

SI401

Programação para a Web

2º Semestre - 2024

Programação *Back-End*: PHP

Unidade 18

Prof. Guilherme Palermo Coelho
guilherme@ft.unicamp.br

Roteiro

- *Cookies* e Sessões em PHP;
- Referências.

COOKIES E SESSÕES EM PHP

Cookies e Sessões em PHP

- Quando se acessa um *website* pelo protocolo HTTP, o servidor trata cada requisição a um documento (página) como um evento ***independente***;
 - HTTP não permite o armazenamento de informações de estado (**ex.:** um mesmo usuário acessando dois documentos diferentes);
- Neste contexto, ***cookies*** e ***sessões*** são muito úteis, pois permitem que se mantenha informações sobre os usuários enquanto eles navegam pelo site (mesmo em páginas diferentes);

Cookies e Sessões em PHP

- **Cookies:** são arquivos-texto que ficam armazenados no computador do usuário e podem ser recuperados posteriormente pelo servidor em que o site está hospedado;
- **Sessões:** são recursos que podem ser usados para manter uma conexão com o usuário enquanto ele estiver navegando pelo site:
 - São **variáveis** que armazenam informações sobre cada usuário enquanto ele estiver acessando as páginas de um site.

Cookies e Sessões em PHP

- Entre as utilidades de *cookies* e *sessões*, pode-se citar:
 - **Autenticação de usuários**: criação de um sistema que permita operações de *login* (*autenticação*) e *logout*, garantindo o acesso do conteúdo de um site apenas a usuários autorizados;
 - **Carrinho de compras**: usado por sites de comércio eletrônico;
 - **Exibição de anúncios e imagens**: para não exibir o mesmo anúncio ou imagem repetidamente;
 - **Personalização de páginas**: por exemplo, uma livraria poderia exibir um anúncio de um livro de culinária se o mesmo usuário tiver feito buscas sobre este tópico em visitas anteriores ao site.

Cookies

- São **arquivos-texto** armazenados no computador do cliente e recuperados posteriormente pelo servidor;
- Um *cookie* é formado por um par **nome/valor** (possui um *nome* pelo qual ele é referenciado e um *valor* a ele associado);
- *Cookies* podem ser utilizados em qualquer aplicação que:
 - Precise *compartilhar* dados entre **diferentes páginas**;
 - Precise *compartilhar* dados entre **diferentes acessos** (em dias e horários distintos);
- *Cookies* podem ser mantidos na máquina do usuário por vários dias;

Cookies

- Para **criar** e **excluir** cookies da máquina do usuário, PHP disponibiliza a função **setcookie()**;
- **Sintaxe:** `bool setcookie(string nome [, string valor [, int validade [, string caminho [, string domínio [, int seguro [, bool somente_http]]]]]]);`
 - **nome:** indica o nome do cookie sendo enviado ao cliente (é o único parâmetro obrigatório);
 - **valor:** é o valor do cookie;
 - **validade:** define o tempo de validade do cookie (em número de segundos a partir de 01/01/1970 – 0:00h). Caso não definido, expira quando o browser for fechado;
 - **caminho:** caminho do servidor para o qual o cookie estará disponível;

Cookies

- **domínio**: domínio para o qual o *cookie* estará disponível;
- **seguro**: indica se o *cookie* é seguro ('1' só permitirá a transmissão do *cookie* se a conexão for HTTPS);
- **somente_http**: se for TRUE, indica que o *cookie* só estará disponível via protocolo HTTP (não estará disponível para linguagens de *script*).

Cookies

- **Ex.:** criar um *cookie*
 - ***setcookie***("nome", "Guilherme");
- **Ex.:** excluir o cookie criado anteriormente (prazo de validade anterior ao tempo atual)
 - ***setcookie***("nome", "", time()-3600);
- **Ex.:** criar um cookie válido por 48h:
 - ***setcookie***("nome", "Guilherme", time()+172800);

Cookies

- **Atenção**: o envio de *cookies* para o cliente deve ser a ***primeira coisa*** na execução da página, ou seja, a função ***setcookie()*** deve ser utilizada **antes** de qualquer *tag* HTML;
- Para ***acessar*** um *cookie* enviado pelo navegador do cliente, PHP disponibiliza o *array superglobal* “***\$_COOKIE***”;
 - ***\$_COOKIE*** é um *array* com índices associativos;
 - O ***nome*** do *cookie* é o ***índice no array***;
 - **Ex.:** obter o valor do *cookie* “nome” criado no exemplo anterior:
 - ***\$valor = \$_COOKIE[“nome”];***

Cookies

```
<?php
    if (isset($_COOKIE["nome"]))
    {
        $message = "Bem vindo de volta, " . $_COOKIE["nome"];
        setcookie("nome", "", time()-3600); //apaga
    }
    else
    {
        $message = "Seja bem vindo! <br />";
        setcookie("nome", "Guilherme");
    }
    echo $message;
?>
```

```
// Primeira execução:
Seja bem vindo!
```

```
// Segunda execução:
Bem vindo de volta, Guilherme!
```

Sessões

- Uma **sessão** de PHP permite que sejam armazenados, no servidor, informações sobre o usuário que acessa um determinado website;
 - Estas **informações de sessão** ficarão armazenadas no servidor enquanto o usuário estiver navegando pelas páginas do site (enquanto a **sessão estiver aberta**);
- Cada sessão tem um **número identificador único** → “*session id*”;
 - Para uma página ter acesso aos dados da sessão, ela deve conhecer este identificador;
 - É necessário **transmitir** o *session id* entre páginas:
 - Geralmente é feito via *cookies* (automaticamente, se estiver habilitado na diretiva ***session.use_cookies*** do arquivo `php.ini`);

Sessões

- O primeiro passo para armazenar informações em uma sessão é ***iniciar uma sessão*** no servidor;
 - Isto pode ser feito através da função ***session_start()***;
- A função ***session_start()*** serve tanto para ***criar uma sessão*** quanto para ***restaurar os dados de uma sessão com base no identificador corrente (session ID)***;
 - **Sintaxe:** **`bool session_start();`**
 - **ATENÇÃO:** A chamada a esta função deve aparecer **antes** de qualquer ***tag*** HTML;

Sessões

- Após a criação de uma sessão, pode-se então **registrar variáveis** a esta sessão;
- Ao registrar uma variável a uma sessão, ela se torna disponível para todas as páginas que serão acessadas até o término da sessão;
- A maneira recomendada de **registrar variáveis** a uma sessão é através da **adição direta de entradas** ao array “\$_SESSION”;

```
<?php
    session_start();
    if (!isset($_SESSION['contador'])) {
        $_SESSION['contador'] = 1;
    } else {
        $_SESSION['contador']++;
    }
?>
```

Sessões

- Por fim, para ***apagar os dados de uma sessão***, pode-se adotar duas abordagens:
 - Utilizar a função ***unset()*** do PHP, que permite a eliminação de variáveis específicas:
 - `<?php unset($_SESSION['contador']); ?>`
 - Utilizar a função ***session_destroy()***, que encerra a sessão corrente e elimina ***todas*** as variáveis de sessão:
 - `<?php session_destroy(); ?>`
- Maiores informações sobre sessões em PHP podem ser acessadas em:
<http://www.php.net/manual/en/book.session.php>

Sessões

```
?php

    session_start();
    echo "Esta é a primeira página";
    $_SESSION['nome'] = "Guilherme";
    $_SESSION['sobrenome'] = "Coelho";
    $_SESSION['data'] = date('d/m/Y', time());
    echo '<br /><br /><a href="session2.php">Página 2</a>';

?>
```

Esta é a primeira página

[Página 2](#)

Sessões

?php

```
session_start();  
echo "Esta é a segunda página <br />";  
echo $_SESSION['nome'] . "<br />";  
echo $_SESSION['sobrenome'] . "<br />";  
echo $_SESSION['data'] . "<br />";  
echo '<br /><a href="session1.php?">Página 1</a>';  
session_destroy();
```

?>

Esta é a segunda página

Guilherme

Coelho

03/05/2012

[Página 1](#)

EXERCÍCIO SUGERIDO 4

Exercício Sugerido 4

- Crie um site que permita o cadastramento de veículos em um estacionamento. Este site deve conter um menu, à esquerda, que leve a duas páginas. Na primeira, deverá ser exibido um formulário com dois campos (placa e marca/modelo) e um botão de submissão. Quando os dados forem inseridos e submetidos, eles deverão ser cadastrados em uma base de dados em memória (um *array* ou outra estrutura que você queira usar). A segunda página deverá exibir todos os veículos cadastrados, em ordem alfabética de placa. Utilize os mecanismos de Sessão de PHP.

REFERÊNCIAS

Referências

[1] Niederauer, J. “Desenvolvendo Websites com PHP”, 2ª ed. Novatec, 2011.

[2] W3Schools PHP Tutorial:

<http://w3schools.com/php/default.asp>

[3] Manual do PHP:

http://php.net/manual/pt_BR/index.php

[4] Hudson, P. “Hacking with PHP”:

<http://www.hackingwithphp.com/>

[5] Prettyman, S. "Learn PHP 7", Apress, 2016:

<https://link.springer.com/book/10.1007/978-1-4842-1730-6>