Índice

[**¿Cómo se renderiza una página web en mi computadora?** 1](#_Toc107772787)

[**Imprime texto y etiquetas HTML** 1](#_Toc107772788)

[**Condicionales** 1](#_Toc107772789)

[**Envío de un formulario a través de GET** 2](#_Toc107772790)

[**Envío de un formulario a través de POST** 3](#_Toc107772791)

[**Tipos de input** 5](#_Toc107772792)

[**Input simple** 5](#_Toc107772793)

[**Arreglos** 5](#_Toc107772794)

[**Arreglos asociativos** 5](#_Toc107772795)

[**Checkbox** 5](#_Toc107772796)

[**Radios** 6](#_Toc107772797)

[**Múltiples archivos** 6](#_Toc107772798)

[**Valida si un formulario ha sido enviado** 6](#_Toc107772799)

[**Sanitizado de datos** 6](#_Toc107772800)

[**Varias formas de sanitizar datos** 7](#_Toc107772801)

# **¿Cómo se renderiza una página web en mi computadora?**

**Client Side Rendering** (CSR): El navegador del cliente arma la estructura de la página web con JavaScript.

**Static Side Generation** (SSG): La página web se construye al momento del **build** o compilación de la aplicación en la máquina del desarrollador.

**Sever Side Rendering** (SSR): El servidor se encarga de procesar y generar la página web, este es el caso de PHP.

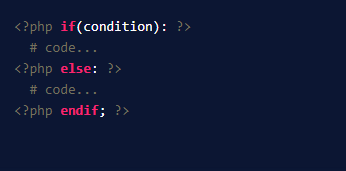
# **Imprime texto y etiquetas HTML**

Cuando PHP analiza un fichero, busca las etiquetas de apertura y cierre, que son **<?php** y **?>**, y que indican a PHP dónde empezar y finalizar la interpretación del código. Este mecanismo permite embeber a PHP en todo tipo de documentos, ya que todo lo que esté fuera de las etiquetas de apertura y cierre de PHP será ignorado por el compilador.

Si un fichero contiene **solamente** código de PHP, es preferible omitir la etiqueta de cierre de PHP al final del mismo.

# **Condicionales**

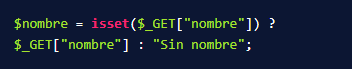
La forma correcta usar el **ifelse** de php en html:



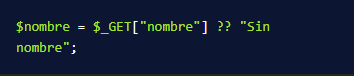
# **Envío de un formulario a través de GET**

Una forma de validar que nuestras variables, recibidas por **GET** en este caso, estén definidas es mediante el operador ternario y la función **isset**.

Los datos son mandados por la URL.

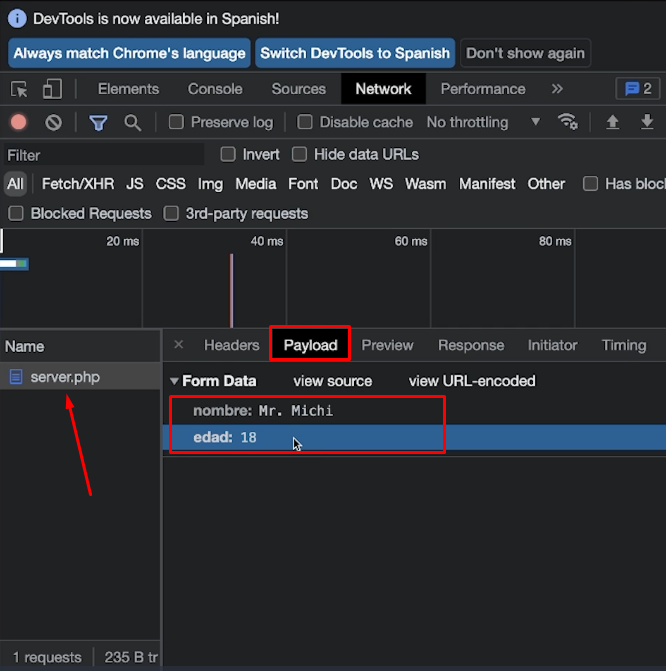


Otra forma es usando **Null Coalescing Operator** (sólo disponible desde PHP 7.1 o superiores)



# **Envío de un formulario a través de POST**

Los datos son mandados por detrás, enmascarados, pero esto no significa que sean totalmente ocultos al usuario. Para verlos, podemos abrir devtools e irnos a **red**/**network**, click en el archivo que recibe el request y en la pestaña **header** podemos ver la información. Si nos dirigimos a Payload en esa sección e visualizan toda la información que se manda en el formulario.

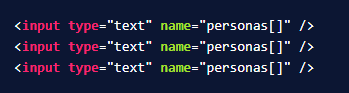


# **Tipos de input**

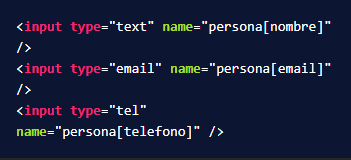
## **Input simple**



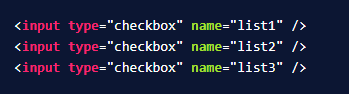
## **Arreglos**



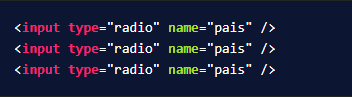
## **Arreglos asociativos**



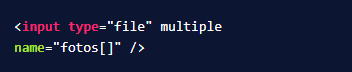
## **Checkbox**



## **Radios**



## **Múltiples archivos**



# **Valida si un formulario ha sido enviado**

* **isset**: Comprueba si se ha seteado un dato o valor, pero no valida si está vacío o no.
* **empty** (vacío en inglés): Valida si un valor está vacío o no.

# **Sanitizado de datos**

**Es el proceso que no permite el acceso a los datos**, es decir, que los datos no se recuperen fácilmente.

En otras palabras, más simples, es validar los datos que llegan desde el frontend para que no se inyecte código malicioso.

## **Varias formas de sanitizar datos**

