

Introducció a l'xml

- XML eXtensible Markup Language
- Està pensat per emmagatzemar dades
- S'ha de poder llegir tant per màquines com per persones
- Està format per tags que s'obrin i tanquen englobant les dades

Hem de tenir en compte que els tags de l'xml han de representar tipus de dades, però mai haurien de representar valors.

Així

```
<dies>
  <dia pos=1>dilluns</dia>
  <dia pos=2>dimarts</dia>
</dies>
```

ho considerarem correcte, però

```
<dies>
  <dilluns>1</dilluns>
  <dimarts>2</dimarts>
</dies>
```

no ho serà.

Estructura d'un XML

Els xml's tenen estructura d'arbre

```
<arrel>
  <branca>
    <fulla></fulla>
    <fulla></fulla>
  </branca>
  <branca>
    <fulla></fulla>
    <fulla></fulla>
  </branca>
</arrel>
```

El pròleg

Defineix la versió de l'xml i la codificació de caràcters que es farà servir

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

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

L'arrel

Engloba tots els elements d'xml

```
<llibres>  
  <llibre>Programació en PHP </llibre>  
  <llibre>Programació en C++ </llibre>  
</llibres>
```

Els elements

- Formen el contingut de l'xml
- Comencen amb una lletra o un _ i són sensibles a majúscules
- No poden contenir espais al nom ni començar amb xml o variacions
- S'han d'obrir i tancar
- Han de tenir un anidament correcte
- Poden tenir a la seva vegada elements i atributs
- Hi pot haver elements buids

A l'exemple anterior, *llibre* és un element.

Els atributs

- Formen part del tag d'obertura de l'element
- Són del tipus clau -> valor
- Sempre van entre cometes "
- N'hi pot haver més d'un

```
<?xml version="1.0" encoding="UTF-8"?>
<curs>
  <materia id="001" lang="ca">Fonaments de programació</materia>
  <materia id="002" lang="en">Sistemes operatius</materia>
  <horari materia="001">
    <dia nom="dilluns">10 a 12</dia>
    <dia nom="divendres">13 a 15</dia>
  </horari>
</curs>
```

Comentaris

Es poden posar comentaris a l'XML per ajudar a entendre què és cada cosa.
Els comentaris són semblants als de l'XML.

```
<!-- això és un comentari -->
```

Entitats especials

Hi ha caràcters que no es poden escriure directament, ja que tenen un significat especial. Així

```
&lt; < menor que
&gt; > major que
&amp; & símbol and
&apos; ' apòstrof
&quot; " cometes
```

Com sabem si un xml és vàlid?

Una vegada creat, provau d'obrir el vostre xml amb un navegador web.

Si es pot obrir i no dona errors vol dir que l'estructura és legible, però si no es pot obrir vol dir que l'xml no està ben format.

PHP i SimpleXML

Aquesta llibreria ens proporciona un conjunt d'utilitats per llegir i crear documents xml des de PHP.
En lectura converteix l'arxiu xml **ben format** en un objecte que pot ser processat amb selectors de propietats i iteradors d'arrays.

Carregar un arxiu XML

Per carregar un arxiu xml executarem la funció `simplexml_load_file` passant-li com a paràmetre la localització de l'arxiu.

Heu de tenir en compte que la localització pot ser absoluta o relativa i que hi pot haver problemes de permisos si sortiu del directori on s'està executant l'aplicació.

Càrrega des d'un arxiu xml

```
if(!$xml = simplexml_load_file('data/usuaris.xml')){
    echo "No s'ha pogut carregar l'arxiu";
    die();
} else {
    echo "Arxiu carregar correctament";
}
```

Si tot ha anat bé la variable `$xml` conté l'estructura i continguts de l'arxiu.

Carregar des d'un string

El procediment és semblant a l'anterior, sols que la funció que es fa servir és `simplexml_load_string`

```
$string='<usuaris><usuari id="1">toni</usuari>
<usuari id="2">maria</usuari>
</usuaris>'
$xml = simplexml_load_string($string);
```

Carregar l'xml des d'una url

Per carregar l'xml d'una url primer hem d'obtenir el contingut de la url mitjançant la funció `file_get_contents` que ens retorna una cadena i llavors passar-ho a la funció anterior `simplexml_load_string`

Exemple

```
$url = "https://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml";  
if (!$xml = file_get_contents($url)) {  
    echo "No hem pogut carregar la url";  
    die()  
} else {  
    $xml = simplexml_load_string($xml);  
}
```

Accés als elements

Suposem que tenim el següent xml, quer haurem llegit des de un fitxer o bé directament com a string.

```

<?xml version="1.0" encoding="UTF-8"?>
<cotxes>
  <marca nom= "BMW" numero= "3">
    <cotxe assegurat="S">
      <matricula>IBX3882</matricula>
      <matriculacio>2018</matriculacio>
      <estat>Bon estat</estat>

    </cotxe>
    <cotxe assegurat="S">
      <matricula>XGP1828</matricula>
      <matriculacio>2019</matriculacio>
      <estat>Bon estat</estat>
    </cotxe>
    <cotxe assegurat="S">
      <matricula>GGH3440</matricula>
      <matriculacio>2018</matriculacio>
      <estat>Bon estat</estat>
    </cotxe>
  </marca>
  <marca nom="Citroën" numero= "2">
    <cotxe assegurat="S">
      <matricula>XPX2878</matricula>
      <matriculacio>2017</matriculacio>
      <estat>Bon estat</estat>
    </cotxe>
    <cotxe assegurat="N">
      <matricula>TXE2838</matricula>
      <matriculacio>2017</matriculacio>
      <estat>Estat deficient</estat>
    </cotxe>
  </marca>
  <marca nom= "Peugeot" numero= "0"></marca>
  <marca nom= "Dacia" numero= "1">
    <cotxe assegurat="S">
      <matricula>RZP3993</matricula>
      <matriculacio>2018</matriculacio>
      <estat>Bon estat</estat>
    </cotxe>
  </marca>
  <total>6</total>
</cotxes>

```

Una vegada carregat l'arxiu o text s'ha transformat en un objecte que pot ser recorregut com un array d'arrays que pot ser recorregut i es pot navegar pels elements.

En el nostre cas l'arrel és `cotxes` i quan llegim l'xml crearà l'equivalent a un array, podem accedir als elements per nivell, partint de l'arrel (en el nostre cas la variable `$xml`) i demanant l'array amb

elements que volguem.

Així si volem accedir a les marques tendrem:

```
echo "<table>";
foreach ($xml->marca as $marca) {
    echo '<tr>';
    echo "<td>".$marca['nom']."</td>";
    echo "<td>".$marca['numero']."</td>";
    echo '<td>';
}
echo "</table>";
```

Anidament

Si els elements a la seva vegada formen una llista, es torna aplicar el mateix criteri, és a dir, els podem recorre o accedir com si fos un array. També podem accedir-hi per posició.

Així podem accedir primer a la marca i després recorre tots els cotxes que té la marca.

Això ho fem amb el símbol de navegació ->

```

foreach ($xml->marca as $marca) {
    echo '<h1>';
    echo $marca['nom'];
    echo '</h1>';
    echo "<ul>";
    foreach ($marca->cotxe as $coche) {
        echo "<li>";
        echo "matricula: ".$coche->matricula. "matriculat: ". $coche->matriculacio. " as
        echo "</li>";
    }
    echo '</ul>';
}

```

Atributs

Se pot accedir als atributs d'un element es amb el mètode `attributes()` de l'element i

```

```php
$atributs = $usuario->attributes();
$actiu = $atributs['activo'];
echo "Actiu: $actiu
";

// o bé

echo $usuario['activo'];

```

## Crear xml per codi

La llibreria ens permet iterar, cercar per xpath i crear elements xml, podeu trobar-ne més informació a <https://www.php.net/manual/es/book.simplexml.php>