



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

PHP – III

Prof. Moser Fagundes

Programação III
ADS

Sumário

- Funções
- Redirecionamento
- Cookies
- Sessions
- Exercícios

Funções

- Uma **função** é uma parte de um programa com um objetivo específico, encapsulado sob uma estrutura única que recebe um conjunto de parâmetros e retorna um dado ou faz uma operação.
- Uma função é declarada apenas uma vez, mas pode ser chamada diversas vezes.

Formato

Forma geral de uma função em PHP:

```
function nome_funcao(lista_argumentos) {  
    // comandos  
}
```

Exemplo

```
<?php
```

```
function quadrado($valor) {  
    return $valor * $valor;  
}
```

```
echo quadrado(4);
```

```
?>
```

Exemplo (2)

```
<?php
```

```
function multiplica($v1,$v2) {  
    return $v1 * $v2;  
}
```

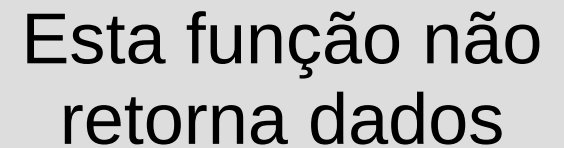
```
echo multiplica(4, 3.2);
```

```
?>
```

Função sem retorno

```
<?php
```

```
function escreve($frase) {  
    echo $frase;  
}
```



Esta função não
retorna dados

```
$retorno = escreve("Nada a declarar");
```

```
echo "<br>";
```

```
var_dump($retorno); // Escreve NULL
```

```
?>
```

Passagem de parâmetros por valor

O parâmetro recebido pela função é tratado como **variável local** dentro do contexto da função, não alterando seu valor externo.

Exemplo:

```
<?php
    function incrementa($variavel,$valor) {
        $variavel += $valor;
    }


    $a = 10;
    incrementa($a,20);
    echo $a;

?>
```


Passagem de parâmetros por referência

Basta usar o operador **&** na frente do parâmetro para que as transformações realizadas pela função sobre a variável sejam válidas no contexto externo à função.

Exemplo:



```
<?php
    function incrementa(&$variavel,$valor) {
        $variavel += $valor;
    }

    $a = 10;
    incrementa($a,20);
    echo $a;

?>
```

include

A instrução **include** inclui e avalia o arquivo informado. Esse arquivo (um código com variáveis, estruturas, comandos, etc.) é incluído no escopo do programa, tornando-se disponível a partir da linha que ocorre a inclusão. Se o arquivo não existir produzirá uma mensagem de advertência (**warning**), exceto se estiver desabilitado na configuração do PHP.

```
<?php
```

```
include "vars.php";  
include "fat.php";
```

```
echo fatorial($a). "<br>\n";  
echo fatorial($b);
```

```
?>
```

exemplo_slide_10.php / vars.php / fat.php

`include_once`

`include_once` funciona da mesma forma que o `include`, a não ser que o arquivo informado já tenha sido incluído, não refazendo a operação.

Torna-se útil em casos em que o programa pode passar pela mesma instrução mais de uma vez, evitando sobreposição de variáveis.

Redirecionamento com header

- O PHP fornece uma função chamada **header()** que pode ser usada para o redirecionamento de uma página para outra.

- **Exemplo:**

```
<?php
    header("Location:exemplo_slide_12B.php");

    // O código que vem após não será executado
?>
```

exemplo_slide_12.php

Cookies e sessões

- **Problema:** O protocolo HTTP não armazena informações de estado, ou seja, ele trata de forma independente cada requisição de página recebida.
- **Solução:** Usar cookies e sessões, que servem para manter informações sobre os usuários enquanto eles navegam no site.
 - **Cookies** são arquivos texto armazenados no computador do usuário, que podem ser recuperados posteriormente pelo servidor.
 - **Sessões:** são recursos que podemos usar para manter uma conexão com o usuário enquanto ele navega no site.

Cookies e sessões

- Algumas utilidades de cookies e sessões:
 - Manter usuários autenticados
 - Carrinhos de compras
 - Exibição de anúncios
 - Personalização de conteúdo e layouts

Cookies

Características de um cookie:

- Um cookie é formado por um par **nome/valor**: possui um nome pelo qual é referenciado e um valor associado.
- Permitem compartilhar dados entre diferentes páginas e diferentes acessos (dias e horários distintos).
- Cookies podem ser mantidos na máquina do usuário por vários dias, diferente de sessões!
- Um cookie é enviado junto com o header, logo você deve criar ele antes de gerar qualquer conteúdo da página.

Como criar cookies

setcookie(string nome [, string valor [, int validade]])

- **nome**: nome do cookie, único parâmetro obrigatório.
- **valor**: valor do cookie (se não for informado, o servidor tentará excluir o cookie).
- **validade**: tempo de validade (expresso em padrão UNIX – segundos). Se o valor for 0, ou se for omitido, o cookie irá expirar no final da sessão (quando o navegador fechar).
- A função **setcookie** retorna true se o cookie for criado com sucesso. Caso contrário, retorna false.

Exemplo com cookies

```
<?php
    setcookie("nome", "Lance Armstrong");

    // Descomente para apagar o cookie abaixo
    //setcookie("nome");

    // Coloca o tempo de 15 segundos
    // A funcao time() retorna o tempo em segundos
    setcookie("pais", "Spain", time()+15);

?>
<!DOCTYPE html>
<html>
    <head> ... </head>
    <body>
        <p>Os cookies foram criados.</p>
        <p><a href="exemplo_slide_18.php">Clique ... </a></p>
    </body>
</html>
```

Exemplo com cookies

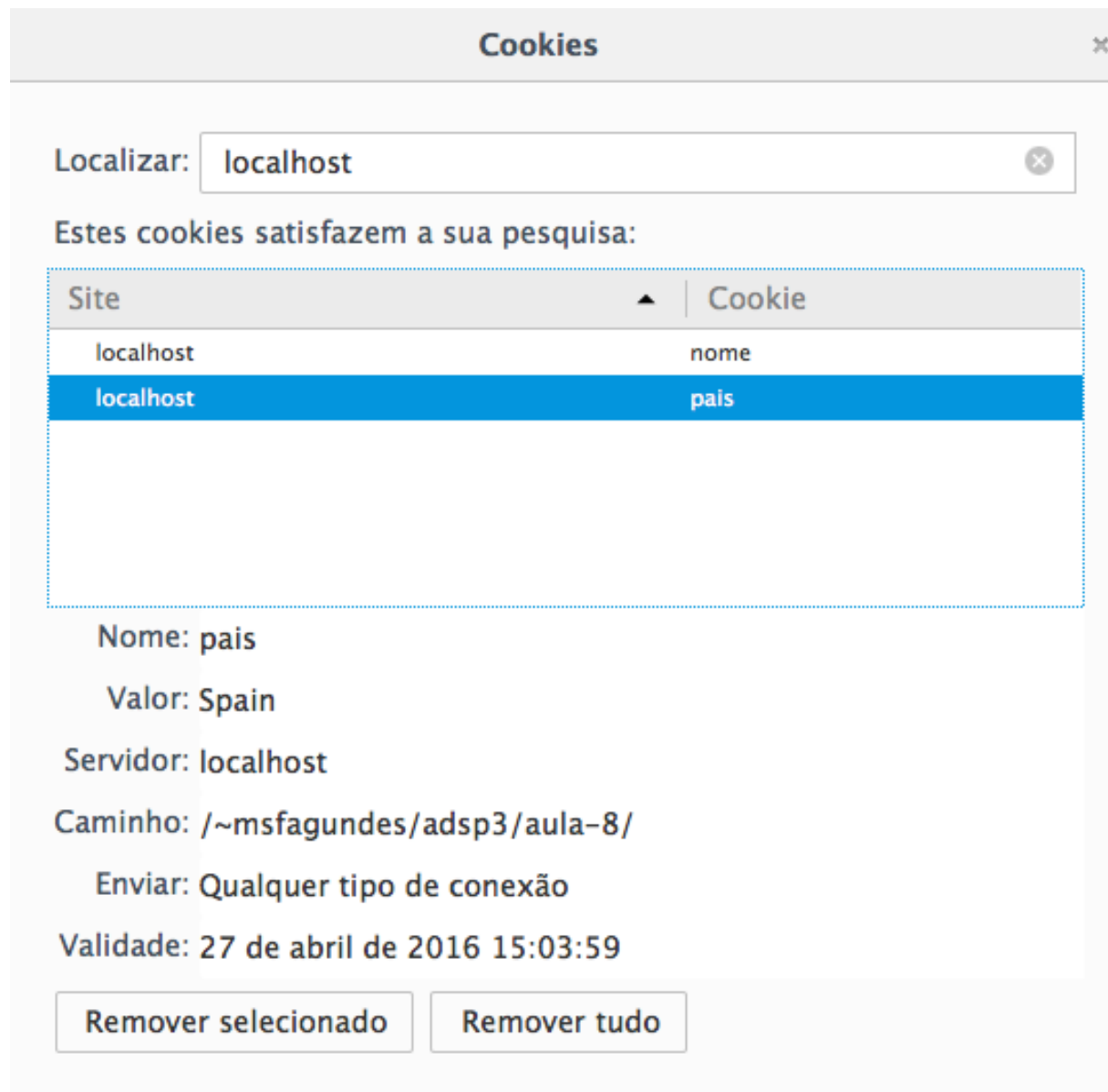
```
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body><p>

    <?php
      if($_COOKIE["nome"] == null) {
        echo "nome == null<br>"; }
      else {
        echo $_COOKIE["nome"]."<br>";
      }

      if($_COOKIE["pais"] == null) {
        echo "pais == null<br>";
      } else {
        echo $_COOKIE["pais"]."<br>";
      }
    ?>

  </p></body>
</html>
```

Cookies nas preferências do browser



Sessão

- Uma **sessão** é um **período** de tempo no qual o usuário navega em um site.
- Em uma sessão podemos registrar diversas variáveis, que ficarão gravadas em arquivo no servidor e poderão ser acessadas enquanto o site estiver aberto.

Criando uma sessão

Para criar uma sessão no PHP devemos chamar a função **`session_start()`**. Para criar variáveis na sessão devemos inserir valores no array **`$_SESSION`**. Neste exemplo, iniciamos a variável contador e incrementamos ela a cada atualização da página.

```
<?php
    session_start();

    if(!isset($_SESSION['contador'])) {
        $_SESSION['contador'] = 1;
    } else {
        $_SESSION['contador']++;
    }

    echo $_SESSION['contador'];
?>
```

Apagando variáveis de uma sessão

Se quisermos eliminar a variável `$_SESSION['contador']` podemos usar a função `unset($_SESSION['contador'])`.

```
<?php
    session_start();

    $_SESSION['contador'] = 1000;

    unset($_SESSION['contador']);

    if($_SESSION['contador'] == null) {
        echo "Contador nulo!";
    }
?>
```

Acessando variáveis de uma sessão

É possível acessar variáveis de sessão em diferentes páginas. Apenas lembre-se de chamar a função **`session_start()`** no começo da página antes de gerar qualquer conteúdo.

`exemplo_slide_23A.php`

`exemplo_slide_23B.php`

Destruindo de uma sessão

É possível destruir uma sessão e as suas variáveis. Para tal, chamamos a função **`session_destroy()`**.

Arrays em uma sessão

É possível incluir arrays em uma sessão.

Veja o exemplo no arquivo, o qual insere números aleatórios em um array da sessão.

`exemplo_slide_25.php`