



# ***COOKIES***

# COOKIES



*Cookies* são “persistências temporárias” feitas no **lado do usuário**, e sessões são persistências dependentes de *cookies*, mas realizadas no **lado do servidor**.



## Para que servem *Cookies* e *Sessões* .

O protocolo HTTP é *stateless*, ou seja, ele não mantém um estado/conexão.

Toda a interação que o seu cliente fizer com um servidor Web acarretará em uma nova requisição e resposta.



As requisições são independentes e possuem um **tempo de vida** (conexão, envio de mensagem, resposta, encerramento da conexão).

O servidor *Web* não é capaz de identificar se duas requisições vieram de um mesmo navegador, e o mesmo não faz nenhum gerenciamento em memória para que mensagens sejam compartilhadas entre requisições.

**É para suprir esta necessidade que entram os *cookies* e sessões.**

Através de **cookies** o servidor *Web* é capaz de trocar informações de estado com o navegador do usuário.



Desse modo, somos capazes de adicionar produtos a um carrinho de compras, sem perder estas informações ao mudar de página, sair do *website* ou até mesmo fechar o navegador.



MiniInTheBox.com

Os *cookies* não estarão disponíveis até o próximo carregamento da página a qual o *cookie* deverá estar visível.

Para testar se um *cookie* foi enviado com sucesso, verifique o *cookie* no próximo carregamento da página antes que ele expire.

O tempo para expirar é configurado via o parâmetro **expira**.

A função `setcookie()` define  
um *cookie* para ser  
enviado juntamente com o  
resto dos cabeçalhos  
HTTP.

# Uso da Superglobal \$\_COOKIE





# Superglobais

Superglobais são variáveis nativas que estão sempre disponíveis em todos escopos.

Várias variáveis pré-definidas no PHP são "**superglobais**", que significa que elas estão disponíveis em todos escopos para todo o *script*. Não há necessidade de fazer **global \$variable**; para acessá-lo dentro de funções ou métodos.

\$GLOBALS

\$\_SERVER

\$\_GET

\$\_POST

\$\_FILES

\$\_COOKIE

\$\_SESSION

\$\_REQUEST

\$\_ENV

# A função setcookie()

## Exemplo:

```
Setcookie('cookie_teste', 'Algum valor', time()+60);
```

**NAME:** O nome do *cookie*.

**VALUE:** O valor do *cookie*. Esse valor é guardado no computador do cliente; não guarde informação sensível. Supondo que o nome seja 'nomedocookie', o valor pode ser lido através de `$_COOKIE['nomedocookie']`

O tempo para o *cookie* expirar. Esse valor é uma timestamp Unix, portanto é o número de segundos. Em outras palavras, você provavelmente irá utilizar isso com a função `time()` mais o número de **segundos** que você quer que ele expire. O `time()+60*60*24*30` irá configurar o *cookie* para expirar em 30 dias. Se configurado para 0, ou omitido, o *cookie* irá expirar ao fim da sessão (quando o navegador fechar).

## PRIMEIRO EXERCÍCIO

- 1) Dê Start no Apache pelo Xampp.
- 2) Crie o arquivo de nome testecookie.php
- 3) Coloque-o na pasta htdocs.
- 4) No navegador digite localhost e localize o arquivo feito.



Observe o funcionamento mudando os parâmetros da função setcookie().

```
1 <!DOCTYPE html>
2 <meta charset="UTF-8">
3 <?php
4     if(isset($_COOKIE['cookie_teste'])) {
5         echo 'Voce já esteve aqui recentemente, ' . $_COOKIE['cookie_teste'];
6     }
7     else {
8         setcookie('cookie_teste', 'Fulano de Tal', time()+10);
9         echo '<p>Bem Vindo!</p> Este é o INICIO da sua visualização';
10        echo '<p> O cookie durará 10 segundos!</p>';
11    }
12
13 }
14 ?>
15
16 </html>
```

Note que foi colocado 10 segundos.

Note como utilizar a duração, por segundos.

```
1 <?php
2
3 // Cria um cookie chamado 'usuario' com o valor 'Fulano'
4 setcookie('usuario', 'Fulano');
5
6 // Cria o mesmo cookie acima só que irá durar três dias
7 setcookie('usuario', 'Fulano', (time() + (3 * 24 * 3600)));
8
9 // Cria o novo cookie para durar duas horas
10 setcookie('nome', 'Ciclano', (time() + (2 * 3600)));
```

[snippet.php](#) hosted with ♥ by [GitHub](#)

## SEGUNDO EXERCÍCIO

São dois arquivos: login.php

index.php

```
1 <?php
2 /* Conteúdo das variáveis */
3 $usuario = 'BobEsponja';
4 $senha_usuario = '12345';
5
6 if (isset($_POST['nome_usuario']) && isset($_POST['senha'])) {
7
8     if (($_POST['nome_usuario'] == $usuario) && ($_POST['senha'] == $senha_usuario)) {
9
10         /* Setando o tempo do cookie para 2 minutos*/
11         setcookie('nome_usuario', $_POST['nome_usuario'], time()+120);
12         setcookie('senha', md5($_POST['senha']), time()+120);
13
14         header('Location:index.php');
15
16     } else {
17         echo 'Nome de Usuário e/ou Senha Inválidas.';
18     }
19 } else {
20     echo 'Você deve preencher login e senha.';
21 }
22 ?>
23
24 <!DOCTYPE html>
25 <html lang="pt-br">
26 <meta charset="utf-8">
27 <head>
28 <title>Tela de Login</title>
29 </head>
30 <body>
31 <h2>Tela de Login </h2>
32 <form name="login" method="post" action="login.php">
33     Usuário: <input type="text" name="nome_usuario"><br>
34     Senha: <input type="password" name="senha"><br>
35     <input type="submit" name="submit" value="ACESSAR">
36 </form>
37 </body>
38 </html>
```

```
1 <?php
2
3 if (null == ($_COOKIE['nome_usuario']) && null == ($_COOKIE['senha'] ))
4 {
5     header('Location: login.php');
6 } else {
7     echo 'Bem Vindo de volta, ' . $_COOKIE['nome_usuario'];
8 }
9 ?>
```



`$_SESSION`

***SESSÕES***

O que são *sessions*? *Sessions* são arquivos temporários que guardam informações no servidor.

Para que servem? Tem a mesma funcionalidade do famoso *COOKIE*, mas a vantagem é que o computador do cliente não precisa estar habilitado para usá-la.

Para iniciarmos uma session, usamos a função **session\_start()**.

Para um bom funcionamento, ela não pode estar depois de nenhum OUTPUT de dados (echo, print, códigos HTML, etc.).

É recomendável que esteja na primeira linha do código. É também recomendável usar um nome para identificar a session.



Lembrando:

Cookie → lado usuário

Sessão → lado servidor



A sessão é individual  
para cada acesso.

É uma variável  
GLOBAL, Se eu quiser  
acessar os dados da  
sessão em outros  
arquivos PHP dentro do  
meu script, basta iniciar  
a sessão e exibir os  
dados.

```
<?php  
// Inicia a sessão  
session_start();  
// Mostra uma frase na tela  
echo 'Olá ' . $_SESSION['nome'];
```



```
1  <?php
2  // Inicia a sessão
3  session_start();
4
5  // Cria uma chave e valor para um item da sessão
6  $_SESSION['usuario'] = 'luiz';
7
8  // Mostra o valor
9  echo $_SESSION['usuario'];
10
11 // Apaga a chave
12 unset( $_SESSION['usuario'] );
13
14 // Tenta mostrar o valor
15 // Mas gera um erro, pois ela foi apagada
16 echo $_SESSION['usuario'];
```

```
1 <?php
2 /* Conteúdo das variáveis */
3 session_start('login');
4 $usuario = 'BobEsponja';
5 $senha_usuario = '12345';
6 $_SESSION['logado'] = '0'; // uso da variável session de nome logado, com valor 0, pois o usuário ainda não fez login
7
8 if (isset($_POST['nome_usuario']) && isset($_POST['senha'])) {
9
10     if (($_POST['nome_usuario'] == $usuario) && ($_POST['senha'] == $senha_usuario)) {
11         $_SESSION['logado'] = '1'; //a variável logado recebe o valor 1 pois o usuário inseriu login e senha corretas
12         // Setando o tempo do cookie para 2 minutos
13         setcookie('nome_usuario', $_POST['nome_usuario'], time()+120);
14         setcookie('senha', md5($_POST['senha']), time()+120);
15
16         header('Location:indexcookiesession.php');//chama outra página
17
18     } else {
19         echo 'Nome de Usuário e/ou Senha Inválidas.';//login e senha inválidos
20     }
21
22 } else {
23     echo 'Você deve preencher login e senha.';
24 }
25 ?>
26 <!DOCTYPE html>
27 <html lang="pt-br">
28     <meta charset="utf-8">
29     <head>
30         <title>Tela de Login</title>
31     </head>
32     <body>
33         <h2>Tela de Login </h2>
34         <form name="login" method="post" action="loginsession.php">
35             Usuário: <input type="text" name="nome_usuario"><br>
36             Senha: <input type="password" name="senha"><br>
37             <input type="submit" name="submit" value="ACESSAR">
38         </form>
39     </body>
40 </html>
```



```
1 <html lang="pt-br">
2   <meta charset="UTF-8">
3   <?php
4   session_start('login');
5   if (null == ($_COOKIE['nome_usuario']) && null == ($_COOKIE['senha']))//não permite carregar esta página diretamente sem login
6   {
7       header('Location: loginsession.php');//redirecionamento para página de login
8
9   }elseif('0'==($_SESSION['logado']))//não permite retorno pelo navegador
10  {
11      header('Location: loginsession.php');//redirecionamento para página de login
12  }
13  else{
14      //permissão concedida com login e session de valor 1 que veio da página que chamou
15      echo 'Bem Vindo de volta, ' . $_COOKIE['nome_usuario'];
16      echo '<br />Seu status logado é: ' . $_SESSION['logado'];//apenas mostrando conteúdo da variável de sessão
17      $_SESSION['logado']='0';//a sessão não permite refresh do navegador, comente esta linha, salve e observe o que acontece
18  }
19  ?>
20
21 </html>
```



## REFERÊNCIAS:

[http://php.net/manual/pt\\_BR/function.setcookie.php](http://php.net/manual/pt_BR/function.setcookie.php)

<http://klauslaube.com.br/2012/04/05/entendendo-os-cookies-e-sessoes.html>

<https://www.codigofonte.net>

<http://www.tutsup.com/2014/07/18/session-em-php/>

<http://blog.thiagobelem.net/trabalhando-com-cookies-no-php>