

## PHP Perfect Form Items v1.0 (Formularios sin errores)

Cuántas veces que hemos necesitado integrar código PHP con elementos de formularios, hemos conseguido una mezcla de HTML y etiquetas `<?php ?>` disminuyendo la legibilidad del código o haciendo que tengamos que escribir más líneas de la cuenta??

Con estas sencillas pero útiles funciones podremos sin problemas emplazar campos a los formularios que diseñemos sin tantos quebraderos de cabeza.

Este documento debe ser proporcionado con su ZIP correspondiente que incluye el php de código fuente con las funciones necesarias.

Y disponible desde <http://www.distintiva.com/jose>

El código puede ser cambiado, mejorado, distribuido libremente. Ha sido programado totalmente por Jose Carlos García (yo) de Distintiva Solutions ([www.distintiva.com](http://www.distintiva.com)).  
Experto en desarrollos web, windows, linux de grandes dimensiones ...

Empecemos con el ejemplo más difícil y el causante de que tuviera que programar estas funciones.

### **<SELECT>**

Crear elementos de selección `<select></select>`

Con php es muy habitual trabajar con arrays, ya sea con datos que nos proporcione una consulta SQL o simples valores.

Veamos un ejemplo en el que queremos mostrar un `<select>` para que el usuario seleccione su rango de edad:

```
<select name='edad'>
  <option value=0>Seleccione</option>
  <option value=1>Entre 0-18</option>
  <option value=2>Entre 19-30</option>
  <option value=3>Entre 31-50</option>
  <option value=4>Más de 50</option>
</select/>
```

Esto se complica cuando hay que hacerlo de forma dinámica ya que esta lista puede variar, para evitar tener que cambiar el HTML "a pelo" ya que necesitamos construir este fragmento de HTML con un bucle etc etc..

Se complica aún más cuando una opción debe estar preseleccionada

La función que nos salvará la vida es:

**frm\_select(\$name, \$arr\_txt, \$arr\_vals, \$default="", \$extra\_tag=")**

\$name = Nombre del elemento del formulario

\$arr\_txt = Array con los textos a mostrar

\$arr\_vals = Array con los valores asociados a cada texto  
\$default= [opcional] si se indica el valor aparecerá preseleccionado dicha opción  
\$extra\_tag= [opcional] por si necesitamos incluir información adicional al select, como por ejemplo \$extra\_tag=" class=cssazul " o \$extra\_tag=" onChange=alert()"

Ejemplo:

```
$arr_txt=array('españa', 'portugal', 'francia');  
$arr_vals=array('ES', 'PT', 'FR');
```

```
<?=frm_select ('países', $arr_txt, $arr_vals) ?>
```

De esta forma cualquier modificación sólo hay que hacerla en los arrays sin alterar la parte visual.

A veces es conveniente conservar el valor de un campo del formulario entre llamadas o posts del mismo por ejemplo cuando estamos validando entradas y hay que volver al formulario para que rellene algún campo requerido. Para hacer esto de forma sencilla tan solo tenemos que usar el parámetro \$default de la siguiente forma.:

```
<?=frm_select ('países', $arr_txt, $arr_vals, $_POST['países']) ?>
```

(o \$\_GET dependiendo el método que usemos en nuestro formulario)

### **<SELECT> (tipo lista con varios elementos visibles)**

Funciona exactamente igual que el anterior pero en este caso tenemos la típica lista de selección con scroll mostrando X elementos.

La función es:

```
frm_list($name,$size, $arr_txt, $arr_vals, $default="", $extra_tag=")
```

En este caso el nuevo parámetro es \$size que indica cuántos elementos serán mostrados visiblemente en la lista.

### **<SELECT> (tipo lista con varios elementos visibles y con multiselección)**

Igual que el ejemplo anterior pero permitimos que el usuario seleccione uno o varios elementos de la lista con el CTRL+Click o SHIFT+Click

La función es:

```
frm_list_multi($name, $size, $arr_txt, $arr_vals, $default="", $extra_tag=")
```

... Y también tenemos funciones correspondientes para los demás elementos de formulario aprovechando la capacidad de mantener el valor entre posts.

### **<Checkboxes>**

No hace falta decir para que sirven estos elementos .

La función es:

**frm\_check(\$name, \$ck\_val, \$var\_in="", \$extra\_tag=")**

\$name= Nombre del campo

\$ck\_val= Valor que se enviará cuando esté seleccionado

\$var\_in=[opcional] Funciona como el \$default y permitirá que se muestre checkeado

Ejemplo:

Se muestra un checkbox que se mantiene seleccionado entre posts del formulario

```
<?= frm_check('fumador', 'SI', $_POST['fumador'])?>
```

**<Radio>**

En este caso para mostrar radiobuttons se hace exactamente como el caso anterior pero con la siguiente función:

**frm\_radio(\$name, \$val, \$var\_in="", \$extra\_tag=")**

**<INPUT>**

Permite mostrar cuadros de texto con las ventajas que voy ofreciendo en todas las funciones y para ello hay que usar:

**frm\_text(\$name, \$val, \$size, \$max\_length, \$extra\_tag=")**

\$name = Nombre del campo

\$val = Valor o texto que se mostrará (funciona como el \$default de las otras funciones)

\$size = Tamaño del campo de texto

\$max\_length = Longitud máxima permitida

**<PASSWORD>**

El caso es exactamente como el anterior pero esta vez oculta el texto con asteriscos

**frm\_password(\$name, \$val, \$size, \$max\_length, \$extra\_tag=")**