Cookies

Introducción

Definición

Las cookies son datos guardados por los navegadores a instancias del servidor. Los sitios web las utilizan con propósitos diversos: identificación de usuarios, sistemas de personalización, publicidad, estudios de perfil, etc. De hecho, la mayor parte de los portales no funcionarían sin cookies, o al menos se verían obligados a implementar algún mecanismo sustituto, como por cookies uno de los recursos más utilizados en el desarrollo de aplicaciones Web.

Un ejemplo de ello es la reescritura de parámetros en la URL. Su simplicidad y su gran utilidad hacen de las cookies un elemento de información de tamaño reducido enviado por el servidor para que el navegador lo guarde de manera que se pueda recuperar su valor en cualquier momento. Una de las características del protocolo HTTP es que se trata de un protocolo "sin memoria". Esto significa que, siempre que el usuario sigue un enlace y la página se carga en el navegador, es como si se hiciera por primera vez, no existiendo ninguna información disponible con respecto a lo que se cargó o dejó de descargar anteriormente.

Las cookies nacen con la intención de proporcionar un mecanismo para almacenar en el ordenador de cada usuario datos con los que formar un hilo de comunicación entre las distintas páginas que participan en una aplicación web. Uno de los casos más típicos es el que se produce cuando se registra en un sitio web. Cuando el mismo usuario visita de nuevo esas páginas es reconocido de forma automática. Esto es así porque la primera vez que se identificó se creó una cookie en el ordenador local de dicho usuario con los datos necesarios para ser reconocido.

En las versiones actuales de los navegadores las cookies pueden ser activadas o desactivadas de manera voluntaria por el usuario. Otra de las opciones consiste en que el navegador pida permiso cada vez que se quiere establecer o eliminar una cookie.

Caducidad

Todas las cookies tienen un período de caducidad. Cuando este tiempo se cumple la cookie desaparece. Teniendo en cuenta este comportamiento cabe distinguir cuatro tipos de cookies:

Cookies de sesión: Desaparecen cuando se acaba la sesión, entendiendo por sesión el período de tiempo que un usuario está en un sitio web de manera continuada con la misma ventana del navegador.

Cookies permanentes: Si la fecha de caducidad de la cookie se corresponde con un futuro lejano, se puede decir que la cookie es permanente. Esto no es del todo cierto, como se ha observado, ya que los navegadores cuentan con un espacio limitado para almacenar cookies de manera que las más nuevas hacen desaparecer a las más viejas cuando dicho espacio está totalmente ocupado.

Cookies con fecha de caducidad: A veces los sitios web establecen cookies con una fecha de caducidad concreta. Por ejemplo, se puede utilizar una cookie para recordar a un cliente que existe una oferta. Esta cookie tendrá necesariamente la fecha de caducidad de la propia oferta, más allá de la cual no tienen ningún sentido que siga existiendo.

Cookies para borrar: Borrar una cookie existente es como volver a escribirla con una fecha de caducidad anterior a la fecha actual.

DOMINIOS Y SUBDOMINIOS

Toda cookie está vinculada con un dominio o subdominio. Además, existe una restricción impuesta por el propio estándar de manera que no es posible acceder a las cookies de otro dominio o subdominio que no sea el de la propia URL donde se encuentra la página web. Esto significa que una página web en la dirección www.educacionit.com, por ejemplo, sólo podrá acceder a las cookies vinculadas con los dominios www.educacionit.com y .educacionit.com. Como se verá más adelante, las distintas tecnologías, ya sean de cliente o de servidor, disponen de los mecanismos necesarios para implementar dicho vínculo.

Una cookie que es del dominio www.educacionit.com será accesible desde cualquier URL que sea del tipo de ordenadores intranet.educacionit.com, o incluso, www.educacionit. intranet.com.ar. La restricción se extiende desde el caso general hasta los casos particulares.

Características

- Es una forma de guardar información en una máquina cliente
- Se realiza en forma transparente para el usuario a través de un browser
- Se utiliza para personalizar una página según las preferencias de usuario
- Se utiliza también para mantener valores entre distintas páginas
- Dentro de PHP se manejan como variables, a través del arreglo \$HTTP_COOKIE_VARS[]

Limitaciones

Existe una limitación con respecto al número de cookies que el navegador puede almacenar, así como el tamaño que pueden tener estas:

- 300 cookies.
- 4 Kb por cookie.
- 20 cookies por dominio.
- En general cuando se exceden los límites, el navegador empieza a descartar aquellas cookies que menos se han utilizado recientemente.



Funciones de PHP para interactuar con Cookies

Construcción

- Se utiliza la función setcookie()
- La forma de setearla es setcookie(nombreDeCookie, valor)
- Ejemplo de uso:

```
setcookie(nombre, "Jose");
setcookie(apellido, "Perez");
```

Utilización

- Para obtener las cookies se utiliza el vector \$_COOKIE[]
- La forma de obtener una cookie es \$_COOKIE['nombreDeCookie']
- Ejemplo de uso:

```
echo "Bienvenido," . $_COOKIE['nombre'];
```

Expiración

- Por defecto, las cookies expiran con el cierre de la ventana del browser
- Se puede setear el tiempo de expiración de una cookie
- Para esto, se utiliza la función setcookie() con una parámetro adicional
- La forma de setear una cookie con tiempo de expiración es setcookie(nombreDeCookie, valor, tiempoDeExpiracion)
- En estos casos, se guarda un archivo de texto (en forma automática) en el "cache" del browser
- Ejemplo de uso (setea como tiempo de expiración un mes):

```
setcookie(nombre, "Jose", time() + 60*60*24*30);
```

Eliminación

- Para eliminar una cookie se utiliza nuevamente la función setcookie()
- La forma de eliminarla es setcookie(nombreDeCookie, "")
- Ejemplo de uso:

```
setcookie(nombre, "");
```

La función isset()

- Se utiliza para saber si una cookie está seteada
- La forma de utilizarla es isset(\$_COOKIE['nombreDeCookie'])
- Retorna verdadero en caso de estar seteada, en caso contrario retorna falso

• Ejemplo de uso:

```
if( isset($_COOKIE['nombre']) )
     echo "Está seteada!";
else
echo "No está seteada!";
```

Mails

Envío de e-Mails en PHP

Características

La función de correo de PHP es sumamente útil y necesaria en muchas aplicaciones web. En este capítulo aprenderás cómo enviar correo electrónico a través de la función mail().

Funciones de PHP para enviar e-Mails

Un e-mail puede ser enviado a través de un script PHP en forma increíblemente fácil, utilizando la función mail (). Su prototipo es:

boolean mail(string to, string subject, string message [, string addl_headers[, string addl_params]])

Ejemplo de uso:

mail("miAmigo@hotmail.com", "Este es el asunto", "Mensaje");

Parámetros de la función mail

Parámetro	Descripción
to	Requerido. Especifica el receptor del e-mail.
subject	Requerido. Especifica el asunto del e-mail. Nota: Este parámetro no puede contener caracteres de newline "\n"
message	Requerido. Define el mensaje a ser enviado. Cada línea debe estar separada con un newline LF (\n). Las líneas no pueden exceder los 70 caracteres.
headers	Opcional. Especifica cabeceras adicionales como From, Cc y Bcc. La cabecera debe estar separada con un newline CRLF (\r\n)

Ejemplo completo de envío de un mail

```
$para = "juan@hotmail.com, diego@hotmail.com";
$asunto = "Este es el asunto";
$mensaje = "Este es el mensaje";
$cabeceras = "To: Juan < juan@hotmail.com>, Diego < diego@hotmail.com>\r\n";
$cabeceras .= "From: Esteban < esteban@hotmail.com>\r\n";
$cabeceras .= "Cc: Otro < otro@hotmail.com>\r\n";
$cabeceras .= "Bcc: OtroMas < otromas@hotmail.com>\r\n";
mail($para, $asunto, $mensaje);
```