

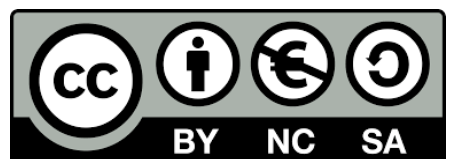
# PHP

## `filter_input()`

Antonio Boronat

Desarrollo Aplic Web Servidor

CFGs DAW-2



## Índice

filter_input().....	3
---------------------	---

## filter\_input()

Cuando una aplicación Web toma datos para que sean procesados, es conveniente que sean filtrados para evitar problemas tales como la inyección de código malicioso, cambios en las trayectorias de los componentes de la aplicación o simplemente que estén mal formateados y no puedan ser tratados adecuadamente, por ejemplo cuando se pide un dato numérico, una URL o una dirección de correo electrónico y el valor introducido no se corresponda con el esperado. Esto es especialmente importante cuando los datos se recuperan de formularios HTML cuyos valores los proporcionan los usuarios.

El proceso de filtrado puede realizarse en parte en el propio formulario aplicando las facilidades ofrecidas por HTML 5, pero aún así se recomienda emplear algún tipo de filtrado en la aplicación, esta puede establecerse con la función PHP *filter\_input()* :

```
$dato_form=filter_input(TIPO,"name_del_input",FILTER[,FILTER][,$op_filtros]);
```

donde:

TIPO será una constante que indica el array de datos de entrada que se va a leer y filtrar, los valores son: INPUT\_GET es \$\_GET[], INPUT\_POST es \$\_POST[], INPUT\_COOKIE es \$\_COOKIE[], INPUT\_SERVER es \$\_SERVER[] y INPUT\_ENV es \$\_ENV[]. Como vemos, hacen referencia a los arrays de entrada que ofrece PHP para obtener datos. En el caso de formularios las que emplearemos son INPUT\_GET e INPUT\_POST según se haya definido en el formulario.

El segundo parámetro, "name\_del\_input", será el nombre del campo del formulario que se desea obtener, será el valor del atributo *name* del *input* del formulario, por ejemplo \$\_POST["entrada"].

Finalmente, se indican filtros a aplicar, si no se pone ninguno, se toma el que haya definido por defecto, si no se establece, no realiza ningún filtrado y devuelve el valor de la variable. Se puede poner más de un filtro en la llamada a la función. Al final, como último parámetro, se pueden añadir opciones que determinen el comportamiento de alguno de los filtros si es necesario. [Lista de filtros](#).

El resultado de *filter\_input()* será el valor de la variable filtrada o FALSE si falla algún filtro o NULL si la variable indicada no está definida.

El código mostrado a continuación, permite comprobar de forma sencilla la necesidad de aplicar filtros en los datos recibidos desde formularios, está el mismo con y sin filtrado para ver la diferencia en su ejecución.

```
<!DOCTYPE html>
<!--
Toma una línea de texto y la ejecuta al no estar filtrada
-->
<html>
  <head>
    <meta charset="UTF-8">
    <title>Prova input_filter()</title>
  </head>
  <body>
    <a href="filtrado.php">Con filtro</a>
    <p>Pon en el text alguna de las órdenes siguientes y comprueba el resultado:</p>
    <form name="inputfilter" method="post" action="<?php $PHP_SELF ?>" >
      <input type="text" name="entrada">
      <input type="submit" name="enviar">

    </form>
  </body>
</html>

<?php
$linea = "&lt;h1&gt;HOLA&lt;/h1&gt;";
echo $linea . "<br>";
$linea='&lt;meta http-equiv="refresh" content="1; url=http://www.levante-emv.com"&gt;';
echo $linea. "<br>";
$linea='&lt;script&gt; window.location.replace("http://www.levante-emv.com");&lt;/script&gt;';
echo $linea. "<br>";
if(isset($_POST["entrada"])){
  echo $_POST["entrada"];
}
?>
```

```
<!DOCTYPE html>
<!--
Toma una línea de texto y la ejecuta pero filtrada y no tiene efecto
-->
<html>
  <head>
    <meta charset="UTF-8">
    <title>Prova filter_input()</title>
  </head>
  <body>
    <a href="index.php">Sin filtro</a>
    <p>Pon en el text alguna de las órdenes siguientes y comprueba el resultado:</p>
    <form name="inputfilter" method="post" action="<?php $PHP_SELF ?>" >
      <input type="text" name="entrada">
      <input type="submit" name="enviar">

    </form>
  </body>
</html>

<?php
```

## filter\_input

---

```
$linea = "&lt;h1&gt;HOLA&lt;/h1&gt;";  
echo $linea . "<br>";  
$linea='&lt;meta http-equiv="refresh" content="1; url=http://www.levante-emv.com"&gt;';  
echo $linea. "<br>";  
$linea='&lt;script&gt; window.location.replace("http://www.levante-emv.com");&lt;/script&gt;';  
echo $linea. "<br>";  
$filtered = filter_input(INPUT_POST,"entrada", FILTER_SANITIZE_STRING);  
if($filtered != NULL){  
    echo "<br>" . $filtered . "<br>";  
}  
?>
```

Puedes consultar como ejemplo la solución propuesta para el ejercicio Tema 4 – 12.