-05-14T06:10:38Z

c. Valeur entre balises

Dans l'exemple suivant : `<Cadence>21</Cadence>`
La balise Cadence contient la valeur *21*.

D) Programmation à mettre en place en PHP pour un Parser HTML

Il existe un ensemble de fonctions PHP permettant de parser des fichiers XML, de façon native. Mais comme le langage HTML est moins strict que le XML, le chargement d'une page HTML dans un parser XML risque d'échouer. Nous allons donc utiliser une bibliothèque qui a été développée pour cela : simple HTML DOM.
Il faut donc copier le fichier *simple_html_dom.php* dans l'espace de votre application et en faire un *include*.
ATTENTION : cette bibliothèque fonctionne très bien en PHP 5 mais pas en PHP 7, veillez à votre version de PHP

Comme il s'agit d'une bibliothèque orientée objet, il faut créer une instance de la classe *simple_html_dom*. Et ensuite lier le fichier à parser avec la fonction (non objet ;-(( ) :

```
file_get_html(localisation relative du fichier);
```

Vous pouvez aussi créer un parser à partir d'une chaîne de caractères contenant du code HTML avec :

```
str_get_html(untextehtml);
```

Donc le code avec l'objet nommé $html sera:

```
$html= simple_html_dom();
$html= file_get_html (localisation relative du fichier);
```

Maintenant nous pouvons récupérer une collection contenant tous les éléments html qui correspondent à la même balise :

```
$macollection_input = $html->find('input');
```

Grâce à la boucle foreach nous pouvons la parcourir :

```
foreach($macollection_input as $un_element){
      ...
}
```

Vous pouvez aussi parcourir de manière classique :

```
$nb=count($macollection);
for($i=0; $i<$nb;$i++){
      ...
}
```

Lorsque vous voulez cibler un élément par son id vous pouvez procéder ainsi :

```
$macollection = $html->getElementById('unid');
```

L'élément de la collection étant lui-même un objet, il sera doté d'un tableau attr qui permettra de récupérer la valeur de chaque attribut:

```
$un_element -> lenomdunattribut
```
ou:
```
$un_element -> attr['lenomdunattribut']
```

Vous aurez probablement besoin de tester l'existence d'un attribut car certaines balises de la même famille peuvent avoir certains attributs et d'autres non.

Vous disposez de la fonction PHP isSet:

```
isSet(unevariable)
```
renvoie true si la variable existe ou false sinon.

Et donc pour savoir si un élément possède ou non un attribut :

```
isSet($un_element -> lenomdunattribut)
```
ou :
```
isSet($un_element -> attr['lenomdunattribut'])
```

Et enfin si vous voulez accéder à ce qui est contenu entre une balise ouvrante et fermante :

```
$unelement->nodes[…]
```
ou
```
$unelement->innertext
```
ou
```
$unelement->plaintext
```

innertext et *plaintext* ne sont pas équivalents dans la façon dont ils récupèrent les données
nodes permet de récupérer les nœuds du DOM contenus dans la balise, c'est-à-dire les sous-contenus des balises situées dedans ; sous la forme d'un tableau.
innertext fait un affichage de tous les *nodes* de l'élément les uns à la suite des autres.
Donc *innertext* correspond à afficher tous les *nodes* d'un coup.

```
// Exemple : Texte HTML à analyser
$html = str_get_html("<div>foo <b>bar</b></div>");

// Renvoie le tableau des éléments ou la valeur "null"
// si balise non trouvée (avec le zéro en 2ème argument)
$e = $html->find("div", 0);

echo $e->tag; // Renvoie: " div"
echo $e->outertext; // Renvoie: " <div>foo <b>bar</b></div>"
echo $e->innertext; // Renvoie: " foo <b>bar</b>"
echo $e->plaintext; // Renvoie: " foo bar"

/*
Nom d'attribut    Utilisation
$e->tag           Lit ou écrit le nom de balise d'un élément
$e->outertext     Lit ou écrit le contenu HTML texte extérieur de l'élément.
$e->innertext     Lit ou écrit le contenu HTML texte intérieur de l'élément.
$e->plaintext     Lit ou écrit le contenu texte brut de l'élément.
*/
```

E) Compléments pour la programmation PHP

a) Parcours de dossier

Pour parcourir un dossier, il faut un objet qui représente le dossier cible:

```
$dossier = opendir(cheminrelatifdudossiercible);
```

Notes : opendir — Ouvre un dossier, et récupère un pointeur dessus

```
opendir ( string $path [, resource $context ] ) : resource|false
```

opendir() retourne un pointeur sur un dossier qui pour être utilisé avec les fonctions closedir(), readdir() et rewinddir()

Ensuite on pourra faire une boucle de parcours :

```
while($element_dossier = readdir($dossier)){
...
```

```
}
```

Notes : readdir — Lit une entrée du dossier

```
readdir ( [ resource $dir_handle ] ) : string|false
```

readdir() retourne le nom de la prochaine entrée du dossier identifié par dir_handle.
Les entrées sont retournées dans l'ordre dans lequel elles sont enregistrées dans le système de fichiers.

Par contre chaque élément du dossier peut être un fichier ou un sous-dossier.
La fonction `is_dir(unelement)` donne true si l'élément est un *dossier*, false si c'est un *fichier*.

b) Ecriture dans un fichier

Il faut d'abord créer un fichier en mode écriture :

```
$fic =fopen(localiationrelativeetnomdufichier,'w');
```

Attention: le dossier cible doit exister !

Pour écrire une ligne dans le fichier :

```
fputs($fic,letexteàécrire);
```

Note: Si vous voulez provoquer un retour à la ligne dans le fichier au format fin de ligne Windows, il faut concaténer "\r\n" à la fin de la ligne
(attention aux guillemets)

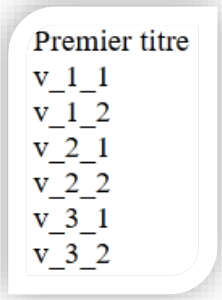
Manuel en ligne de simple dom HTML :
http://simplehtmldom.sourceforge.net/manual.htm

Vous pourrez y trouvez des compléments vous permettant de produire un code plus concis...

ACTIVITES préparatoires en PHP

Activité 1

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo1.html grâce à l'objet principal de la bibliothèque simple HTML DOM :



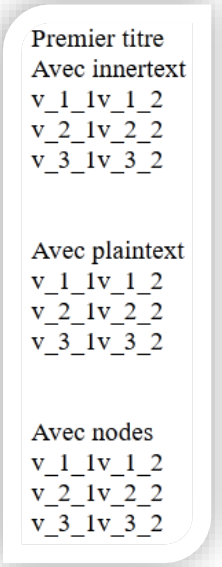
Indications :

- *Analyser le contenu du fichier de la page HTML avec un objet simple HTML DOM*
- *Parcourir l'ensemble des balises <h1> (pour récupérer le contenu de l'unique balise h1) et en afficher le contenu*
- *Parcourir l'ensemble des balises <td> et en afficher le contenu*

Rappel : lorsqu'on récupère le contenu d'une balise, qu'il y en ait une ou plusieurs dans le document, c'est un objet de type tableau associatif PHP qui est récupéré. S'il y a un seul élément à prendre, le tableau associatif aura donc une seule case, la première qu'on prend.

Activité 2

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo1.html grâce à l'objet principal de la bibliothèque simple HTML DOM :

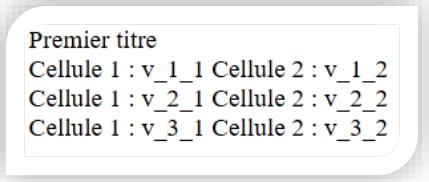


Le but est de faire le même affichage que précédemment trois fois, mais avec des méthodes différentes : (et en ajoutant un affichage texte en PHP de la méthode utilisée avant chaque affichage)

- une fois avec : `$unelement->innertext`
- une fois avec : `$unelement->plaintext`
- une fois avec : `$unelement->nodes[…]`

Activité 3

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo1.html grâce à l'objet principal de la bibliothèque simple HTML DOM :



Le but ici est de récupérer chaque ligne du contenu du tableau (chaque contenu de balise <tr>) et de traiter chacune de ces lignes de texte HTML (contenant chacune des balises HTML <td>) par un autre objet simple HTML DOM afin d'en afficher le contenu (le contenu des balises <td>).

Comme vous le constatez on fait un affichage PHP supplémentaire (« Cellule 1 », « Cellule 2 ») qui permet de faire une étiquette d'information des éléments affichés, qui sont numérotés (avec une variable de comptage de cellules en PHP)

Rappel : *pour alimenter un objet simple HTML DOM par une ligne de texte HTML :*

```
str_get_html(untextehtml);
```

On pourra aussi faire une version bis qui parcourt les lignes <tr> du tableau avec une boucle for classique.

Activité 4

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo2.html grâce à l'objet principal de la bibliothèque simple HTML DOM :

```
Tableau 1
ligne 1
Cellule 1 : v_1_1 Cellule 2 : v_1_2
ligne 2
Cellule 1 : v_2_1 Cellule 2 : v_2_2
ligne 3
Cellule 1 : v_3_1 Cellule 2 : v_3_2
Tableau 2
ligne 1
Cellule 1 : v2_1_1 Cellule 2 : v2_1_2
ligne 2
Cellule 1 : v2_2_1 Cellule 2 : v2_2_2
ligne 3
Cellule 1 : v2_3_1 Cellule 2 : v2_3_2
```

Indications :

- *Analyser le contenu du fichier de la page HTML avec un objet simple HTML DOM*
- *Parcourir l'ensemble des balises <table> (tableaux en HTML)*
- *Pour chaque <table> trouvé, afficher une information Tableau i (+ retour à la ligne)*
- *Puis analyser le contenu texte HTML de chaque balise <table> avec un objet simple HTML DOM*
- *Parcourir l'ensemble des balises <tr> (lignes du tableau)*
- *Pour chaque <tr> trouvé, afficher une information ligne i (+ retour à la ligne)*
- *Puis analyser le contenu texte HTML de chaque balise <tr> avec un objet simple HTML DOM*
- *Parcourir l'ensemble des balises <td> (cellules de la ligne du tableau)*
- *Afficher le contenu de la cellule (en le faisant précéder de l'affichage d'une information Cellule i)*
  (+ retour à la ligne après l'ensemble des contenus des cellules du <td>)

On pourra aussi faire une version bis utilisant le fichier page_exo2bis.html qui affiche les titres <h1> des tableaux

Activité 5

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo3.html grâce à l'objet principal de la bibliothèque simple HTML DOM :

```
Ciblage du tableau d'id tbl2
ligne 1
Cellule 1 : v2_1_1 Cellule 2 : v2_1_2
ligne 2
Cellule 1 : v2_2_1 Cellule 2 : v2_2_2
ligne 3
Cellule 1 : v2_3_1 Cellule 2 : v2_3_2
```

On cherche non pas à afficher le contenu de tous les tableaux de la page HTML, mais à afficher seulement le tableau qui a l'identifiant « tbl2 » du fichier HTML. On « cible » donc la balise cherchée par son id

Ensuite on adopte la même stratégie pour afficher le contenu qu'à l'activité précédente.

Rappel : *pour cibler une balise par son id:*

```
$macollection = $html->getElementById('unid');
```

Activité 6

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo4.html grâce à l'objet principal de la bibliothèque simple HTML DOM :

```
Contenu (attribut content) : text/html; charset=utf-8
Contenu (attribut content) : IUT_Carcassonne
Nom (attribut name) : author
```

Le but est ici de faire afficher le contenu des attributs de certaines balises.
Dans cet exemple on n'utilise que des balises situées dans l'en-tête du document, le <head>, mais on pourrait très bien faire exactement la même chose pour des balises situées dans le <body> ayant des attributs !

Rappel 1: *Si on a une collection de tous les éléments HTML qui correspondent à la même balise*:

```
$macollection_input = $html->find('nomdebalise');
```

Alors on peut accéder à un attribut d'une variable élément d'une collection:

```
$un_element -> attr['lenomdunattribut']
```

ou : `$un_element -> lenomdunattribut`

Avec : `$un_element` est une des « cases » du tableau associatif `$macollection_input`
Et : « `lenomdunattribut` » est un attribut de la balise « `nomdebalise` »

Rappel 2: Pour savoir si un élément possède ou non un attribut :

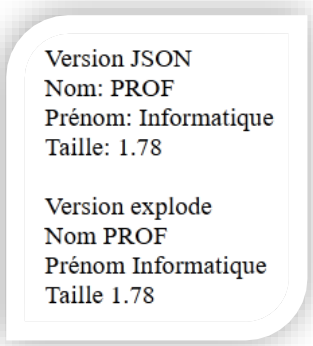
```
isSet($un_element -> attr['lenomdunattribut'])
```

ou : `isSet($un_element -> lenomdunattribut)`

→ `renvoie True si l'attribut existe`

Activité 7

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant en extrayant les données depuis le fichier page_exo5.html grâce à l'objet principal de la bibliothèque simple HTML DOM :



On récupère le contenu JSON de l'attribut 'data' :

```
{"Nom":"PROF","Prénom":"Informatique","Taille":1.78}
```

Et on affiche le contenu de deux façons différentes :

- Avec : json_decode
- Avec : explode

Concernant la fonction PHP explode : Scinde une chaîne de caractères en segments

```
explode(separateur,string,limit)
```

```
explode (string $delimiter, string $string [,int $limit = PHP_INT_MAX]): array
```

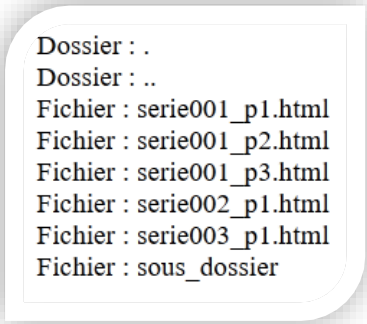
explode() retourne un tableau de chaînes de caractères, chacune d'elle étant une sous-chaîne du paramètre string extraite en utilisant le séparateur delimiter.

```
// Exemple 1
$pizza  = "piece1 piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
echo $pieces[0]; // piece1
echo $pieces[1]; // piece2

// Exemple 2
  $chaine = 'one|two|three|four';
  print_r(explode('|', $chaine));
  //Affiche: Array ( [0] => one [1] => two [2] => three [3] => four )
```

Activité 8

Ecrire le programme PHP qui permet d'obtenir l'affichage suivant (parcours du dossier rep_captures):



On parcourt l'ensemble des fichiers et dossiers contenus au premier niveau dans le dossier « rep_captures »

Rappel :

Pour parcourir un dossier, il faut un objet qui représente le dossier cible:

```
$dossier = opendir(cheminrelatifdudossiercible);
```

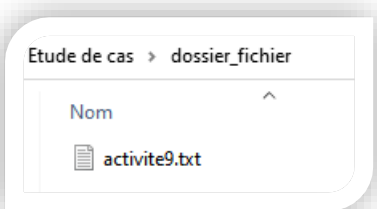
Ensuite on pourra faire une boucle de parcours :

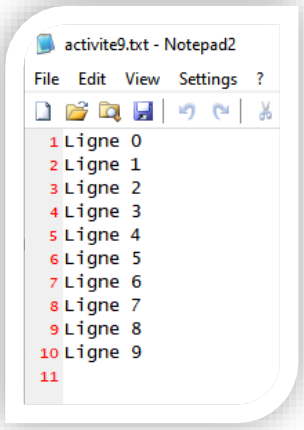
```
while($element_dossier = readdir($dossier)){
...
}
```

Activité 9

Ecrire le programme PHP qui permet de créer un fichier « activite9.txt »texte stocké dans le sous-dossier « dossier_fichier » et qui contiendra 10 lignes de la forme :

Ligne xx où xx sera une valeur allant de 0 à 9
(à alimenter par une boucle !)





Construction du parser PHP par étapes successives

Activité 10 : Parser v1 *(on parse un seul fichier)*

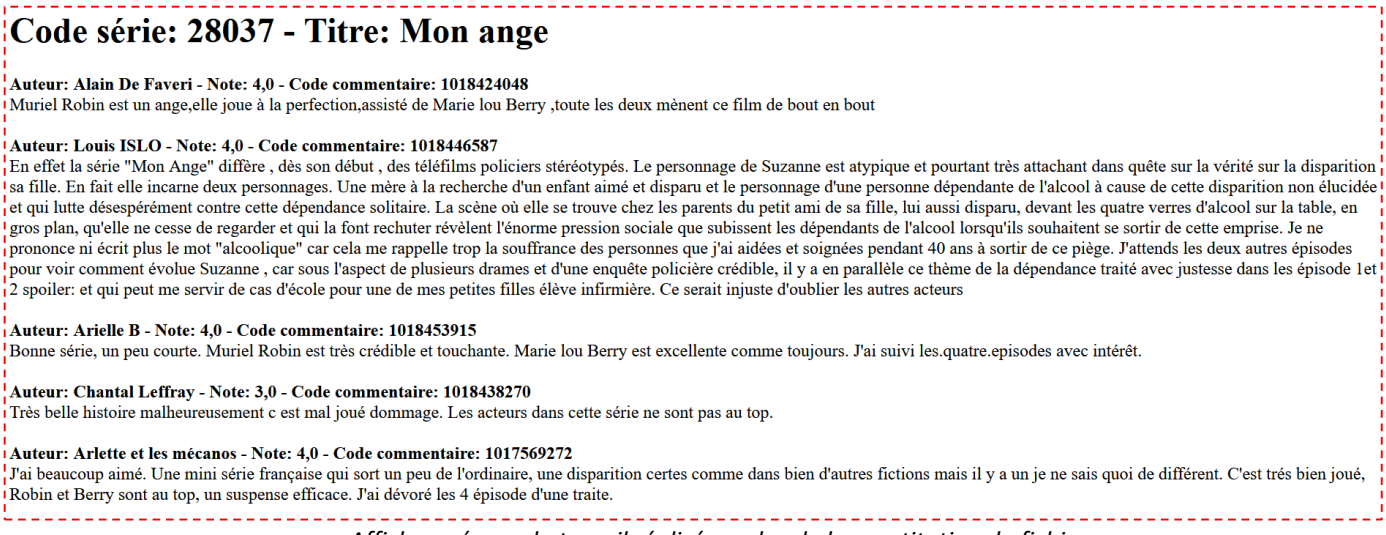
Ecrire le programme PHP qui parse le fichier « critiquesspect0001_00001-28037_001.html » du dossier « rep_capture_2022 » (contenant 61 fichiers de capture AlloCine de janvier 2022) et qui crée dans le sous-dossier « rep_sorties_xml » le fichier « corpus_ critiquesspect0001_00001-28037_001.xml ».

Le fichier XML devra respecter la forme suivante :

```
<?xml version="1.0" encoding="utf-8" ?>
<CORPUS>
<SERIE_COMMENTAIRES>
<SERIE id="28037" >Mon ange</SERIE>
<COMMENTAIRES>
<COMMENTAIRE id="1018424048_1018424048" auth="Alain De Faveri" eval="4,0">Muriel Robin est un ange,elle joue à la perfection,assisté de Marie lou Berry ,toute les deux mènent ce film de bout en bout
</COMMENTAIRE>
<COMMENTAIRE ...>
  ...
</COMMENTAIRE>
  ...
</COMMENTAIRES>
</SERIE_COMMENTAIRES>
</CORPUS>
```

Où trouver les données dans le HTML?

Balise <meta>, attribut "property", valeur=" og:title" → Titre de série
Balise <meta>, attribut "property", valeur= "apple-itunes-app" → Code de série
Balise , attribut "id", valeur= "review_..." → Bloc contenant la critique et ses informations annexes
Balise , attribut "class", valeur= "content-txt review-card-content" → Texte du commentaire critique
Balise , attribut "class", valeur= "meta-title" , puis dedans, → Auteur du commentaire critique
Balise , attribut "class", valeur= " stareval-note" → Evaluation laissée par l'auteur du commentaire critique



Code série: 28037 - Titre: Mon ange

Auteur: Alain De Faveri - Note: 4,0 - Code commentaire: 1018424048
Muriel Robin est un ange,elle joue à la perfection,assisté de Marie lou Berry ,toute les deux mènent ce film de bout en bout

Auteur: Louis ISLO - Note: 4,0 - Code commentaire: 1018446587
En effet la série "Mon Ange" diffère , dès son début , des téléfilms policiers stéréotypés. Le personnage de Suzanne est atypique et pourtant très attachant dans quête sur la vérité sur la disparition sa fille. En fait elle incarne deux personnages. Une mère à la recherche d'un enfant aimé et disparu et le personnage d'une personne dépendante de l'alcool à cause de cette disparition non élucidée et qui lutte désespérément contre cette dépendance solitaire. La scène où elle se trouve chez les parents du petit ami de sa fille, lui aussi disparu, devant les quatre verres d'alcool sur la table, en gros plan, qu'elle ne cesse de regarder et qui la font rechuter révèlent l'énorme pression sociale que subissent les dépendants de l'alcool lorsqu'ils souhaitent se sortir de cette emprise. Je ne prononce ni écrit plus le mot "alcoolique" car cela me rappelle trop la souffrance des personnes que j'ai aidées et soignées pendant 40 ans à sortir de ce piège. J'attends les deux autres épisodes pour voir comment évolue Suzanne , car sous l'aspect de plusieurs drames et d'une enquête policière crédible, il y a en parallèle ce thème de la dépendance traité avec justesse dans les épisode 1et 2 spoiler: et qui peut me servir de cas d'école pour une de mes petites filles élève infirmière. Ce serait injuste d'oublier les autres acteurs

Auteur: Arielle B - Note: 4,0 - Code commentaire: 1018453915
Bonne série, un peu courte. Muriel Robin est très crédible et touchante. Marie lou Berry est excellente comme toujours. J'ai suivi les.quatre.episodes avec intérêt.

Auteur: Chantal Leffray - Note: 3,0 - Code commentaire: 1018438270
Très belle histoire malheureusement c est mal joué dommage. Les acteurs dans cette série ne sont pas au top.

Auteur: Arlette et les mécanos - Note: 4,0 - Code commentaire: 1017569272
J'ai beaucoup aimé. Une mini série française qui sort un peu de l'ordinaire, une disparition certes comme dans bien d'autres fictions mais il y a un je ne sais quoi de différent. C'est trés bien joué, Robin et Berry sont au top, un suspense efficace. J'ai dévoré les 4 épisode d'une traite.

Affichage écran du travail réalisé en plus de la constitution du fichier

Activité 11 : Parser v2 *(on parse plusieurs fichiers)*

Ecrire le programme PHP qui parse tous les fichiers du dossier « rep_captures_2022 » et qui crée dans le sous dossier « rep_sorties_xml » le fichier « corpus.xml ».

C'est donc le travail fait précédemment à l'activité 10 mais on le fait sur tous les fichiers HTML du répertoire et le résultat est mis dans un même fichier corpus.

De plus tous les commentaires concernant une même série (de même code numérique) qui sont dans des fichiers HTML source consécutifs du répertoire (il faut que la capture soit prévue ainsi), doivent être mis dans la même section <SERIE> du fichier XML, on ne crée des sections <SERIE> différentes que pour des séries différentes.

Le fichier XML devra respecter la forme suivante :

```
<?xml version="1.0" encoding="utf-8" ?>
<CORPUS>
<SERIE_COMMENTAIRES>
<SERIE id="28037" >Mon ange</SERIE>
<COMMENTAIRES>
<COMMENTAIRE id="1018424048_1018424048" auth="Alain De Faveri" eval="4,0">Muriel Robin est un ange,elle joue à la perfection,assisté de Marie lou Berry ,toute les deux mènent ce film de bout en bout
</COMMENTAIRE>
<COMMENTAIRE ...>
  ...
</COMMENTAIRE>
  ...
</COMMENTAIRES>
</SERIE_COMMENTAIRES>

<SERIE_COMMENTAIRES>
<SERIE id="..."
>...</SERIE>
<COMMENTAIRES>
<COMMENTAIRE id="..."
auth="..." eval="...">
  ...
</COMMENTAIRE>
<COMMENTAIRE ...>
  ...
</COMMENTAIRE>
  ...
</COMMENTAIRES>
</SERIE_COMMENTAIRES>
  ...
</CORPUS>
```

corpus.xml - Notepad2

File Edit View Settings ?

```
 1 <?xml version="1.0" encoding="utf-8" ?>
 2 <CORPUS>
 3 <SERIE_COMMENTAIRES>
 4 <SERIE id="28037" >Mon ange</SERIE>
 5 <COMMENTAIRES>
 6 <COMMENTAIRE id="1018424048" auth="Alain De Faveri" eval="4,0">Muriel Robin
 7 </COMMENTAIRE>
 8 <COMMENTAIRE id="1018446587" auth="Louis ISLO" eval="4,0">En effet la série
 9  Ce serait injuste d'oublier les autres acteurs
10 </COMMENTAIRE>
                    • • •
25 <COMMENTAIRE id="1018154056" auth="Stephane Rouzeau" eval="5,0">Vue sur Sal
26 </COMMENTAIRE>
27 </COMMENTAIRES>
28 </SERIE_COMMENTAIRES>
29 <SERIE_COMMENTAIRES>
30 <SERIE id="27552" >Ne t'éloigne pas</SERIE>
31 <COMMENTAIRES>
32 <COMMENTAIRE id="1018373334" auth="Catherine V." eval="2,5">C'est du Harlan
33 Nous avons droit, une fois de plus, à un personnage qui, à première vue, mèr
34 C'est toujours à partir de là que l'intrigue commence, avec des retours en
35 Les deux dernières adaptations de H.Coben ne m'avaient pas enthousiasmée p
36 En effet, contrairement à certains qui déplorent un "twist" bien trop prévi
37 Par contre, d'où ma note qui n'est pas des meilleures, bien qu'il n'y ait "
38 </COMMENTAIRE>
                    • • •
93 <COMMENTAIRE id="1018421723" auth="Rotk79" eval="1,0">Clairement j ai tenu 1
94  Qu est ce que c est que ce choix d'actrice principale...pas à la hauteur et
95 Dommage l'intrigue avait l'air pas mal!
96 </COMMENTAIRE>
97 <COMMENTAIRE id="1018389886" auth="Serre 64" eval="4,5">Tres bonne série hal
98 </COMMENTAIRE>
99 <COMMENTAIRE id="1018452147" auth="jb bufferand" eval="3,5">sympa , plutôt c
100 </COMMENTAIRE>
101 <COMMENTAIRE id="1018435912" auth="Baba" eval="2,0">Bof bof rien de surprena
102 </COMMENTAIRE>
103 <COMMENTAIRE id="1018473945" auth="Hupatrick78340" eval="3,5">Bon thriller h
104 Par contre cette version anglaise est très lente.
105 A voir
106 </COMMENTAIRE>
107 </COMMENTAIRES>
108 </SERIE_COMMENTAIRES>
109 <SERIE_COMMENTAIRES>
110 <SERIE id="23405" >The Undoing</SERIE>
111 <COMMENTAIRES>
```

Affichage écran par le script PHP des informations recueillies ci-dessous :

Fichier traité: critiquesspect0001_00001-28037_001.html

Code série: 28037 - Titre: Mon ange

Auteur: Alain De Faveri - Note: 4,0 - Code commentaire: 1018424048
Muriel Robin est un ange,elle joue à la perfection,assisté de Marie lou Berry ,toute les deux mènent ce film de bout en bout

Auteur: Louis ISLO - Note: 4,0 - Code commentaire: 1018446587
En effet la série "Mon Ange" diffère , dès son début , des téléfilms policiers stéréotypés. Le personnage de Suzanne est atypique et pourtant très attachant dans quête sur la vérité sur la disparition sa fille. En fait elle incarne deux personnages. Une mère à la recherche d'un enfant aimé et disparu et le personnage d'une personne dépendante de l'alcool à cause de cette disparition non élucidée et qui lutte désespérément contre cette dépendance solitaire. La scène où elle se trouve chez les parents du petit ami de sa fille, lui aussi disparu, devant les quatre verres d'alcool sur la table, en gros plan, qu'elle ne cesse de regarder et qui l font rechuter révèlent l'énorme pression sociale que subissent les dépendants de l'alcool lorsqu'ils souhaitent se sortir de cette emprise. Je ne prononce ni écrit plus le mot "alcoolique" car cela me rappelle trop la souffrance des personnes que j'ai aidées et soignées pendant 40 ans à sortir de ce piège. J'attends les deux autres épisodes pour voir comment évolue Suzanne , car sous l'aspect de plusieurs drames et d'une enquête policière crédible, il y a en parallèle ce thème de la dépendance traité avec justesse dans les épisode 1et 2 spoiler: et qui peut me servir de cas d'école pour une de mes petites filles élève infirmière. Ce serait injuste d'oublier les autres acteurs

Auteur: Arielle B - Note: 4,0 - Code commentaire: 1018453915
Bonne série, un peu courte. Muriel Robin est très crédible et touchante. Marie lou Berry est excellente comme toujours. J'ai suivi les.quatre.episodes avec intérêt.

...

Auteur: Stephane Rouzeau - Note: 5,0 - Code commentaire: 1018154056
Vue sur Salto, cette série de 4 épisodes tient en haleine de bout en bout. J'ai adoré. Comme d'habitude, Muriel Robin est au top.

Fichier traité: critiquesspect0001_00002-27552_001.html

Code série: 27552 - Titre: Ne t'éloigne pas

Auteur: Catherine V. - Note: 2,5 - Code commentaire: 1018373334
C'est du Harlan Coben... donc la trame narrative est pratiquement toujours la même, mais reste toutefois assez efficace. Nous avons droit, une fois de plus, à u personnage qui, à première vue, mène une vie qui n'a rien d'extraordinaire mais qui, un jour, se retrouve bouleversé après avoir reçu soit un courrier, soit une photographie, soit après avoir croisé quelqu'un dans la rue qu'il reconnaît, etc... C'est toujours à partir de là que l'intrigue commence, avec des retours en arrière qui permettent de comprendre le pourquoi du comment, "qui est qui", et "qui a fait quoi"... Les deux dernières adaptations de H. Coben ne m'avaient pas enthousiasmée plus que ça mais celle-ci a su plutôt me séduire avec une fin que je n'ai pas senti venir. En effet, contrairement à certains qui déplorent un "twist" bien trop prévisible, je ne l'ai pas deviné dès le 2è épisode quand même. Par contre, d'où ma note qui n'est pas des meilleures, bien qu'il n'y ait "que" épisodes et un casting que j'ai su apprécier, d'autant que chaque acteur interprète bien son rôle, j'ai déploré tout de même, quelques longueurs (proches du remplissage). J'ai regretté aussi qu'il y ait trop de personnages dont certains qui n'apportent pas grand chose à l'histoire.

Activité 12 : Parser v3 *(version finalisée)*

On va enfin refaire exactement ce qu'on faisait à l'étape précédente (activité 11, Parser v2) en y apportant quelques subtiles différences.

- On va faire en sorte que dans le fichier XML corpus produit, les codes des séries ne soient pas laissés avec les codes d'origine provenant des fichiers sources mais soient renumérotés à partir de 1. Chaque nouvelle série rencontrée prendra comme code la valeur entière suivante, en commençant à 1 pour la première rencontrée. On appellera cette numérotation une numérotation locale.

- On voudra aussi que les codes commentaires soient numérotés de façon plus simple. On numérotera chaque commentaire avec le numéro de la série, suivi du souligné, suivi du numéro d'un numéro de commentaire local. Le numéro de commentaire local sera de 1 pour le premier commentaire sur la série concernée, et augmentera de 1 en 1 en prenant les valeurs entières.

Résumé :
Les codes de séries seront de la forme : xx (entier de 1 en 1 commençant à 1)
Les codes commentaires seront donc de la forme : xx_yy (avec yy entier de 1 en 1 commençant à 1 et xx le code de série)

- On peut en profiter pour compter le nombre d'octets écrits dans le fichier XML afin de ne pas constituer des fichiers XML trop gros, disons de l'ordre de 5Mo et continuer dans un nouveau fichier corpus (on numérotera les corpus créés) pour la suite dès qu'on arrive aux 5Mo. On crée ainsi une série de fichiers corpus XML de 5Mo.

- On en profite aussi pour présenter un peu mieux les données dans le fichier XML en laissant sur une même ligne la balise de commentaire et son contenu, cela rend plus lisible le fichier corpus produit.

On pourra ensuite lancer la version finale de ce Parser v3 sur le dossier « rep_captures_2022 » (contenant 61 fichiers pour 25,5Mo environ au total). Vous pourrez ainsi observer les fichiers XML produits. Si vous laissez la limite de constitution des fichiers XML à 5Mo, vous n'aurez qu'un seul fichier produit.

Le rendu est le suivant sur un des fichiers produits :

corpus_1.xml - Notepad2

File Edit View Settings ?

```
<?xml version="1.0" encoding="utf-8" ?>
<CORPUS>
<SERIE_COMMENTAIRES>
<SERIE id="1" >Mon ange</SERIE>
<COMMENTAIRES>
<COMMENTAIRE id="1_1" auth="Alain De Faveri" eval="4,0"> Muriel Robin est
<COMMENTAIRE id="1_2" auth="Louis ISLO" eval="4,0"> En effet la série "Mon
 Ce serait injuste d'oublier les autres acteurs </COMMENTAIRE>
<COMMENTAIRE id="1_3" auth="Arielle B" eval="4,0"> Bonne série, un peu cou
<COMMENTAIRE id="1_4" auth="Chantal Leffray" eval="3,0"> Très belle histoi
<COMMENTAIRE id="1_5" auth="Arlette et les mécanos" eval="4,0"> J'ai beauc
<COMMENTAIRE id="1_6" auth="Beatrice D." eval="3,5"> rien d'exceptionnel s
<COMMENTAIRE id="1_7" auth="Michele B." eval="3,5"> C'est bien , a part l'
<COMMENTAIRE id="1_8" auth="galli j" eval="3,5"> Vu sur Salto; points fort

Par contre, le dénouement n'est guère en rapport avec le casting des perso
<COMMENTAIRE id="1_9" auth="Stephane Rouzeau" eval="5,0"> Vue sur Salto, c
</COMMENTAIRES>
</SERIE_COMMENTAIRES>
<SERIE_COMMENTAIRES>
<SERIE id="2" >Ne t'éloigne pas</SERIE>
<COMMENTAIRES>
<COMMENTAIRE id="2_1" auth="Catherine V." eval="2,5"> C'est du Harlan Cobe
Nous avons droit, une fois de plus, à un personnage qui, à première vue, m
C'est toujours à partir de là que l'intrigue commence, avec des retours en
Les deux dernières adaptations de H.Coben ne m'avaient pas enthousiasmée
En effet, contrairement à certains qui déplorent un "twist" bien trop prév
Par contre, d'où ma note qui n'est pas des meilleures, bien qu'il n'y ait
<COMMENTAIRE id="2_2" auth="Jean N." eval="2,5"> Exactement le même problè
```

<u>Fichier avec reprise donné:</u> Parser v4 avec reprise *(version à utiliser sur le dossier de l'aspirateur web)*

Pour effectuer un travail assez complet dans cette étude de cas, il faut beaucoup de données.
Le programme aspirateur PHP personnalisé (qui récupère donc exclusivement les données dont on a besoin) est fourni et s'appelle «aspirateur.php». Il enregistre les fichiers dans le dossier «rep_aspirateur».

Une version du même fichier Parser v3 mais avec une fonctionnalité de sauvegarde du dernier n° de fichier XML produit, du dernier fichier source parsé et du dernier n° de série locale mémorisé a été faite.

Elle permet de sauvegarder ces informations dans un fichier et de les reprendre au lancement si ces informations sont disponibles. Ceci permet de poursuivre le travail de parsing là où il s'était arrêté en cas d'interruption du programme PHP (qui a lieu quand la durée maximale d'exécution autorisée est atteinte, soit 9999s au max avec Wampserveur).

La fonctionnalité est utile pour traiter les données volumineuses de la capture complète du dossier «rep_aspirateur».

Ainsi on peut traiter de grandes quantités de données en plusieurs fois sans recommencer.

→ La version FINALE UTILISABLE sur les données aspirées est donnée : c'est le Parser v4 (avec reprise)