

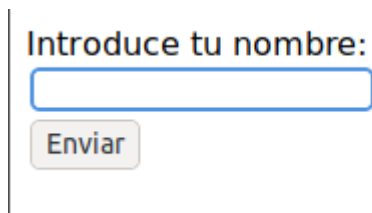
2. Formularios

2.1 Recogida de datos

En una página web, los datos se introducen mediante formularios. En la mayoría de las ocasiones tendremos dos páginas: una página con un formulario que recoge los datos y otra página que los procesa.

A continuación tenemos una página con nombre index.php (se podría llamar también index.html ya que no contiene código PHP):

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7     Introduce tu nombre:
8     <form action="saluda.php" method="get">
9         <input type="text" name="nombre"><br>
10        <input type="submit" value="Enviar">
11    </form>
12 </body>
13 </html>
14 |
```



Introduce tu nombre:

Al pulsar el botón Enviar, el contenido del formulario se envía a la página que indicamos en el atributo action. La página saluda.php tendría el siguiente código:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7 Hola
8
9 <?php echo $_GET['nombre'] ?>,
10
11 bienvenido a nuestro centro.
12 </body>
13 </html>
14 |

```

Hola Javier, bienvenido a nuestro centro.

Observa que la información que se quiere enviar se introduce en un cuadro de texto dentro del formulario (página index.php). A esa información la llamamos en este caso nombre. Para recoger esa información se utiliza `$_GET['nombre']`.

Cuando los datos que se recogen y se manipulan son números en lugar de cadenas de caracteres, el procedimiento es el mismo.

En el siguiente ejemplo tenemos una aplicación que suma dos números. El programa index.php recoge los datos y el programa suma.php suma los dos números que recibe y muestra el resultado.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7 Introduce tu nombre:
8 <form action="saluda.php" method="get">
9 <input type="text" name="nombre" value=""><br>
10 <input type="submit" value="Enviar">
11 </form>
12 </body>
13 </html>
14

```

Calcula la suma de dos números

Primer número:

Segundo número:

Fíjate que los campos input del formulario son de tipo number. Se podrían declarar de tipo text y el programa funcionaría sin problemas pero se declaran number para que el formulario no deje pasar los datos si se introduce algo que no sea un número. Imagínate que el usuario introduce la palabra hola en un campo en lugar de un número. Si el input es de tipo text, la palabra hola llegaría a suma.php y se intentaría sumar, lo que provocaría un error. Sin embargo, si el input es de tipo number, el formulario no envía hola sino que muestra un mensaje diciéndole al usuario que introduzca un número.

A continuación se muestra el fichero suma.php:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7 <?php
8     $a = $_GET['a'];
9     $b = $_GET['b'];
10    echo "La suma de $a mas $b es ", $a + $b;
11 ?>
12 </body>
13 </html>
```

Como hemos visto anteriormente, en PHP no hace falta declarar las variables especificando el tipo. Una misma variable puede contener incluso valores de distintos tipos durante su vida útil. Esto hace que no necesitemos ninguna conversión como habría que hacer utilizando otros lenguajes.

2.2 Métodos GET y POST

Mediante el atributo method de la etiqueta form se debe especificar un método de envío; los dos métodos posibles son get y post. El resultado final es el mismo, la única diferencia radica en que con

el método get se pueden ver los parámetros que envía el formulario en la barra de direcciones del navegador. Metiante en método POST se envía en el cuerpo de la petición HTTP.

Ejercicio 2.1

Programa una web que convierta de millas a kilómetros.

Ejercicio 2.2

Programa una web que dados dos números, calcule suma, resta, multiplicación y división de ambos números.

.