

9. Cookies

Um cookie é frequentemente usado para identificar um usuário. Um cookie é um pequeno arquivo que o servidor incorpora no computador do usuário. Cada vez que o mesmo computador solicita uma página com um navegador, ele também enviará o cookie. Com o PHP, você pode criar e recuperar valores de cookies.

Um cookie é criado com a função `setcookie()`. A sua sintaxe é:

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

Apenas o parâmetro de nome é necessário. Todos os outros parâmetros são opcionais. O exemplo mostrado no quadro 45 cria um cookie chamado "usuário" com o valor "Pedro". O cookie irá expirar após 30 dias ($86400 * 30$). O "/" significa que o cookie está disponível em todo o site (caso contrário, selecione o diretório de sua preferência).

Em seguida, recuperamos o valor do cookie "usuário" (usando a variável global `$_COOKIE`). Também usamos a função `isset()` para descobrir se o cookie está definido.

Quadro 45 – Exemplo de uso de cookie

```
<!DOCTYPE html>

<?php
    $cookie_name = "user";
    $cookie_value = "Pedro";
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 dia
?>

<html>
<body>

<?php
    if(!isset($_COOKIE[$cookie_name])) {
        echo "Cookie chamado " . $cookie_name . " não está definido!";
    } else {
        echo "Cookie " . $cookie_name . " está definido!<br>";
        echo "Valor : " . $_COOKIE[$cookie_name];
    }
}
```

```
}  
?>  
<p><strong>Note:</strong> Pode ser necessário recarregar a página para ver o valor do  
cookie.</p>  
</body>  
</html>
```

Fonte W3schools

Resumindo, os cookies são arquivos de texto simples, enviados pelo site ao navegador, na primeira vez que você o visita. Em seu próximo acesso, o navegador reenvia os dados ao site para que suas informações sejam configuradas de forma automática. É por isso que você não precisa digitar seu email e senha toda vez que entra no Facebook.

10. Sessões

Uma sessão é uma maneira de armazenar informações (em variáveis) a serem usadas em várias páginas. Ao contrário de um cookie, as informações não são armazenadas no computador do usuário. Sessões permite a criação de aplicações mais personalizadas e aumenta o apelo do seu web site.

Ao trabalhar com um aplicativo, você o abre, faz algumas alterações e, em seguida, fecha-o. Isso é muito parecido com uma sessão. O computador sabe quem você é. Ele sabe quando você inicia e quando você termina o aplicativo. Mas na internet há um problema: **o servidor web não sabe quem você é ou o que você faz, porque o endereço HTTP não mantém o estado.**

As variáveis de sessão resolvem esse problema armazenando informações do usuário para serem usadas em várias páginas (por exemplo, nome de usuário, cor favorita etc.). Por padrão, as variáveis de sessão duram até que o usuário feche o navegador. Mas caso você precise de um armazenamento permanente, o mais recomendável é a utilização de uma banco de dados.

Uma sessão é iniciada com a função `session_start()`. Variáveis de sessão são definidas com a variável global PHP: `$_SESSION`. O quadro 46 ilustra um exemplo de uso.

Quadro 46 – Exemplo de criação de variáveis de sessão

```
<?php
    session_start();

    $_SESSION['nomeContato'] = $_POST['nomeContato'];
    $_SESSION['email'] = $_POST['email'];
    $_SESSION['celular'] = $_POST['celular'];
    ...
    $email = $_SESSION['email'];
    $celular = $_SESSION['celular'];
?>
```

O exemplo do quadro 47 mostra uma página sendo “abortada” caso a variável de sessão email não tenha sido instanciada.

Quadro 47 – Exemplos de uso das variáveis de sessão

<pre>if (session_status()!==PHP_SESSION_ACTIVE) { session_start(); if (!isset(\$_SESSION['email'])) return; }</pre>	<pre>if (session_status()!==PHP_SESSION_ACTIVE) { session_start(); } if (trim(\$_SESSION['login'])=='') header("Location: index.html"); else { \$login= \$_SESSION['login']; }</pre>
---	--

A função `session_status()` retorna o status atual da sessão. Os possíveis valores de retorno são:

- **PHP_SESSION_DISABLED** se as sessões estiverem desabilitadas.
- **PHP_SESSION_NONE** se as sessões estiverem habilitadas, mas nenhuma existir.
- **PHP_SESSION_ACTIVE** se as sessões estiverem habilitadas, e uma existir.

A função `session_start()`, inicia uma nova sessão ou continua uma sessão existente, no caso do exemplo do quadro 47, a função prossegue na sessão.

Para remover ou destruir uma sessão, o PHP fornece as funções `session_unset()` e `session_destroy()` respectivamente.