Bloque B

Páginas Web Dinámicas

Unidad 9. Cookies & Sesiones

Contenidos

1. Introducción

Introducción

Para crear páginas web que contengan datos personales como un nombre de usuario, una foto de perfil o una lista de páginas vistas recientemente, **el sitio necesitaría saber quién solicita cada página**.

El protocolo HTTP proporciona reglas que especifican cómo debe solicitar un navegador una página web y cómo debe responder el servidor, pero **trata cada solicitud y respuesta por separado**. HTTP **no proporciona un mecanismo para que un sitio web sepa qué visitante está solicitando una página**.

Introducción

Si un sitio necesita saber quién solicita una página web o mostrar información personalizada, puede hacer un **seguimiento de cada visitante** y **almacenar información sobre sus preferencias** utilizando una **combinación de cookies y sesiones**.

- Las **cookies** son archivos de texto que se almacenan en el navegador del usuario. Un sitio puede indicar al navegador qué datos almacenar en una cookie, y el navegador devolverá esos datos al sitio web con cada página posterior que solicite de ese sitio.
- Las **sesiones** permiten a un sitio almacenar temporalmente datos sobre un usuario en el servidor. Cuando un visitante solicita otra página del sitio, el intérprete de PHP puede acceder a los datos de la sesión de ese usuario.

Introducción

Las cookies y las sesiones **almacenan pequeñas cantidades de datos temporalmente**, pero no se garantiza que almacenen datos durante un largo periodo de tiempo porque los usuarios pueden borrar las cookies (o acceder al sitio desde un navegador diferente que no tenga la cookie), y las sesiones sólo están diseñadas para durar lo que dura una visita al sitio (no almacenan datos entre visitas).

Cuando los datos sobre los usuarios deben almacenarse durante **períodos más largos**, **se guardan en una base de datos**. Aprenderás cómo hacerlo en la unidad 13. Para ello es necesario saber cómo funcionan las cookies y las sesiones.

2. Cookies

Cookies

Un sitio web puede indicar a un navegador que almacene datos sobre el usuario en un archivo de texto llamado **cookie**. Entonces, cada vez que el navegador solicita otra página de ese sitio, envía los datos de la cookie al servidor.

Cookies

¿Que es una cookie?

Un sitio web puede indicar a un navegador que cree una cookie, que es un archivo de texto que se almacena en el navegador.

Cada cookie tiene un nombre, que debe describir el tipo de información que contiene. El nombre de la cookie será el mismo para cada visitante.

El valor almacenado en la cookie de cada usuario puede cambiar. Así que **una cookie es como una variable que se almacena en un archivo de texto en el navegador del usuario**.

Cookies

Creando cookies

Cuando un navegador solicita una página web, el sitio web puede enviar una cabecera HTTP adicional de vuelta al navegador con la página.

Esa cabecera HTTP indica al navegador el nombre de la cookie que debe crear y el valor que debe almacenarse en la cookie.

El valor almacenado en la cookie es **texto**, y no debe tener más de 4.096 caracteres. **Un sitio también puede crear más de una cookie**.

Cookies

Obteniendo datos de las cookies

Si el navegador solicita otra página del sitio que creó la cookie, enviará el nombre de la cookie y el valor que almacena al servidor junto con la solicitud de página.

El intérprete de PHP añade entonces los datos de la cookie a un array superglobal llamado `$_COOKIE` para que el código PHP de esa página pueda utilizarlos. **El nombre de la cookie es la clave, y su valor es el valor que la cookie almacenó**.

Cookies

¿Quién puede acceder a una cookie?

Los navegadores sólo envían datos en una cookie a un servidor cuando éste solicita una página del mismo dominio que la creó.

Por ejemplo, si google.com crea una cookie, sólo se envía cuando el navegador solicita páginas de google.com. Nunca se enviaría a facebook.com.

> JavaScript, en general, también puede acceder a los datos de las cookies.

Cookies

Las cookies están vinculadas a un navegador

Porque es el navegador el que crea y almacena las cookies:

- Si hay más de un navegador instalado en un dispositivo, **la cookie sólo se envía desde el navegador en el que se almacenó** (no desde cualquier otro navegador instalado en ese dispositivo).
- Si un usuario consigue un nuevo dispositivo, ese dispositivo **no tendrá la cookie para enviar al servidor**.

Cookies

¿Cuánto duran las cookies?

El servidor puede **especificar una fecha y hora en la que expira una cookie**. Esta es la fecha y hora en la que el navegador debe dejar de enviar los datos de la cookie al servidor. Si no se indica una fecha de caducidad, **el navegador deja de enviar la cookie al servidor cuando el usuario cierra su navegador**.

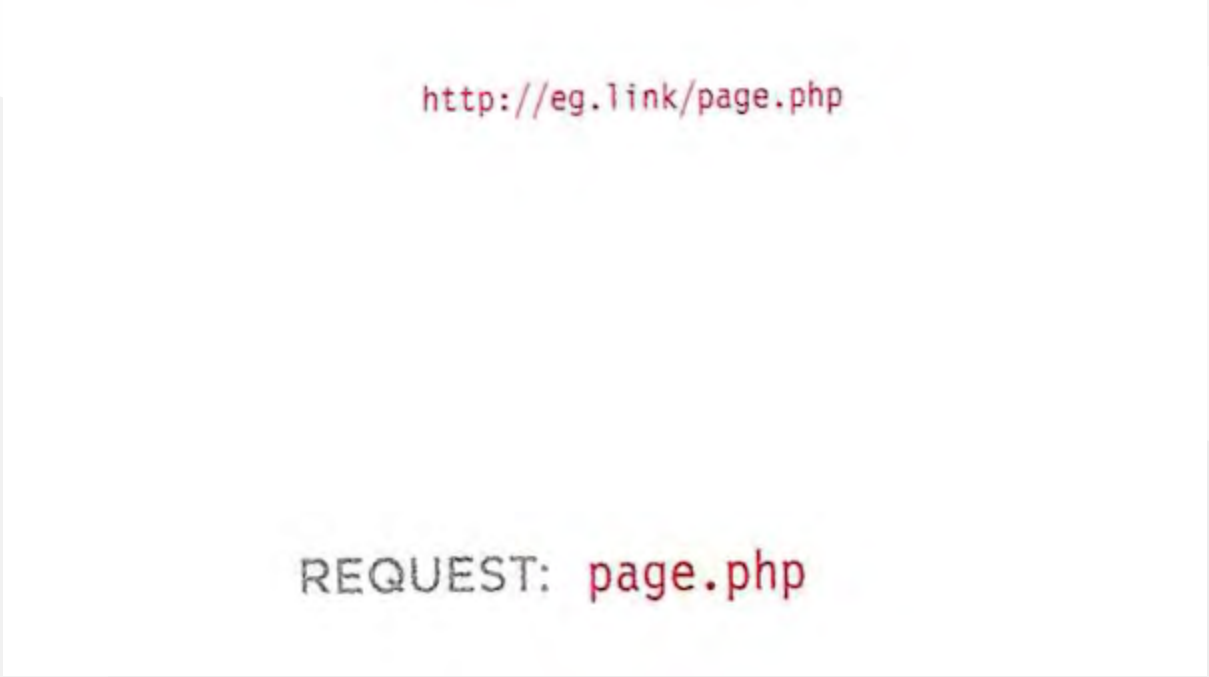
> Los usuarios también pueden rechazar o eliminar las cookies, por lo que un sitio debería poder funcionar sin ellas.

Cookies

Ejemplo de peticiones usando Cookies

Primera solicitud de la página

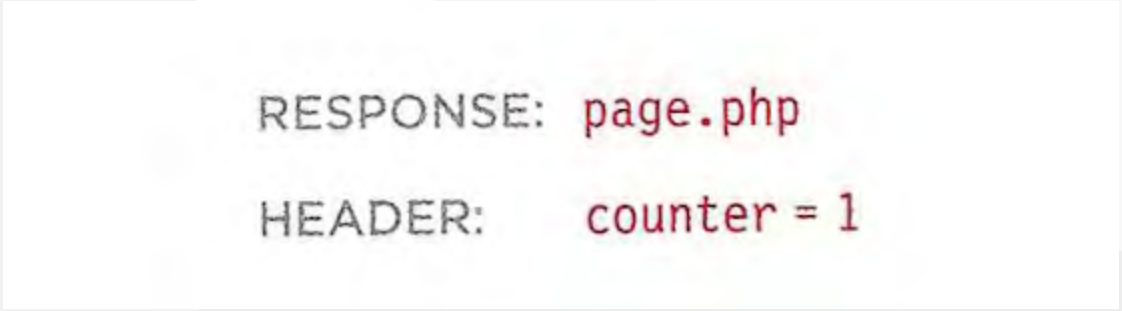
- El navegador solicita una página utilizando una petición HTTP



Cookies

Primera solicitud de la página

- El servidor devuelve la página solicitada
- Añade una cabecera HTTP que indica al navegador el nombre de la cookie que debe crear y su valor.



Cookies

Primera solicitud de la página

- El navegador muestra la página
- Crea una cookie utilizando los datos de la cabecera HTTP

Cookies

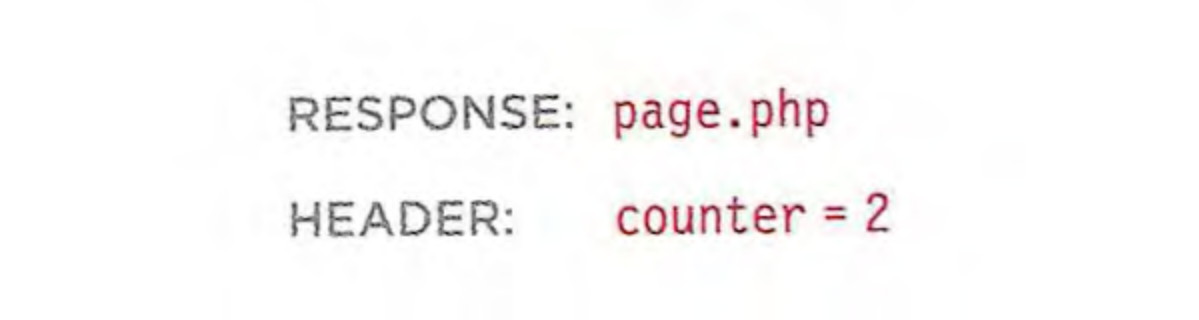
Peticiones de página posteriores

- El navegador solicita una página utilizando una petición HTTP
- Envía la cabecera HTTP con el nombre y el valor de la cookie

Cookies

Peticiones de página posteriores

- El servidor añade los datos de la cookie a `$_COOKIE`
- Crea la página utilizando los datos de `$_COOKIE`
- Devuelve la página solicitada
- Puede actualizar el valor almacenado en la cookie



Cookies

Peticiones de página posteriores

- El navegador muestra la página creada utilizando los datos de su cookie
- Actualiza la cookie utilizando los datos de la cabecera HTTP

Cookies

Una cookie **no debe utilizarse para almacenar datos confidenciales** (por ejemplo, direcciones de correo electrónico o números de tarjetas de crédito) porque el contenido de la cookie puede verse en las herramientas de desarrollo de un navegador y se envía entre un navegador y un servidor como texto sin formato.

Para evitar que alguien lea las cabeceras HTTP cuando se envían entre el navegador y el servidor hay que ejecutar nuestro sitio utilizando **HTTPS** en lugar de HTTP (ver *introducción del bloque B*). Esto **encripta la información de las cabeceras**.

Cómo crear y acceder a cookies

La función incorporada `setcookie()` de PHP se utiliza para **crear una cookie**. Para acceder a las cookies se puede utilizar el array superglobal `$_COOKIE`, o las funciones `filter_input()` y `filter_input_array()`.

Cómo crear y acceder a cookies

La función `setcookie()` de PHP **crea una cabecera HTTP que se envía con la página web y le indica al navegador que cree una cookie**. La función permite establecer un **nombre** y un **valor** para la cookie.

Cómo crear y acceder a cookies

Como esta función crea una cabecera HTTP, debe utilizarse **antes de enviar el contenido al navegador** (como se mostró con la función `header()` en la unidad 5). Incluso un espacio antes de la etiqueta `<?php` de apertura se trata como contenido.

Si no se establece una fecha de caducidad para una cookie, el navegador deja de enviar los datos de la cookie al servidor **cuando el usuario cierra su navegador**. Más adelante veremos cómo establecer una fecha de caducidad para una cookie.

```php
setcookie($name, $value);
```

Cómo crear y acceder a cookies

Una vez que el navegador ha almacenado la cookie, si ese navegador solicita otra página del sitio, **el nombre y el valor de la cookie se envían al servidor con la solicitud**. Cuando el intérprete de PHP recibe la petición, **añade los datos de la cookie a un array superglobal** llamado `$_COOKIE`.

Cómo crear y acceder a cookies

Se añade un nuevo elemento al array por cada cookie. Para cada elemento:

- La **clave** es el nombre de la cookie
- El **valor** es el dato que contiene la cookie (se almacena como una cadena)

Estos datos **suelen recogerse y almacenarse en una variable**.

Cómo crear y acceder a cookies

Si el código intenta acceder a una clave que no existe en `$_COOKIE` se genera un error. Para evitar esto, se puede utilizar el **operador de coalescencia nula** para comprobar si la clave está en el array. Si lo está, el valor de la cookie se almacena en la variable. Si no, la variable almacena el valor *null*.

```php
$preference = $_COOKIE['name'] ?? null;
```

Cómo crear y acceder a cookies

Las funciones `filter_input()` y `filter_input_array()` de PHP (ver unidad 6) también pueden recopilar datos de cookies. El tipo de entrada debe establecerse como *INPUT_COOKIE*.

El segundo parámetro es el **nombre de la cookie**. El tercer y cuarto parámetro son opcionales; especifican el ID del filtro a utilizar y cualquier opción para el filtro.

Si no se ha enviado una cookie, la función no genera ningún error. Además, si se utilizan filtros de tipo entero, flotante o booleano, convierten el valor a ese tipo de datos.

```
$preference = filter_input(INPUT_COOKIE, $name[, $filter[, $options]]);
```

Ejemplo: Establecer y acceder a cookies

Este ejemplo utiliza una cookie para contar el número de páginas que ha visto un visitante.

1. La variable `$counter` almacena el número de páginas que ha visto el visitante. Si el navegador envía datos de una cookie llamada counter al servidor, `$counter` almacenará ese valor. Si no, se utiliza el operador null-coalescing para almacenar un valor de 0 en su lugar.
2. Se añade 1 al valor en `$counter`, ya que el visitante acaba de ver una página.
3. La función `setcookie()` se utiliza para indicar al navegador que cree o actualice una cookie llamada *counter*, y almacene el valor de `$counter` en esa cookie.

Ejemplo: Establecer y acceder a cookies

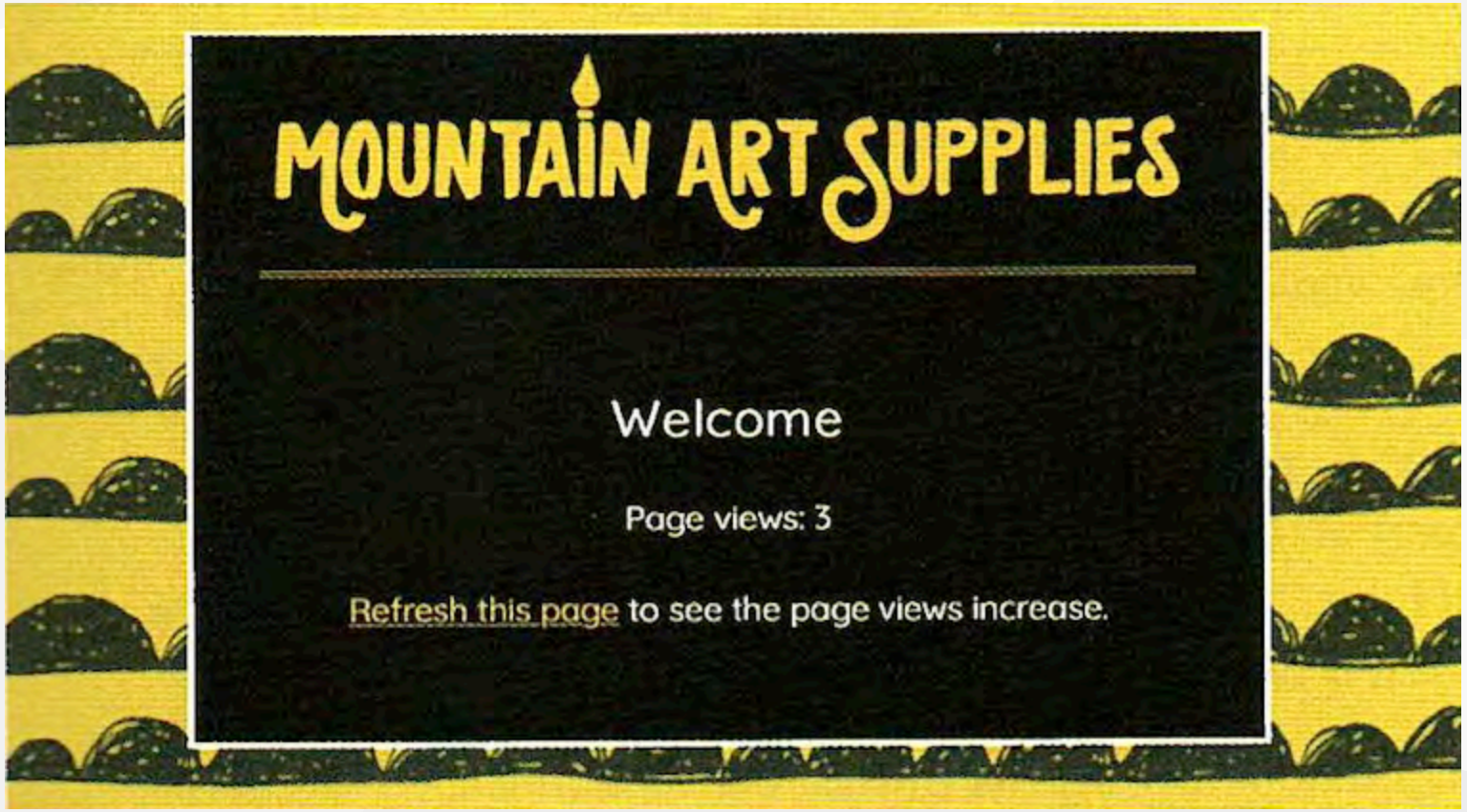
4. La variable `$message` almacena un mensaje que dice el número de páginas que ha visto el visitante.
5. Se muestra el mensaje.

> PRUEBA: Una vez que haya visto la página una vez, actualiza la página y observa cómo sube el contador.
> PRUEBA: Almacena tu nombre en una cookie llamada nombre y muéstrala después de las visitas a la página.

Ejemplo: Establecer y acceder a cookies

```php
<?php
① $counter = $_COOKIE['counter'] ?? 0;   // Get data
② $counter = $counter + 1;               // +1 to counter
③ setcookie('counter', $counter);        // Update cookie

④ $message = 'Page views: ' . $counter; // Message
?>
<?php include 'includes/header.php'; ?>

<h1>Welcome</h1>
⑤ <p><?= $message ?></p>
<p><a href="sessions.php">Refresh this page</a> to see
the page views increase.</p>

<?php include 'includes/footer.php'; ?>
```

Ejemplo: Establecer y acceder a cookies



Actividad: Establecer y acceder a cookies

Imagina que trabajas en un sitio web donde se quiere personalizar la bienvenida de los usuarios, recordando su nombre durante la sesión del navegador actual. Cada vez que el usuario visite la página durante esa sesión, verá un mensaje de bienvenida personalizado.

Actividad: Establecer y acceder a cookies

Instrucciones

1. **Crea un formulario** en `welcome.php` que pida el nombre del usuario si no se ha almacenado previamente en una cookie. Si el nombre ya está en la cookie, dale la bienvenida automáticamente.
2. **Almacena el nombre en una cookie** al enviar el formulario sin definir un tiempo de expiración, para que la cookie se elimine cuando el usuario cierre el navegador.

Actividad: Establecer y acceder a cookies

3. **Muestra un mensaje de bienvenida** en función de la cookie:
    - Si el usuario ya tiene una cookie de nombre, muestra: "Bienvenido de nuevo, [nombre]".
    - Si el usuario es nuevo (sin cookie), muestra un formulario que pida su nombre.

Controlando la configuración de las Cookies

`setcookie()` tiene parámetros que controlan cómo utilizan las cookies los navegadores. También es conveniente validar los datos recibidos de las cookies y utilizar `htmlspecialchars()` si su contenido se muestra en una página.

Controlando la configuración de las Cookies

- Para **actualizar un valor almacenado en una cookie**, hay que llamar de nuevo a `setcookie()` con un nuevo valor para la cookie.
- Para que el navegador **deje de enviar una cookie**, se debe volver a llamar a `setcookie()` estableciendo el valor a una cadena en blanco y la caducidad a un tiempo en el pasado.

Controlando la configuración de las Cookies

Siempre que actualicemos el valor o el tiempo de expiración de una cookie, los últimos cuatro argumentos deben utilizar los mismos valores que se utilizaron cuando se creó la cookie, lo que asegura que la cookie sea tratada de manera coherente en el navegador (si no se hace así, el navegador podría interpretar que estás trabajando con una cookie distinta, en lugar de actualizar o eliminar la existente).

```
setcookie($name[, $value, $expire, $path, $domain, $secure, $httponly])
```

Controlando la configuración de las Cookies

Además, debido a que es posible enviar cabeceras HTTP que imiten cookies con una petición de página:

- El servidor debería **validar los datos de las cookies antes de utilizarlos** (utilizando las técnicas estudiadas en la unidad 6).
- Si se muestra un valor de cookie en una página, se debería utilizar `htmlspecialchars()` para prevenir un ataque XSS.

Controlando la configuración de las Cookies

PARAMETER	DESCRIPTION
$name	The name of the cookie.
$value	The value the cookie should hold (this gets treated as a string - cookies do not store data types).
$expire	The date and time the browser should stop sending the cookie to the server (as a Unix timestamp). To set the timestamp, use PHP's `time()` function and add the period you want the cookie to last for.
$path	If a cookie is only needed for part of the site, specify the directories it should be used for. By default, the path is the root folder '/' which means all directories. Setting this to `/members` means it is only sent to pages in the `members` folder of the site.
$domain	If the cookie is only needed on a subdomain, set the URL for the subdomain. By default, it is sent to all subdomains of a site. If it is set to the subdomain `members.example.org`, the cookie is only sent to files in the subdomain `members.example.org`.
$secure	If this is given a value of `true`, the cookie will be created in the browser, but the browser only sends it back to the server if the page is requested using a secure HTTPS connection (see p184).
$httponly	If given a value of `true`, the cookie is only sent to the server (it cannot be accessed by JavaScript).

PERIOD	NOW	SECS	MINS	HRS	DAYS
1 day	`time() +`	`60 *`	`60 *`	`24`	
30 days	`time() +`	`60 *`	`60 *`	`24 *`	`30`

Ejemplo: Controlando la configuración de las Cookies

Este ejemplo **permite al usuario seleccionar un esquema de color** ("dark" o "light") **y guarda la preferencia en una cookie**; al recargar la página, se aplica el esquema de color almacenado, o el esquema "dark" por defecto si no se ha definido ninguna preferencia válida.

Ejemplo: Controlando la configuración de las Cookies

1. La variable `$color` almacena el valor enviado para una cookie llamada color (o null si no se envió).
2. Un array almacena las opciones permitidas para la combinación de colores.
3. Una sentencia if comprueba si el formulario ha sido enviado.
4. Si lo ha sido, el valor de la caja de selección llamada color se almacena en la variable `$color`. Esto sobrescribe el valor del paso 1.

Ejemplo: Controlando la configuración de las Cookies

5. La función `setcookie()` es llamada para establecer una cookie llamada color. Su valor es la opción que el usuario seleccionó de la caja de selección. También:

- Caduca en una hora
- Se envía a todas las páginas del sitio
- Se envía a través de HTTP o HTTPS
- Se oculta a partir de JavaScript

6. La condición de un operador ternario comprueba si el valor en `$color` está en el array `$options`. Si lo está, el valor se guarda en la variable llamada `$scheme`. Si no lo está, `$scheme` guarda el valor dark.
7. Se incluye una nueva cabecera. Escribe el valor en la variable `$color` en el atributo class de la etiqueta `<body>` para asegurar que las reglas CSS de la página utilizan el esquema de color correcto.

Ejemplo: Controlando la configuración de las Cookies

cookie-preferences.php

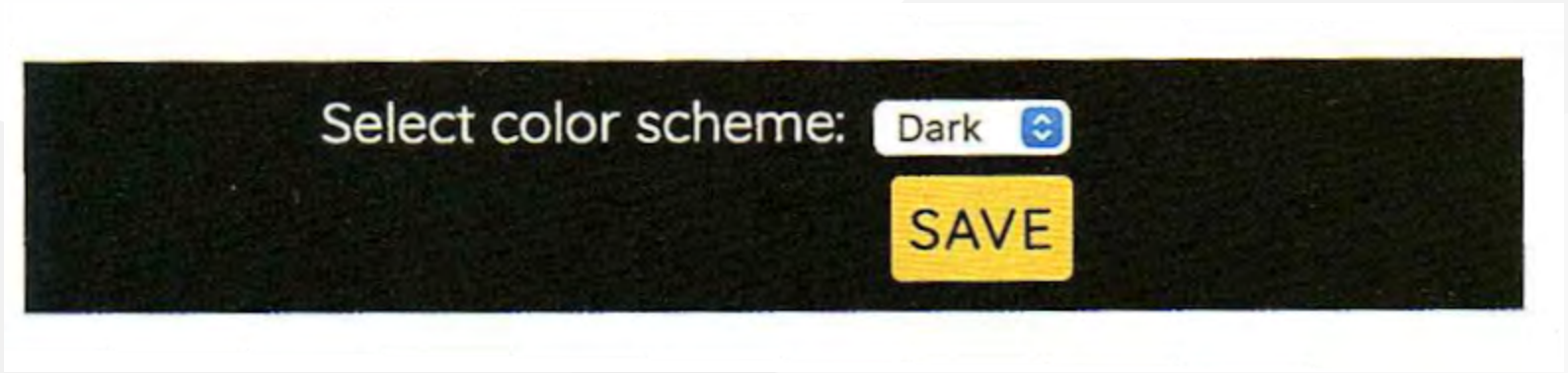
```
<?php
① $color   = $_COOKIE['color'] ?? null;        // Get data
② $options = ['light', 'dark',];                // Options

③ if ($_SERVER['REQUEST_METHOD'] == 'POST') { // If posted
④     $color = $_POST['color'];                 // Get color
⑤     setcookie('color', $color, time() + 60 * 60,
          '/', '', false, true);                // Set cookie
  }

  // If color is valid option, use it - otherwise use dark.
⑥ $scheme = (in_array($color, $options)) ? $color : 'dark';
  ?>
⑦ <?php include 'includes/header-style-switcher.php'; ?>
    <form method="POST" action="cookie-preferences.php">
      Select color scheme:
      <select name="color">
        <option value="dark">Dark</option>
        <option value="light">Light</option>
      </select><br>
      <input type="submit" value="Save">
    </form>
  <?php include 'includes/footer.php'; ?>
```

Ejemplo: Controlando la configuración de las Cookies

includes/header-style-switcher.php

```
<body class="<?= htmlspecialchars($scheme) ?>">
```

Ejemplo: Controlando la configuración de las Cookies



Actividad: Controlando la configuración de las Cookies

Imagina que trabajas en un proyecto de un sitio web multilingüe y quieres que la página recuerde la preferencia de idioma del usuario cada vez que regrese. De esta manera, al visitar el sitio, los usuarios no necesitan seleccionar su idioma cada vez.

Actividad: Controlando la configuración de las Cookies

Instrucciones

1. **Crear un formulario de selección de idioma** en `language-preferences.php` que permita al usuario seleccionar su idioma preferido de entre dos opciones: Inglés y Español.
2. **Almacenar la preferencia en una cookie** al enviar el formulario, con una duración de 30 días.
3. **Mostrar contenido en el idioma seleccionado**:
    - Si existe la cookie de idioma, muestra un mensaje de bienvenida en ese idioma.
    - Si no existe la cookie, muestra el formulario de selección de idioma.