

## Tema 6. FORMULARIS

### 6.1. FORMULARIS EN HTML

Aquí trobarem un resum dels conceptes més importants associats a la definició de formularis en HTML. Un formulari HTML és una secció d'un document que conté text normal, etiquetes HTML i elements especials anomenats *controls* (caselles de verificació o checkbox, radiobuttons, menús desplegable, etc...) i ròtuls (labels) associats a aquests controls.

Els usuaris normalment completen un formulari modificant els seus controls (introduint text, seleccionant elements d'un menú, etc...), abans d'enviar el formulari a un agent per a que el processi.

#### 6.1.1. L'ELEMENT FORM

Tots els controls presents en un formulari, per a que siguin efectius, han d'aparèixer inclosos dins d'un element *form* de HTML. L'etiqueta FORM actua com a contenidor de controls, però a més, especifica:

- x El programa que manegarà les dades del formulari una vegada hagi estat completat i enviat (atribut *action*). El programa receptor ha de ser capaç d'interpretar les parelles nom/valor per poder fer ús d'elles.
- x La forma com s'enviaran les dades de l'usuari al servidor (atribut *method*).
- x El tipus MIME utilitzat per a enviar les dades del formulari (atribut *enctype*). El valor per defecte d'aquest atribut és *application/x-www-form-urlencoded*.
- x Jocs de caràcters que accepta el servidor per a poder manegar aquest formulari (atribut *accept-charset*). Els agents d'usuari poden avisar a l'usuari del valor d'aquest atribut i/o restringir a l'usuari la possibilitat d'introduir caràcters no reconeguts.

#### 6.1.2. ENVIAMENT DE FORMULARIS AL SERVIDOR

L'atribut *method* de l'element form especifica el mètode HTTP utilitzat per a enviar les dades del formulari a l'agent processador. Aquest atribut pot tenir 2 valors:

- **get**: El conjunt de dades del formulari s'afegeix (amb el caràcter ? com a separador) al URI especificat a l'atribut *action* i aquest nou URI és el que s'envia.
- **post**: les dades del formulari s'inclouen en el cos del missatge que s'envia.

El conjunt de dades del formulari que s'envia és una seqüència de parelles **nom\_de\_control/valor** construïda a partir dels elements del formulari. Cadascun dels controls té associat un nom que vé donat pel seu atribut *name*.

Cada control té un valor inicial i un valor actual, ambdós són cadenes de caràcters. En general el valor inicial d'un control es pot especificar mitjançant l'atribut *value* de l'element de control. Si l'element de control no té un valor inicial se li assigna el valor nul.

**NOTA:** El mètode *get* restringeix els valors del conjunt de dades del formulari a caràcters ASCII. Només el mètode *post* (amb *enctype="multipart/formdata"*) cobreix el conjunt de caràcters complet.

El mètode *get* s'ha d'utilitzar quan es treballi amb un formulari que no implica modificacions a una base de dades o subscripcions a un servei, per a aquestes funcionalitats s'ha de fer servir el mètode *post*. El mètode *get* és recomanable per utilitzar formularis de cerca, per exemple.

### 6.3. FORMULARIS EN PHP

Quan s'envia un formulari per al seu processament, es produexi un emparellament dels seus controls, mitjançant els seus noms, amb els valors actuals que tenen. Aquestes parelles **variable-valor** configuren el conjunt de dades del formulari que s'envia al servidor.

Com s'ha vist amb anterioritat, depenent del mètode HTTP utilitzat (*get* o *post*), unes vegades la informació s'enviarà formant part d'una cadena de consulta (*query string*) que configura la sol·licitud i, altres vegades, formant part del cos del missatge.

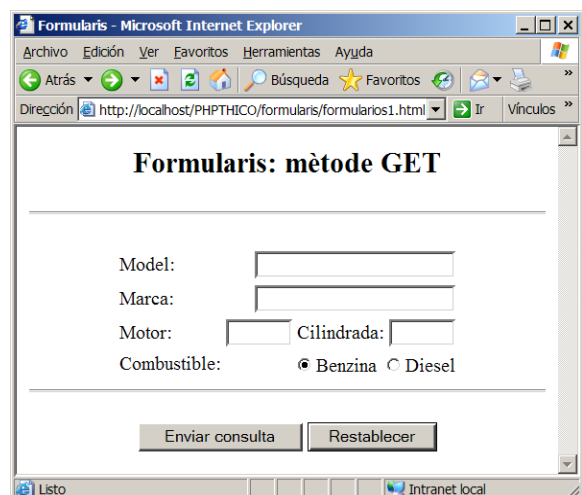
PHP, mitjançant un conjunt de variables globals, és capaç de recuperar el conjunt de dades del formulari que han estat enviats des del client per a, després, poder treballar amb ells. Les 3 variables principals per a realitzar aquesta operació són:

variable	contingut
<code>_POST</code>	Array que conté les variables passades mitjançant el mètode <b>POST</b> . El seu ús és semblant a l'array <code>HTTP_POST_VARS</code> de versions anteriors a la 4.2.0 de PHP
<code>_GET</code>	Array que conté les variables passades mitjançant el mètode <b>GET</b> . El seu ús és semblant a l'array <code>HTTP_GET_VARS</code> de versions anteriors a la 4.2.0 de PHP
<code>_REQUEST</code>	Array que conté les variables passades mitjançant qualsevol mecanisme d'entrada

### 6.3.1. FORMULARIS EN PHP 4.2.x I VERSIONS SUPERIORS

En el següent exemple es mostra el pas d'informació i el seu posterior processament fent ús del mètode *get*. Per a això es necessita un document HTML que conté el formulari que fa ús d'aquest mètode:

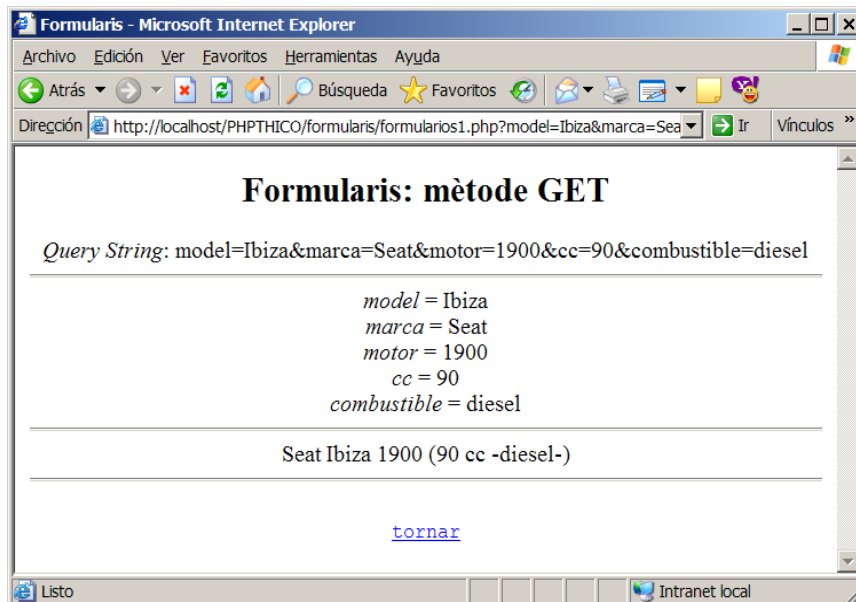
```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H2>Formularis: mètode GET</H2><HR>
    <FORM METHOD="GET" ACTION="formularios1.php">
      <TABLE>
        <TR>
          <TD ALIGN="LEFT">Model:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <INPUT TYPE="TEXT" NAME="model" SIZE="25"></TD>
        </TR>
        <TR>
          <TD ALIGN="LEFT">Marca:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <INPUT TYPE="TEXT" NAME="marca" SIZE="25"></TD>
        </TR>
        <TR ALIGN="LEFT">
          <TD>Motor:</TD>
          <TD><INPUT TYPE="TEXT" NAME="motor" SIZE="5"></TD>
          <TD>Cilindrada:</TD>
          <TD><INPUT TYPE="TEXT" NAME="cc" SIZE="5"></TD>
        </TR>
        <TR>
          <TD ALIGN="LEFT">Combustible:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <INPUT TYPE="RADIO" NAME="combustible" VALUE="benzina" CHECKED>Benzina
            <INPUT TYPE="RADIO" NAME="combustible" VALUE="diesel">Diesel
          </TD>
        </TR>
      </TABLE><HR><BR>
      <INPUT TYPE="SUBMIT"> <INPUT TYPE="RESET">
    </FORM>
  </CENTER>
</BODY>
</HTML>
```



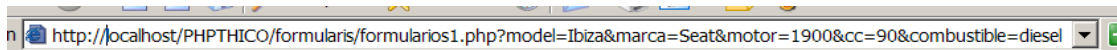
El *script* encarregat de processar aquesta informació (valor de l'atribut *action* del formulari és el següent:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <?php
      $metodo=$HTTP_SERVER_VARS['REQUEST_METHOD'];
      $cad_consulta=$HTTP_SERVER_VARS['QUERY_STRING'];
      $vars_formulario=($metodo=="GET")?$HTTP_GET_VARS:$HTTP_POST_VARS;

      echo "<H2>Formularis: mètode $metodo</H2>";
      echo "<I>Query String</I>: $cad_consulta <HR>";
      foreach ($vars_formulario as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR>$marca $model $motor ($cc cc -$combustible-)<BR><HR>";
      echo "<PRE><A HREF='javascript:history.go(-1)''>tornar</A></PRE>";
    ?>
  </CENTER>
</BODY>
</HTML>
```



A la següent imatge es pot veure com apareix en el URL tant el nom dels control com dels seus valors actuals.



## Formularis: mètode GET

En el cas d'haver utilitzat el mètode *post* per a l'enviament de les dades, la única diferència a nivell del formulari HTML seria:

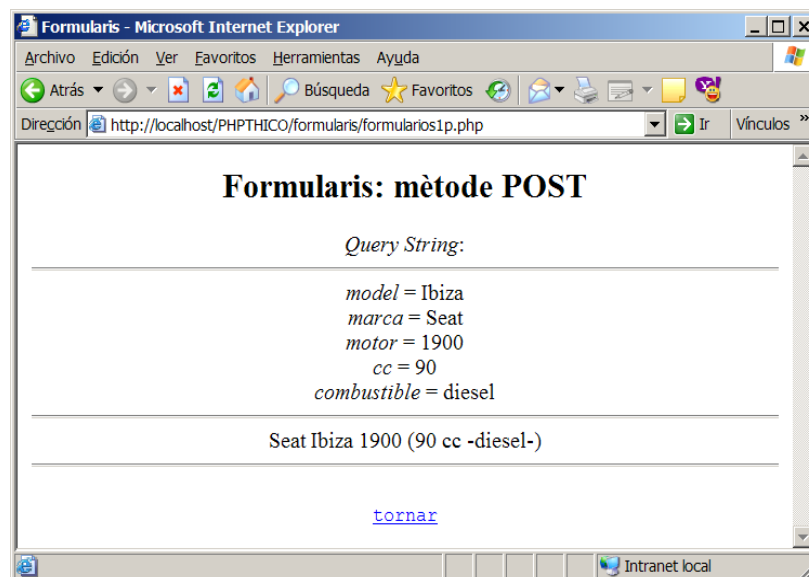
```
<FORM METHOD="POST" ACTION="formulari1p.php">
.
.
.
</FORM>
```

El codi de l'*script* sí que mostra canvis notables:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <?php
      $metodo=$_SERVER['REQUEST_METHOD'];
      $cad_consulta=$_SERVER['QUERY_STRING'];

      echo "<H2>Formularis: mètode $metodo</H2>";
      echo "<I>Query String</I>: $cad_consulta <HR>";
      foreach ($_POST as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR>$_POST[marca] $_POST[model] $_POST[motor]
            ($_POST[cc] cc -$_POST[combustible]-)<BR><HR>";
      echo "<PRE><A HREF='javascript:history.go(-1)''>tornar</A></PRE>";
    ?>
  </CENTER>
</BODY>
</HTML>
```

El resultat és igual al de l'exercici anterior, amb la diferència de que en aquest cas la cadena de consulta (*query string*) està buida:



Existeixen dos formes per a que, independentment del mètode utilitzat en el formulari, l'script per a processar-lo sigui el mateix.

El primer consisteix en fer ús d'una **variable intermitja** que reculli els continguts, com es pot veure al següent exemple:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <?php
      $metodo=$_SERVER['REQUEST_METHOD'];
      $vars_formulario=(($metodo=="GET")?$_GET::$_POST);
      $cad_consulta=$_SERVER['QUERY_STRING'];

      echo "<H2>Formularis: mètode $metodo</H2>";
      echo "<I>Query String</I>: $cad_consulta <HR>";
      foreach ($vars_formulario as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR>",$vars_formulario[marca], " ";
      echo $vars_formulario[model], " ";
      echo $vars_formulario[motor], " (";
      echo $vars_formulario[cc], " cc -",$vars_formulario[combustible],"-)<BR><HR>";
      echo "<PRE><A HREF='javascript:history.go(-1)'>tornar</A></PRE>";
    ?>
  </CENTER>
</BODY>
</HTML>
```

L'altra possibilitat és mitjançant l'ús de la variable **\_REQUEST**, que conté la informació enviada des del client al servidor independentment del mètode utilitzat:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <?php
      $metodo=$_SERVER['REQUEST_METHOD'];
      $cad_consulta=$_SERVER['QUERY_STRING'];

      echo "<H2>Formularis: mètode $metodo</H2>";
      echo "<I>Query String</I>: $cad_consulta <HR>";
      foreach ($_REQUEST as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR>$_REQUEST[marca] $_REQUEST[model] $_REQUEST[motor]
          ($_REQUEST[cc] cc -$_REQUEST[combustible]-)<BR><HR>";
      echo "<PRE><A HREF='javascript:history.go(-1)'>tornar</A></PRE>";
    ?>
  </CENTER>
</BODY>
</HTML>
```

### 6.3.2. FORMULARIS EN VERSIONS ANTERIORS A PHP 4.2

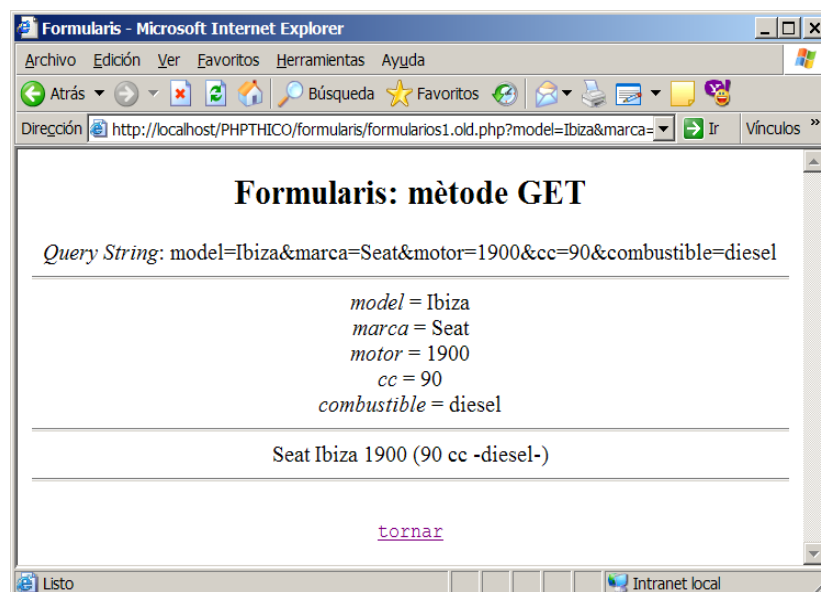
En versions anteriors a PHP 4.2.0., a més de poder accedir a les variables passades en el formulari mitjançant els arrays **HTTP\_GET\_VARS**(equivalent a l'actual **\_GET**) i **HTTP\_POST\_VARS**(equivalent a **\_POST**), tots els valors associats als controls d'un formulari estan accessibles mitjançant variables globals, és a dir, les variables d'un formulari pasaven a estar automàticament disponibles a l'script PHP, ja que aquest generava un conjunt de variables globals on els seus noms coincidien amb els donats als controls del formulari mitjançant els atributs *name*.

**NOTA:** Aquesta característica estava disponible en activar en el fitxer de configuració **php.ini** la directiva **register\_globals** (opció que apareixia activada per defecte). Actualment, per qüestions de seguretat, aquesta directiva apareix deshabilitada, si bé, per qüestions de compatibilitat, encara es permet la seva utilització.

En el següent exemple podem veure l'ús combinat d'aquestes variables de tipus global utilitzades en versions anteriors a PHP 4.2.0, amb un resultat idèntic als exemples vistos fins ara:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <?php
      $metodo=$HTTP_SERVER_VARS['REQUEST_METHOD'];
      $cad_consulta=$HTTP_SERVER_VARS['QUERY_STRING'];
      $vars_formulario=(($metodo=="GET")?$HTTP_GET_VARS:$HTTP_POST_VARS;

      echo "<H2>Formularis: mètode $metodo</H2>";
      echo "<I>Query String</I>: $cad_consulta <HR>";
      foreach ($vars_formulario as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR>$marca $model $motor ($cc cc -$combustible-)<BR><HR>";
      echo "<PRE><A HREF='javascript:history.go(-1)''>tornar</A></PRE>";
    ?>
  </CENTER>
</BODY>
</HTML>
```



### 6.3.3. FORMULARIS AVANÇATS

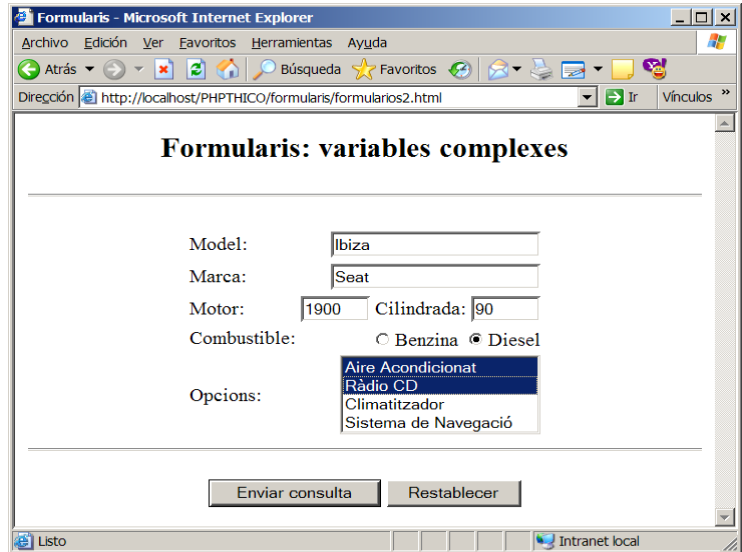
A més de la translació directa de les variables d'un formulari en variables de PHP, es pot fer un ús avançat d'aquestes variables: en el formulari HTML es poden agrupar variables relacionades en forma d'array. Posteriorment, desde PHP es pot accedir a aquests mateixos arrays i, com a tals, recòrrer-los per a obtenir les seves claus i respectius valors. Aquesta característica també es pot utilitzar per a recuperar els valors d'un camp *select* de tipus múltiple.

Podem veure un exemple a continuació:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H2>Formularis: variables complexes</H2><HR>
    <FORM METHOD="POST" ACTION="formularios2.php">
      <TABLE>
        <TR>
          <TD ALIGN="LEFT">Model:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <INPUT TYPE="TEXT" NAME="cotxe[model]" SIZE=25"></TD></TR>
        <TR>
          <TD ALIGN="LEFT">Marca:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <INPUT TYPE="TEXT" NAME="cotxe[marca]" SIZE=25"></TD></TR>
        <TR ALIGN="LEFT">
          <TD>Motor:</TD><TD><INPUT TYPE="TEXT" NAME="cotxe[motor]" SIZE="5"></TD>
          <TD>Cilindrada:</TD><TD><INPUT TYPE="TEXT" NAME="cotxe[cc]" SIZE="5"></TD>
        </TR>
        <TR>
          <TD ALIGN="LEFT">Combustible:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <INPUT TYPE="RADIO" NAME="cotxe[combustible]" VALUE="benzina" CHECKED>Benzina
            <INPUT TYPE="RADIO" NAME="cotxe[combustible]" VALUE="diesel">Diesel
          </TD>
        </TR>
        <TR>
          <TD ALIGN="LEFT">Opcions:</TD>
          <TD ALIGN="RIGHT" COLSPAN="3">
            <SELECT MULTIPLE NAME="opcions[]">
              <OPTION VALUE="AA">Aire Acondicionat</OPTION>
              <OPTION VALUE="CD">Ràdio CD</OPTION>
              <OPTION VALUE="CA">Climatitzador</OPTION>
              <OPTION VALUE="SN">Sistema de Navegació</OPTION>
            </SELECT>
          </TD></TR>
      </TABLE><HR><BR>
      <INPUT TYPE="SUBMIT"> <INPUT TYPE="RESET">
    </FORM>
  </CENTER>
</BODY>
</HTML>
```



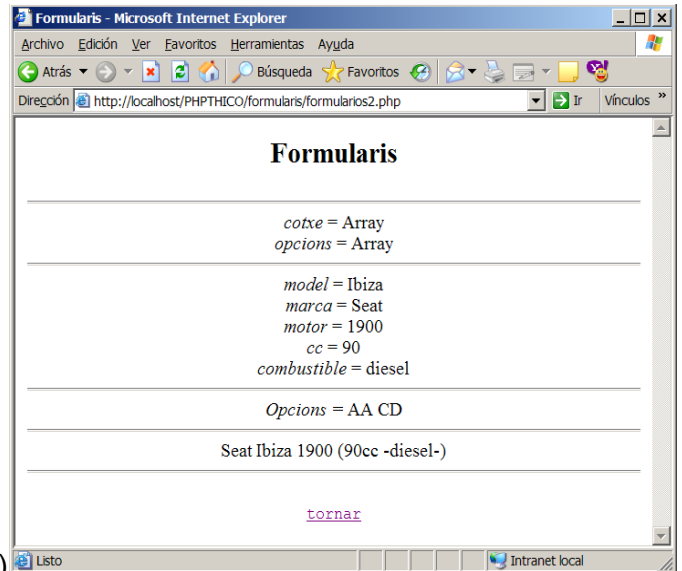
Com es pot veure al formulari, es defineixen dos variables de tipus array (un d'ells de tipus associatiu). El formulari resultant és:



Per a recuperar la informació subministrada pel client, s'utilitza el següent codi:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <?php
      echo "<H2>Formularis</H2>";
      echo "<HR>";
      foreach ($_POST as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR>";
      foreach ($_POST[cotxe] as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
      echo "<HR><I>Opcions = </I>";
      foreach ($_POST[opcions] as $clave => $valor)
        echo "$valor ";
      echo "<HR>{$_POST[cotxe][marca]} {$_POST[cotxe][model]}
        {$_POST[cotxe][motor]} {$_POST[cotxe][cc]}cc -
        {$_POST[cotxe][combustible]}-<BR><HR>";
      echo "<PRE><A HREF='javascript:history.go(-1)''>tornar</A></PRE>";
    ?>
  </CENTER>
</BODY>

</HTML>
```



### EXERCICI 1

Fer un formulari d'alta d'alumnes en html i un fitxer php posterior per mostrar les dades enviades implementant el mètode GET, POST i utilitzant la variable REQUEST. Ha de tenir un camp múltiple on es puguin escollir diferents assignatures.

Una pràctica molt habitual en PHP és agrupar en un sol fitxer tant el formulari que se li presenta al client, com el seu tractament. Encara que el funcionament intern és diferent, el resultat que veu l'usuari és idèntic.

En el següent exemple es pot veure l'ús d'aquesta forma de treball:

```
<HTML>
<HEAD>
  <TITLE>Formularis</TITLE>
</HEAD>
<BODY>
  <CENTER>
  <H2>Formulari i Resposta</H2>
  <?php
  if(!$_POST){
  ?>
  <FORM METHOD="POST" ACTION="<?php echo $_SERVER['PHP_SELF'] ?>"
  <TABLE>
    <TR>
      <TD ALIGN="LEFT">Model:</TD>
      <TD ALIGN="RIGHT" COLSPAN="3">
        <INPUT TYPE="TEXT" NAME="cotxe[model]" SIZE=25"></TD></TR>
    <TR>
      <TD ALIGN="LEFT">Marca:</TD>
      <TD ALIGN="RIGHT" COLSPAN="3">
        <INPUT TYPE="TEXT" NAME="cotxe[marca]" SIZE=25"></TD></TR>
    <TR ALIGN="LEFT">
      <TD>Motor:</TD><TD><INPUT TYPE="TEXT" NAME="cotxe[motor]" SIZE="5"></TD>
      <TD>Cilindrada:</TD><TD><INPUT TYPE="TEXT" NAME="cotxe[cc]" SIZE="5"></TD>
    </TR>
    <TR>
      <TD ALIGN="LEFT">Combustible:</TD>
      <TD ALIGN="RIGHT" COLSPAN="3">
        <INPUT TYPE="RADIO" NAME="cotxe[combustible]" VALUE="benzina" CHECKED>Benzina
        <INPUT TYPE="RADIO" NAME="cotxe[combustible]" VALUE="diesel">Diesel
      </TD>
    </TR>
    <TR>
      <TD ALIGN="LEFT">Opcions:</TD>
      <TD ALIGN="RIGHT" COLSPAN="3">
        <SELECT MULTIPLE NAME="opcions[]">
          <OPTION VALUE="AA">Aire Acondicionat</OPTION>
          <OPTION VALUE="CD">Ràdio CD</OPTION>
          <OPTION VALUE="CA">Climatitzador</OPTION>
          <OPTION VALUE="SN">Sistema de Navegació</OPTION>
        </SELECT>
      </TD>
    </TR>
  </TABLE><HR><BR>
  <INPUT TYPE="SUBMIT" NAME="enviat"> <INPUT TYPE="RESET">
</FORM>
```

```

<?php
} else {
    echo "<HR>";
    foreach ($_POST as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
    echo "<HR>";
    foreach ($_POST[cotxe] as $clave => $valor)
        echo "<I>$clave</I> = $valor <BR>";
    echo "<HR><I>Opcions = </I>";
    foreach ($_POST[opcions] as $clave => $valor)
        echo "$valor ";
    echo "<HR>{$_POST[cotxe][marca]} {$_POST[cotxe][model]} ";
    echo $_POST[cotxe][motor];
    echo " ({$_POST[cotxe][cc]}cc -{$_POST[cotxe][combustible]}-)<BR><HR>";
    echo "<PRE><A HREF='javascript:history.go(-1)'>tornar</A></PRE>";
}
?>
</CENTER>
</BODY>
</HTML>

```

**Formulari i Resposta**

Model:

Marca:

Motor:  Cilindrada:

Combustible:  Benzina  Diesel

Opcions:  Aire Acondionat  
 Ràdio CD  
 Climatitzador  
 Sistema de Navegació

**Formulari i Resposta**

*cotxe = Array*  
*opcions = Array*  
*enviat = Enviar consulta*

*model = Ibiza*  
*marca = Seat*  
*motor = 1900*  
*cc = 90*  
*combustible = diesel*

*Opcions = CD CA*

Seat Ibiza 1900 (90cc -diesel-)

[tornar](#)

Com es pot veure, aquesta forma de treball consisteix bàsicament en dividir el codi en dos seccions: quan l'usuari sol·licita la pàgina en qüestió, el primer que es troba l'interpret PHP és una sentència **if** que avalua si el formulari ha estat enviat (preguntant per la variable global encarregada d'emmagatzemar la informació transferida). En el cas que el formulari no hagi estat enviat, el fluxe de control continuarà per a tractar la part del codi que mostra el formulari HTML al client. En cas contrari, s'avaluarà l'alternativa contrària on es processen les dades enviades en el formulari.

## EXERCICI 2

Implementar l'exercici 1 en un sol fitxer. Només cal implementar un mètode.