

3. Cookies

- **Cookies:** pequeños archivos de texto enviados desde el servidor que se almacenan en el lado del cliente. Es decir, en el navegador.
- Permiten guardar información de forma persistente, de manera que se mantenga entre una petición al servidor y otra. Una cookie puede estar viva durante minutos, horas, días o incluso indefinidamente.
- En PHP
 - `setcookie()`
 - `$_COOKIE['nombreCookie']`

3. Cookies

- **SETCOOKIE()**. Esta función define una *cookie* que se enviará al cliente junto con el resto de las cabeceras de HTTP. Devuelve *true* si la cookie se envía con éxito o *false* en caso contrario.
- *Sintaxis:*
 - `bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]])`

3. Cookies

- **SETCOOKIE()**. Parámetros:
 - *name*: El nombre de la cookie. Este es el único obligatorio.
 - *value*: El valor de la cookie.
 - *expire*: El tiempo que la cookie tardará en expirar. Se trata de una fecha expresada en formato Unix.
 - *path*: La ruta del servidor para la que la cookie estará disponible. Si se utiliza '/', la cookie estará disponible en la totalidad del dominio.
 - *domain*: El dominio para el cual la cookie está disponible.
 - *secure*: Si la cookie solo debería enviarse en caso de conexión https, pon este argument a true.
 - *httponly*: Esta cookie solo será accesible a través de http.

3. Cookies

- **SETCOOKIE()**. Ejemplos:

- `<?php`

- ```
$value = "Alex";
setcookie("nombre", $value);
setcookie("nombre", $value, time()+3600); // la cookie expira en una hora
setcookie("nombre", $value, time()+3600, "/quotes/", "bestquotes.com", 1); ?>
```

Para obtener el valor:

Siempre hay que acceder a `$_COOKIES`:

- `<?php`

- ```
echo $_COOKIE['nombre']; ?>
```

3. Cookies

- **Borrar cookies.** Para eliminar una cookie hay que establecer una fecha de expiración anterior a la fecha actual
 - ```
<?php
 setcookie("nombre", "", time()-3600);
?>
```

## 4. Sesiones

- Son un mecanismo para que un script almacene variables (llamadas variables de sesión) en el servidor de manera persistente, de modo que posteriores ejecuciones de programas en el servidor solicitadas desde el mismo cliente pueden acceder a esas variables.
- Se comportan como si fueran variables globales a toda la aplicación web.
- Normalmente se usan para mantener el ID o nombre de usuario logueado, el estado de una aplicación...etc.
- Antes de acceder a cualquier variable de sesión (ya sea para crearla, para modificarla o para eliminarla) necesitamos indicarle a PHP que queremos usar variables de sesión en ese programa. (`sesión_start()`)

# PHP

## 4. Sesiones. Usar variables de sesión

- Las variables de sesión se manipulan a través del array superglobal **`$_SESSION`**.
- Si necesitamos una variable de sesión llamada, por ejemplo, `nombre_usuario`:
  - `session_start();`  
`$_SESSION['nombre_usuario'] = "valor-de-la-variable";`
- Para eliminarlas usamos `unset()` o `sesión_destroy()`
  - `Unset` destruye cualquier variable → `unset($_SESSION['nombre_usu'])`.
    - `unset();` → destruye todas las variables de sesión.
  - `Session_destroy` elimina la información de la sesión actual:  
`sesión_destroy();`