Senac PR

Senac Umuarama

Ricardo Henrique Vieira Tomba

Explicações de funções

Umuarama

2025

**Superglobais:**  
  
 Como já explicado antes, as superglobais, são elas:

$\_SESSION – Armazena informações do usuário durante a sessão, permitindo a persistência de dados entre diferentes páginas enquanto o navegador estiver aberto ou até a sessão expirar.

Exemplo:

<?php

session\_start(); // Cria a sessão

$\_SESSION['nome'] = "Ricardo"; // Cria a variável "nome" e da o valor "Ricardo"

$\_SESSION['idade'] = 11; // Cria a variável "idade" e da o valor 11

echo "Nome = " . $\_SESSION['nome'] . "; Idade = " . $\_SESSION['idade']; // Exibe as variáveis

?>

$\_POST - Pega dados de formulários que foram enviados através do método POST, que gera um vetor associativo;

Exemplo:

<!-- funcao.html -->

<form action="funcao.php" method="post">

<!-- Form com metódo POST -->

<input type="text" name="nome" id="">

<button type="submit">Enviar</button>

</form>

<!-- index.php -->

<?php

$nome = $\_POST['nome']; // Puxa os dados da superglobal POST

echo $nome; // Imprime a var

?>

$\_COOKIE - são armazenados no navegador do usuário e enviados ao servidor em cada requisição. No PHP, esses dados ficam acessíveis por meio da superglobal $\_COOKIE.  
Exemplo:

<?php

// cria o cookie com nome e validade de tempo

setcookie("usuario", "Ricardo", time() + 360); // dura 1 hora

if (isset($\_COOKIE['usuario'])) { // efetua a verificação do cookie

echo "Olá, " . $\_COOKIE['usuario']; // imprime quando o cookie existe

} else {

echo "Cookie 'usuario' ainda não está definido."; // imprime quando o cookie nao existe

}

?>

**Armazenamento no navegador:**

LocalStorage (JavaScript) - O LocalStorage é o Armazenamento local do Cliente (Navegador), onde utilizando o JavaScript, é possível guardar variáveis no Client para ser acessada depois.

Ex.:

<script>

localStorage.setItem('nome', 'João');

</script>

**Funções importantes:**

isset() - Verifica se a variável tem algo dentro dela e se ela existe.

Ex.:

<?php

$var ; // cria a variável com algo ou sem algo

if(isset($var)){ // Verifica

echo "A variável está setada";

}else {

echo "A variável não está setada";

}

empty() - Verifica se há algo dentro da variável

Ex.:

<?php

$var = "Oi"; // cria a variável com algo ou sem algo

if(empty($var)){ // Verifica

echo "A variável não tem conteúdo dentro dela";

}else {

echo "A variável tem conteúdo dentro dela";

}

filter\_input() - Serve para pegar dados de entrada (ex.: formulários) de forma segura utilizando filtros, para validar e limpar esses dados, garantindo uma segurança extra.

Ex.:

<!-- index.html -->

<form action="validar.php" method="post">

<input type="email" name="email" id="">

<button type="submit">Enviar</button>

</form>

<!-- validar.php -->

<?php

$email = filter\_input(INPUT\_POST, 'email', FILTER\_VALIDATE\_EMAIL);

// INPUT\_POST é o metódo do formulário

// email é o nome do campo

// FILTER\_VALIDATE\_EMAIL é o filtro que será aplicado

if ($email === false) {

echo "email inválido";

} else {

echo "email válido: $email";

}

?>

filter\_var() - O filter var tem um conceito semelhante com o do filter\_input(), porém é aplicado a variáveis

Ex.:

<?php

$email = "[ricardo.9999@aluno.pr.senac.br](mailto:ricardo.4170@aluno.pr.senac.br)";

if (filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

echo "E-mail válido!";

} else {

echo "E-mail inválido!";

}

htmlspecialchars() - É outra função para previnir ataques de XSS, convertendo caracteres especiais em entidades no HTML, ex.: < vira $lt.

Ex.:

<?php

$nome = "<>\"&'";

echo htmlspecialchars($nome); // no código fonte dá página fica: &lt;&gt;&quot;&amp;&#039;

array() - Cria matrizes, que é uma estrutura de dados que armazena diversos valores em uma única variável.

Ex.:

<?php

$nomes = array("Ricardo", "Enzo", "Pedro");

echo $nomes[0] . "<br>"; // imprime Ricardo

echo $nomes[1] . "<br>"; // imprime Enzo

echo $nomes[2]; // imprime Pedro

Foreach - Estrutura de repetição usada para percorrer todos os elementos de uma matriz (array), acessando seus valores um por um.

Ex.:

<?php

$pessoa = array(

"nome" => "Ricardo",

"idade" => 912437

);

foreach ($pessoa as $chave => $valor) {

echo $chave . ": " . $valor . "<br>";

}

array\_push() - Adiciona um ou mais elementos no final de um array.  
Ex.:

<?php

$frutas = array("maçã", "banana");

array\_push($frutas, "laranja", "uva");

print\_r($frutas);

in\_array() - Verifica se um valor existe dentro de um array. Retorna true se encontrar, e false caso contrário.

Ex.:

<?php

$cores = array("vermelho", "azul", "verde");

if (in\_array("azul", $cores)) {

echo "A cor azul está no array.";

}

explode() - Divide uma string em partes, transformando em um array, com base em um separador.

Ex.:

<?php

$texto = "maçã,banana,laranja";

$frutas = explode(",", $texto);

print\_r($frutas);

implode() - Junta os elementos de um array em uma única string, usando um separador.

<?php

$frutas = array("maçã", "banana", "laranja");

$texto = implode(" - ", $frutas);

echo $texto;

strlen() - Conta o tanto de caracteres que tem na String  
Ex.:

<?php

$var = "Lorem ipsum dolor sit amet consectetur adipiscing elit.";

$stringlen = strlen($var); // calcula o tamanho da string

print $stringlen;

?>

str\_replace() - serve para substituir partes de uma string por outra

Ex.:

<?php

// ordem:

// "mundo" (a palavra que vai ser trocada);

// "universo" (a palavra que vai substituir a outra)

// "Ola, mundo!" (é a string que possui as palavras, onde a selecionada vai ser substituida)

echo str\_replace("mundo", "universo", "Ola, mundo!");

// saida: "Olá, universo!"

?>

substr() - É utilizada para obter uma parte de uma string, sendo uma substring, permitindo tirar caracteres a partir de uma posição específica da string original, e dá para definir quantos caracteres extrair

Ex.:

<?php

//substr básico

echo substr("Ola, mundo!", 4); // saida: "mundo!"

// pega o caractere da posição 4 e vai até o fim

// com comprimento

echo substr("Ola, mundo!", 4, 5); // saida: "mundo"

// pega no começo em 4 e pega 5 caracteres a diante

// valor negativo para inicio

echo substr("Ola, mundo!", -6); // saida: "mundo!"

// pega 6 caracteres antes do final

// valor negativo para o comprimento

echo substr("Ola, mundo!", 0, -1); // saida: "Ola, mundo"

// começa do inicio e vai até 1 caractere antes do final

?>

strpos() - Mostra a posição inicial em caracteres de uma frase ou palavra necessitada. (lembrando que a acentuação conta como caractere a mais)  
Ex.:

<?php

$frase = "Ola, mundo!"; // variavel com string

$posicao = strpos($frase, "mundo"); // procura na var "frase" a string mundo e guarda a onde está posição em caracteres

// se n tiver, retorna false

if ($posicao !== false) { // faz a verificação e compara se retornou false na variavel

echo "A palavra \"mundo\" está na posição: $posicao"; //imprime se existe e mostra a posição dela.

} else {

echo "a palavra \"mundo\" não encontrado na string."; // imprime se o resultado for false.

}

?>

**Importação de arquivos:**

Include - Importa um arquivo PHP externo, se o arquivo não for encontrado, gera um aviso (warning), mas o script continua executando.

Ex.:

<?php

include "menu.php";

echo "a";

Require - Também importa um arquivo PHP externo, porém, se o arquivo não for encontrado, gera um erro fatal e o script é interrompido.

Ex.:

<?php

require "menu.php";

echo "b";

include\_once - Funciona como include, mas garante que o arquivo seja incluído apenas uma vez, mesmo que o comando apareça várias vezes.

Ex.:

<?php

include\_once "funcoes.php";

include\_once "funcoes.php"; // esse vai ser ignorado

echo "c";

require\_once - Funciona como require, mas garante que o arquivo seja incluído só uma vez.

Ex.:

<?php

require\_once "conexao.php";

require\_once "conexao.php"; // nao sera carregado novamente

echo "d";

1. O que são variáveis superglobais? Cite outras além das já listadas.

R: As superglobais no PHP representam tipos de arrays distintos que podem ser acessados em qualquer ponto do seu código. Elas guardam informações transmitidas pelos usuários, pelos navegadores ou pelos servidores. Além de $\_GET, $\_POST, $\_SESSION e $\_COOKIE, encontramos também $\_FILES, $\_ENV, $\_SERVER e $GLOBALS. Cada uma é utilizada para um propósito particular dentro da lógica do sistema.

1. Qual a diferença entre isset() e empty()?

R: A diferença entre o isset() e o empty(), é que o isset verifica se a variável tem um conteúdo que existe dentro dela, e o empty verifica se tem pelo menos um caractere (caso da string) na variável.

1. Para que serve htmlspecialchars()?

R: Ele transforma caracteres especiais em códigos onde o navegador interpreta, ajudando na segurança do site, ex: < vira &lt;.

1. Explique a diferença entre explode() e implode() com exemplo.

R: A diferença é que o explode separa uma string e vira array, e o implode junta a um array e vira uma string

<?php

$texto = "maçã,banana,laranja";

$frutas = explode(",", $texto); // vira array

$lista = implode(" | ", $frutas); // junta array

echo $lista; // maçã | banana | laranja

1. O que acontece se usar require para incluir um arquivo que não existe? E se usar include?  
   R: Como já explicado antes, o require retorna erro e para de executar o código e o include apenas avisa o erro e continua
2. O que é LocalStorage e quando ele é mais útil que cookies ou sessões?

R: O localstorage guarda dados no navegador e não expira, apenas se o usuário apagar os dados, e é melhor quando você quer salvar coisas como a preferência de tema, idioma ou preferências de usuário

1. Em um sistema de login, quando você usaria $\_SESSION e quando usaria $\_COOKIE?

R: Os exemplos que são mais utilizados são: Usar o $\_SESSION para deixar o login aberto enquanto o navegador estiver aberto, e o $\_COOKIE para lembrar o login mesmo depois de o navegador estiver fechado.