

[< VOLTAR](#)

# Criação e tratamento de Sessões

Neste tópico apresentaremos como criar, manipular e resgatar dados através de sessions.

NESTE TÓPICO



Marcar  
tópico



## Sessions

Assim como *cookies*, visto no Tópico XII – Criação e tratamento de *cookies*, as sessões também são utilizadas para salvar valores (\$variáveis) durante a visita de um usuário à sua aplicação PHP (CONVERSE, T. & PARK, J., 2005). Diferentemente dos *cookies* que armazenam seus valores no computador do usuário, as *sessions* são armazenadas no servidor, são mais seguras quando pretendemos trabalhar com informações sigilosas, como por exemplo *login* e senha, vale ainda ressaltar, que a sessão é destruída assim que o usuário fechar seu navegador (LOURENÇO, V.).

É interessante utilizar sessões (\$\_SESSION) quando você precisa passar dados de uma página para outra em sua aplicação, sem ter que utilizar \$\_GET, \$\_POST ou \$\_COOKIE. Você só precisa criar a sessão que deseja, com o valor necessário, e, automaticamente, o valor configurado aparecerá em todas as outras páginas ou arquivos do seu script PHP quando necessário.

Para começarmos a trabalhar com uma sessão, precisaremos antes de qualquer outra instrução iniciar a sessão, com a função *session\_start()*.

Vejamos a sintaxe para iniciar uma sessão em PHP:

```
1. <?php
2.     session_start();
3. ?>
```

Quando estivermos trabalhando com sessões, primeira linha de nossa aplicação, será *session\_start()*.

Uma vez iniciada a sessão, diferentemente dos cookies (onde existe limite de 4KB), não há limite de valores salvos nela. A sessão é pessoal de cada visitante. Quando um visitante acessa o site, é gerado um cookie no computador dele informando um id único de sessão e o PHP usa esse identificador para 'organizar' as sessões entre os visitantes do seu site. Mas esse cookie tem validade apenas enquanto o browser estiver aberto.

## Definindo valores para \$\_SESSION

Para definirmos os valores de uma variável de sessão, vamos trabalhar com a variável global `$_SESSION['variável']` (W3Schools). `$_SESSION` é um array associativo que recebe os valores que você quiser definir. Veja um exemplo o exemplo abaixo:

```
1. <?php
2.     session_start();
3.     $_SESSION['nome'] = "Tony Stark";
4. ?>
```

No exemplo acima estabelecemos um valor do tipo *string* para nossa variável de sessão, mas o valor atribuído, pode ser de qualquer tipo primitivo de dados: *string*, *integer*, *float*, *booleano*, etc.

Para recuperar e exibir o valor de uma sessão, podemos utilizar a seguinte sintaxe:

```
1. <?php
2.     session_start();
3. ?>
4. <p>Olá, <?php echo $_SESSION['nome']; ?>, seja bem vindo!! </p>
```

Como resultado, teremos no navegador:

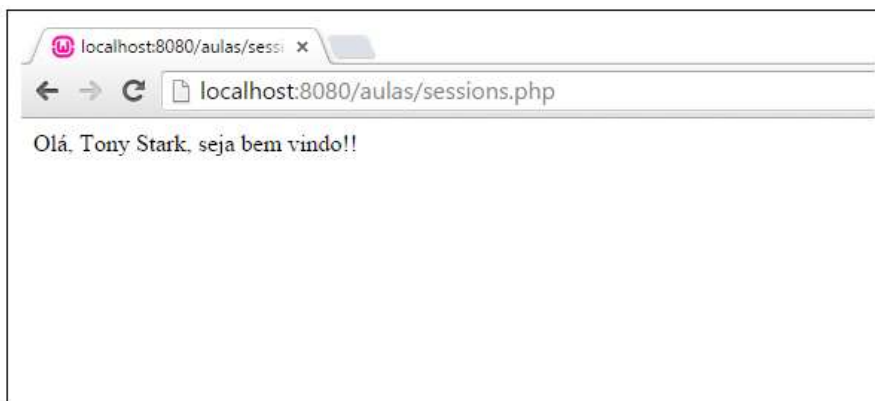


Figura 1: Exemplo de utilização de sessions.

Fonte: Fonte: Prof. Adriano Milanez.

As sessões são muito utilizadas para controlar o *login* de um usuário, o acesso a uma página restrita de nossa aplicação, pois são mais seguras, veremos como efetuar este controle, mais adiante, no Tópico XVI – Criando módulo de *login* e senha com sessões.

Já sabemos como criar e como manipular os valores de uma sessão, mas e se precisássemos apagar uma sessão específica? Poderíamos fazer utilizando a função *unset()*. Vejamos:

```
1. <?php
2.     session_start();
3.     unset($_SESSION['usuario']);
4. ?>
```

```
1. <?php
2.     session_start();
3.     unset($_SESSION['usuario']);
4. ?>
```

Neste exemplo teríamos apagado somente a sessão 'usuario', se tivéssemos alguma outra sessão criada, elas ainda, estariam existindo.

Duas possíveis possibilidades para se apagar todos os dados de uma sessão, seriam: a utilização de um simples *array* nulo para a *\$\_SESSION*, vejamos:

```
1. <?php
2.     session_start();
3.     $_SESSION = array();
4. ?>
```

Vale ressaltar que no exemplo acima a sessão ainda existe, porém vazia.

No segundo método destruiremos a sessão e neste caso, a sessão deixará de existir.

```
1. <?php
2.     session_start();
3.     session_destroy();
4. ?>
```

Destruir a sessão é sempre mais eficaz do que qualquer outro método.

Resumindo: As sessões são métodos de manter, ou preservar, determinadas informações a mantê-las ativas enquanto o navegador do cliente estiver aberto, ou enquanto a sessão não expirar, por *default*, uma sessão tem seu TTL, *time-to-live* ou tempo de vida de 180 minutos (SOARES, L.; AUGUSTO, B., 2007), porém, se precisarmos, por algum motivo modificar seu tempo de vida, utilizaremos a função *session\_cache\_expire(t)*, onde *t*, é o tempo em minutos, mais informações podemos pesquisar em

<<http://php.net/manual/en/function.session-cache-expire.php>  
(<http://php.net/manual/en/function.session-cache-expire.php>)> . Vejamos o exemplo abaixo:

```
1. <?php
2.     session_cache_expire(5);
3.     session_start();
4. ?>
```

No exemplo acima estamos modificando o TTL para 5 minutos.

Existem várias funções disponíveis para as sessões PHP para atender qualquer tipo de necessidade especial, todas estão disponíveis no manual do PHP, .

Espero que o conteúdo tenha sido proveitoso. Estudem e até o próximo tópico!

## Quiz

Exercício Final

Criação e tratamento de Sessões

INICIAR ➤

## Referências

BELEM, T. **Aprendendo a usar sessões no PHP**. Disponível em <<http://blog.thiagobelem.net/aprendendo-a-usar-sesoes-no-php>>, criado em 07/3/2009, acessado em 02/11/15 às 14h15min.

CONVERSE, T. & PARK, J. **PHP 5 - A Bíblia**. 2. Ed. Rio de Janeiro: Campus, 2005.

DALL'OGGIO, P. **Programando com orientação a objetos (Inclui *Design Patterns*)**. São Paulo: Novatec, 2009.

LOURENÇO, V. PHP: trabalhando com sessions. Disponível em <<http://blog.vilourenco.com.br/php-trabalhando-com-sessions/>>, acessado em 01/11/2015, às 15h20min.

MUTO, C. A. **PHP e MySQL: Guia Introductório**. 3. Ed. Rio de Janeiro: Brasport, 2006.

NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 2. Ed. São Paulo: Novatec, 2004.

PHP. **PHP: sessões**. Disponível em <[http://php.net/manual/pt\\_BR/book.session.php](http://php.net/manual/pt_BR/book.session.php)>. Acessado em 02/11/2015, às 13h45min.

SOARES, L.; AUGUSTO, B. **Aprendendo a Linguagem PHP**. 1. Ed. Rio de Janeiro: Ciência Moderna, 2007.

SOARES, W. **PHP 5 - Conceitos, Programação e Integração com Banco de Dados**. 4. Ed. São Paulo: Érica, 2004.

TANSLEY, D. **Como criar Web Pages rápidas e eficientes usando PHP e MySQL**. 1. Ed. Rio de Janeiro: Ciência Moderna, 2002.

W3Schools. PHP Functions. Disponível em: <[http://www.w3schools.com/php/php\\_functions.asp](http://www.w3schools.com/php/php_functions.asp)>, acessado em 26/10/2015, as 21h25min.



Avalie este tópico



ANTERIOR

Criação e tratamento de Cookies

Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Envio de emails e Upload de arquivos

© Todos os direitos reservados

Ajuda?

PRÓXIMO

([https://ava.un](https://ava.uninove.br/seu/AVA/topico/topico.php)

topico=)

