# Formularios

## Enviando los datos a la misma página del formulario

Como hemos visto anteriormente los datos los hemos enviado a través del atributo **action** a un fichero recibe.php

Pero qué pasa si queremos enviar los datos al propio formulario. Esto es muy común cuando queremos antes de tratar los datos validarlos en el propio formulario. Hay diversas formas.

Una sería dejar el atributo action en blanco o quitarlo de la url

<form method="POST" action="" id="form\_contact">

La opción de arriba lo desaconseja la W3C

Otra forma sería introducir una cadena de texto en action como hicimos anteriormente con recibe.php, pero esta vez dirigiendo a nuestro formulario

<form method="POST" action="index.php" id="form\_contact">

Existe otra forma dinámica y que es todavía mejor que la anterior puesto que no hay que pensar en que fichero.php hay que introducir para direccionar la propia página:

<form method="POST" action="<?php echo $\_SERVER['PHP\_SELF'];?>" id="form\_contact">

**$\_SERVER** es un array asociativo que contiene muchos datos de nuestro servidor para trabajar a diario.

**PHP\_SELF** es el índice que nos carga nuestro script/página actual

Esta forma tiene un problema de seguridad. Con lo cual la opción óptima es encerrarlo con la función htmlspecialchars()

<form method="POST" action="<?php echo htmlspecialchars($\_SERVER['PHP\_SELF']);?>" id="form\_contact">

## Comprobar si un formulario fue enviado.

Hay dos formas de comprobar si un formulario es enviado. Uno utilizando la variable global $\_SERVER

if($\_SERVER['REQUEST\_METHOD']=='POST'):

        echo "<pre>";

        print\_r($\_POST);

        echo "</pre>";

        extract($\_POST);

    endif;

Observamos que utilizamos el índice/key REQUEST\_METHOD que nos devuelve el método de envío que puede ser GET (por defecto) o POST.

La otra forma más tradicional es poner un atributo name al botón de envío.

<input type="submit" name="submit" class="btn btn--primary" value="Enviar correo">

A pregunta por esta key/índice el la variable $\_POST o $\_GET según el método de envío:

if($\_POST['submit']):

   echo "<pre>";

   print\_r($\_POST);

   echo "</pre>";

   extract($\_POST);

endif;

El problema surge si el formulario no ha sido enviado. $\_POST o $\_GET existe siempre. Pero el índice (los atributos name de los controles) se crea una vez enviado el formulario por lo que obtendremos un error como éste:

**Notice**: Undefined index: submit in **C:\Users\xurxo\Desktop\CURSO\_MF0492\_3\php\PROYECTOS\FORMARIO\_1\index.php** on line **17**

Existe una función que comprueba si una variable ha sido declarada que es isset(). Por tanto modificaremos el código de la siguiente manera:

if(isset($\_POST['submit'])):

   echo "<pre>";

   print\_r($\_POST);

   echo "</pre>";

   extract($\_POST);

endif;

¿Qué opción coger? La primera opción es la mejor si sólo hay uno. Pero el problema es que si en una página hay dos formularios no sabremos qué formulario estamos tratando puesto que no nos lo indica.