

Exercice 1: (4,00)

Soit le script javascript suivant :

```
var j=((x,y)=>{
    var i=x+y;
    return (z)=> {
        let y=z; i++; y++;
        return i+y;
    }
})(2,4);
```

console.log(j(3)); 11 → 0,75
 console.log(j(3)); 12 → 0,75
 i=j(3); x=j;
 console.log(j(i)); 24 → 0,75
 console.log(x(3)); 15 → 0,75

- Quelles sont les valeurs affichées par ce script.
- écrire le même script sans utiliser les fonctions fléchées

Exercice 2: (11,50)

Soit le script PHP suivant :

```
# SQLite Database
$file_db = new PDO('sqlite:ville');
$file_db->exec("CREATE TABLE IF NOT EXISTS ville (
    id INTEGER PRIMARY KEY,
    N_ville TEXT,
    code_Post TEXT)");
// exemples: setif:19000, alger:16000, Tlemcen:13000
```

En supposant que la table ville est remplie avec les villes citées au-dessus.

- Nous voulons récupérer toutes les villes et les stocker sous les formats XML et JSON:

- Proposer la structure de document XML (bien formé) et la chaîne JSON correspondante.

Handwritten XML structure:

```
<?xml version="1.0"?>
<villes>
  <ville id="1">
    <N_ville>setif</N_ville>
    <code_Post>19000</code_Post>
  </ville>
  <ville id="2">
    <N_ville>alger</N_ville>
    <code_Post>16000</code_Post>
  </ville>
</villes>
```

Handwritten JSON structure:

```
{ "villes": [ { "id": 1, "N_ville": "setif", "code_Post": 19000 }, { "id": 2, "N_ville": "alger", "code_Post": 16000 } ] }
```

- écrire le script PHP qui permet de sélectionner toutes les villes en utilisant une requête paramétrée PDO. Puis, placer toutes les villes sélectionnées dans un document XML en utilisant l'API DOM.

Handwritten PHP code for PDO:

```
$stmt = $file_db->prepare('select * from ville');
$stmt->execute();

$xml = new DOMDocument();
$xml->appendChild($xml->createElement('villes'));
$xml->appendChild($xml->createElement('ville'));
```

Handwritten PHP code for DOM:

```
while ($v = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $ville = $doc->createElement('ville');
    $ville->setAttribute('id', $v['id']);
    $N_ville = $doc->createElement('N_ville');
    $ville->appendChild($N_ville);
    $ville->appendChild($doc->createElement('code_Post'));
    $ville->setAttribute('N_ville', $v['N_ville']);
    $ville->setAttribute('code_Post', $v['code_Post']);
    $villes->appendChild($ville);
}
```


13. En utilisant AJAX, écrire un autre script JavaScript qui permet d'interroger ce script PHP afin de recevoir toutes les villes sous forme XML. Puis afficher toutes les informations de la ville Sétif dans une balise div identifiée par divVille.

Réa 1.3 AJAX

```

0,25 VAR xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
0,25 if (this.readyState == 4 && this.status == 200) {
0,25 var data = this.responseText;
VAR V = data.getElementsByTagName('N_ville');
for (var i = 0; i < V.length; i++) {
1,00 document.getElementById("divVille").innerHTML += "ville : " +
V[i].firstChild.nodeValue + "<br>" + "code : " +
V[i].nextSibling.firstChild.nodeValue + "<br>";
}
0,25 xmlhttp.open("GET", "server.php");
0,25 xmlhttp.send();

```

Exercice 3: 14,50

Soit le document XML suivant :

```

1 <?xml version="1.0" ?>
2 <jours>
3   <jour>
4     <matin heure="8">
5       <aliment refid="A50">pain</aliment>
6       <aliment refid="A80">chocolat</aliment>
7     </matin>
8     <midi heure="12h30">
9       <aliment>viande</aliment>
10    </midi>
11    <soir>
12      <aliment>soupe</aliment>
13    </soir>
14  </jour>
15  <jour numero="52">
16    <midi>
17      <aliment>poisson</aliment>
18    </midi>
19    <soir heure="19h30">
20      <aliment>viande</aliment>
21    </soir>
22  </jour>
23  <jour numero="72">
24    <midi>
25      <aliment>pain</aliment>
26      <aliment>chocolat</aliment>
27    </midi>
28    <soir heure="21h">
29      <aliment>soupe</aliment>
30    </soir>
31  </jour>
32 </jours>

```

Réa 1.1 XSLT

```

<xsl:template match="jours">
  <table border="1">
    <tr>
      <th>jour</th>
      <th>Nb d'aliments</th>
    </tr>
    <xsl:apply-templates select="//jour"/>
  </table>
</xsl:template>
<xsl:template match="//jour">
  <tr>
    <td><xsl:choose>
      <xsl:when test=". / numero">
        <xsl:value-of select="/ numero"/>
      <xsl:when>
        <xsl:otherwise>
          <xsl:choose>
            <xsl:when test="count(. / aliment) = 2">
              2
            <xsl:when test="count(. / aliment) = 3">
              3
            <xsl:otherwise>
              4
            </xsl:choose>
          </xsl:when>
        </xsl:choose>
      </td>
    </tr>
  </xsl:template>

```

1. Proposer le programme XSLT (repas.xsl) qui affiche, jour par jour, le nombre d'aliments consommés. Le résultat doit avoir l'aspect visuel suivant:

jour	Nb D'aliments
???	4
52	2
72	3