

PHP: Servir contenido



**CODE
SPACE**
ACADEMY

Carlos Herrera

Índice

- Responder peticiones con HTML
- Responder peticiones con JSON
- Headers
- CORS
- Superglobals

1. Mostrar información en la pantalla

Para imprimir la información en PHP se utiliza **echo** o **print()** y el valor de salida se puede visualizar en el navegador.

```
<?php  
  
$nombre = 'Carlos';  
  
echo $nombre;  
  
print_r($nombre);
```

EJERCICIO: Realizar una petición a un script PHP desde el navegador y mostrar en la pantalla la información contenida en un array, primero con echo y luego con print_r().

2. Responder peticiones con contenido HTML

El propósito de un lenguaje de servidor es responder a peticiones con contenido HTML (servidor tradicional) o con contenido JSON (servidor API)

EJERCICIO:

Responder a una petición con el siguiente HTML:

```
<h1>Hola, Carlos</h1>  
<p>Hoy es <strong>viernes</strong>. ¿Qué tal estás?</p>
```

NOTA: El día de la semana debe programarse para que responda siempre con el día de la semana correcto.

3. Responder peticiones con contenido JSON

```
<?php  
  
$seasons = ["Primavera", "Verano", "Otoño,", "Invierno"];  
  
echo json_encode($seasons);
```

EJERCICIO:

Responder a una petición con un JSON:

4. Definir las cabeceras de la respuesta

```
<?php  
  
header('Content-type: application/json');  
  
$seasons = ["Primavera", "Verano", "Otoño,", "Invierno"];  
  
echo json_encode($seasons);
```

5. CORS

Las CORS (Cross-Origin Request Headers) son unas cabeceras utilizadas para conceder permisos para acceder a un recurso alojado en un servidor.

Por defecto, si un JavaScript en un dominio X intenta realizar una petición a un servidor en un **dominio Y diferente** al X, la petición se debe **denegar**.

Un servidor puede establecer las cabeceras CORS para indicar a qué dominios da permiso para que le hagan peticiones.

```
<?php
```

```
header('Content-type: application/json');
```

```
header('Access-Control-Allow-Origin: *');
```

```
header("Access-Control-Allow-Headers: X-API-KEY, Origin, X-Requested-With, Content-Type, Accept, Access-Control-Request-Method");
```

```
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
```

```
$seasons = ["Primavera", "Verano", "Otoño,", "Invierno"];
```

```
echo json_encode($seasons);
```

6. SUPERGLOBALS

PHP dispone de una gran cantidad de variables predefinidas. Entre ellas cabe destacar las **"Superglobals"** cuya finalidad es mantener un canal de comunicación entre el cliente y el servidor.

El usuario accede a una URL en el sitio web PHP y este recopila la información del Cliente.

Estas son las "Superglobals" más utilizadas:

- `$_GET`: Contiene información sobre los "query params"
- `$_POST`: Contiene la información enviada en una petición POST (los "name" de los elementos de un `<form>`)
- `$_FILES`: Contiene la información relacionada con los `<input type="file">`
- `$_COOKIES`: Contiene las Cookies guardadas en el cliente.
- `$_REQUEST`: Es la fusión de `$_GET`, `$_POST` y `$_COOKIE`.
- `$_SERVER`: Contiene información acerca de cabeceras, ip del cliente, ...
- `$_SESSION`: Contiene información de sesión
- `$GLOBALS`: Contiene todas las variables definidas en el ámbito global del script.

7. Acceder al body de una request

```
<?php
```

```
$requestBody = file_get_contents('php://input');  
var_dump($requestBody);
```

```
$data = json_decode($requestBody, true);  
var_dump($data);
```

DESCANSO



**CODE
SPACE**
ACADEMY