

COOKIES

COOKIES



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



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.

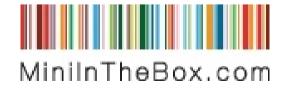
Através de *cookies* o servidor *Web* é capaz de <u>trocar informações de estado</u> com o navegador do usuário.

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.









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:

NAME: O nome do *cookie*.

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

guardado no computador do cliente; não guarde informação sensível. Supondo que o name seja 'nomedocookie', o valor pode ser

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 VALUE: O valor do cookie. Esse valor é quer que ele expire. O time() +60*60*24*30 irá configurar o cookie para expirar em 30 dias. lido través de \$ COOKIE['nomedocookie'] Se configurado para 0, omitido, o cookie irá expirar ao da sessão (quando

O tempo para o cookie expirar. Esse valor é uma timestamp

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>
   <meta charset="UTF-8">
   <?php
 3
    if(isset($ COOKIE['cookie teste'])){
       echo 'Voce já esteve aqui recentemente, '.$ COOKIE['cookie teste'];
 5
6
        }else{
 7
           setcookie('cookie teste', 'Fulano de Tal', time()+10);
8
           echo 'Bem Vindo! Este é o INICIO da sua visualização';
9
           echo ' 0 cookie durará 10 segundos!';
10
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</pre>
```

SEGUNDO EXERCÍCIO

</form> |</body> </html>

São dois arquivos: login.php

index.php

```
<?php
$usuario = 'BobEsponja';
$senha usuario = '12345';
if (isset($ POST['nome usuario']) && isset($ POST['senha'])) {
    if (($ POST['nome usuario'] == $usuario) && ($ POST['senha'] == $senha usuario)) {
            setcookie('nome usuario', $ POST['nome usuario'], time()+120);
            setcookie('senha', md5($ POST['senha']), time()+120);
        header('Location:index.php');
    } else {
        echo 'Nome de Usuário e/ou Senha Inválidas.';
                                                              <?php
 else {
    echo 'Você deve preencher login e senha.';
                                                              if (null == ($ COOKIE['nome usuario']) && null == ($ COOKIE['senha'] ))
                                                                      header('Location: login.php');
<!DOCTYPE html>
                                                                  } else {
<html lang="pt-br">
                                                                      echo 'Bem Vindo de volta, ' . $ COOKIE['nome usuario'];
<meta charset="utf-8">
<head>
<title>Tela de Login</title>
</head>
<body>
  <h2>Tela de Login </h2>
  <form name="login" method="post" action="login.php">
  Usuário: <input type="text" name="nome usuario"><br>
   Senha: <input type="password" name="senha"><br>
   <input type="submit" name="submit" value="ACESSAR">
```



SESSÕES

O que são *sessions*? *Sessions* são arquivos temporários que guardam informações <u>no servidor.</u>

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.

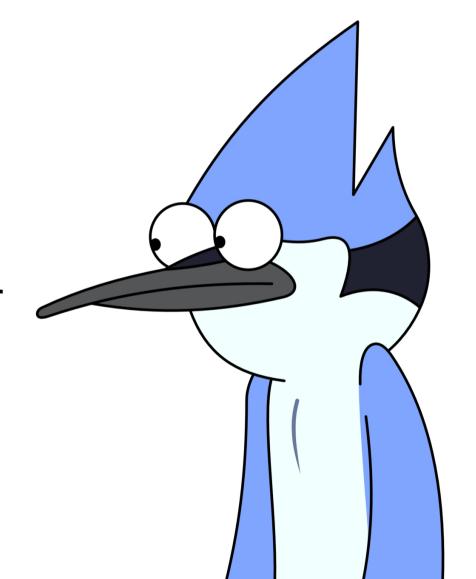
Para iniciarmos uma session, usamos a função session_start().

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'];</pre>
```



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

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

</html>

39 40 </body>

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



REFERÊNCIAS:

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