Disciplina: PROGRAMAÇÃO BACK-END

Professor: Luciano Rocha

Curso Técnico de Desenvolvimento de Sistemas

Funções essenciais do PHP com exemplos — Guia para iniciantes

Este material apresenta as funções nativas mais usadas por programadores PHP no mundo, com explicações simples e exemplos prontos. As funções estão agrupadas por tema para facilitar a consulta.

1) Manipulação de Strings

strlen(string)

Retorna o tamanho (número de caracteres) da string.

```
$txt = "PHP é legal";
echo strlen($txt); // 11
?>
```

Dica: Para strings com acentos/UTF-8, prefira mb_strlen($txt, 'UTF-8').

strtoupper, strtolower, ucfirst, ucwords

Transformam o texto: tudo maiúsculo, tudo minúsculo, primeira letra da string, primeira letra de cada palavra.

```
echo strtoupper("php");      // PHP
echo strtolower("PHP");      // php
echo ucfirst("php é bom");   // Php é bom
echo ucwords("curso de php"); // Curso De Php
?>
```

Dica: Apenas transformam visualmente; não alteram a variável original a menos que você reatribua.

trim, ltrim, rtrim

Removem espaços (e quebras) no início/fim da string.

```
$nome = "  Ana  ";
echo trim($nome);  // "Ana"
?>
```

Dica: Útil para limpar entradas de formulário.

substr(string, inicio, [tamanho])

Extrai parte da string.

```
$txt = "Aprendendo PHP";
echo substr($txt, 0, 10); // Aprendendo
?>
```

str_replace(busca, troca, string) / str_ireplace (case-insensitive)

Substitui todas as ocorrências de um texto por outro.

```
$frase = "Eu gosto de PHP";
echo str_replace("PHP", "Java", $frase); // Eu gosto de Java
?>
```

strpos(string, procura) / stripos (case-insensitive)

Retorna a posição da primeira ocorrência; false se não encontrar.

```
$pos = strpos("Aprender PHP", "PHP");
if ($pos !== false) { echo "Achei na posição $pos"; }
```

```
?>
```

explode(delimitador, string) / implode(delimitador, array)

Converte string → array (explode) e array → string (implode).

```
$linha = "Ana,Bruno,Carlos";
$nomes = explode(",", $linha);  // ["Ana","Bruno","Carlos"]
echo implode(" - ", $nomes);    // Ana - Bruno - Carlos
?>
```

sprintf(formato, ...args)

Monta uma string formatada sem imprimir na tela (semelhante a printf).

```
$preco = 12.5;
$msg = sprintf("Preço: R$ %.2f", $preco);
echo $msg; // Preço: R$ 12.50
?>
```

Dica: Ótima para criar mensagens padronizadas antes de exibir ou salvar.

2) Arrays

count(array)

Retorna a quantidade de elementos no array.

```
$itens = [1,2,3];
echo count($itens); // 3
?>
```

in_array(valor, array) / array_key_exists(chave, array)

Verifica se um valor existe no array, ou se uma chave existe em um array associativo.

```
$cores = ["vermelho","azul"];
echo in_array("azul", $cores) ? "sim" : "não"; // sim

$aluno = ["nome"=>"Ana", "idade"=>20];
echo array_key_exists("nome", $aluno) ? "ok" : "x"; // ok
?>
```

array_push, array_pop, array_shift, array_unshift

Adicionam/removem elementos no fim/início do array.

```
$nums = [10,20];
array_push($nums, 30);  // [10,20,30]
$ultimo = array_pop($nums); // remove 30
array_unshift($nums, 5);    // [5,10,20]
$primeiro = array_shift($nums); // remove 5
?>
```

array_merge, array_slice, array_splice

Junta arrays (merge), recorta parte (slice), remove/insere em posição (splice).

```
$a = [1,2]; $b = [3,4];
$c = array_merge($a, $b); // [1,2,3,4]
$pedaco = array_slice($c, 1, 2); // [2,3]
array_splice($c, 2, 1, [99]); // [1,2,99,4]
?>
```

array_unique, array_reverse

Remove duplicados (unique) e inverte a ordem (reverse).

```
$val = [1,1,2,3,3];
print_r(array_unique($val)); // [1,2,3]
print_r(array_reverse($val)); // [3,3,2,1,1]
?>
```

sort/rsort (valores), asort/arsort (valores mantendo chaves), ksort/krsort (por chave)

Ordenam arrays de diferentes formas.

```
$precos = ["p1"=>10,"p2"=>7];
asort($precos); // ordena por valor: p2->7, p1->10
ksort($precos); // ordena por chave: p1, p2
?>
```

array_filter, array_map, array_reduce

Filtra, transforma e reduz arrays com callbacks.

```
$nums = [1,2,3,4];
$pares = array_filter($nums, fn($n)=>$n%2==0); // [2,4]
$dobro = array_map(fn($n)=>$n*2, $nums);       // [2,4,6,8]
$soma = array_reduce($nums, fn($acc,$n)=>$acc+$n, 0); // 10
?>
```

Dica: Essas funções são muito usadas em processamento de listas.

3) Números e Matemática

abs, round, ceil, floor

Valor absoluto, arredondamento, para cima, para baixo.

```
echo abs(-5);           // 5
echo round(10.456, 2);  // 10.46
echo ceil(10.1);        // 11
echo floor(10.9);       // 10
?>
```

max, min, number_format

Maior, menor e formatação amigável de números.

```
echo max(3,9,2);  // 9
echo min([5,2,8]); // 2
echo number_format(1234.5, 2, ',', '.'); // 1.234,50
?>
```

rand, mt_rand (não seguro), random_int (seguro)

Gera números aleatórios; prefira random_int para fins de segurança.

```
echo rand(1, 10);        // 1..10 (não seguro)
echo mt_rand(1, 10);     // melhor desempenho (não seguro)
echo random_int(1, 10);  // criptograficamente seguro
?>
```

4) Datas e Horas

date, time, strtotime, date_default_timezone_set

Formatação de datas, timestamp atual, conversão de texto em data e fuso horário.

```
date_default_timezone_set('America/Sao_Paulo');
echo date("d/m/Y H:i");  // 31/12/2025 23:59
$agora = time();         // timestamp
$prazo = strtotime("+7 days");
echo date("d/m/Y", $prazo);
?>
```

Dica: Para aplicações maiores, considere a classe DateTime/DateInterval.

5) Arquivos e Sistema

file_get_contents, file_put_contents

Leitura e escrita simples de arquivos inteiros.

```
file_put_contents("dados.txt", "linha 1\nlinha 2");
$txt = file_get_contents("dados.txt");
echo nl2br($txt);
?>
```

fopen, fread, fwrite, fclose

Controle manual de leitura/escrita em streams.

```
$fp = fopen("log.txt", "a");        // abre para anexar
fwrite($fp, date("H:i:s")." - acesso\n");
fclose($fp);
?>
```

file_exists, is_file, is_dir, mkdir, unlink

Verificações e operações básicas no sistema de arquivos.

```
if (!file_exists("uploads")) { mkdir("uploads"); }
if (is_file("velho.txt")) { unlink("velho.txt"); }
?>
```

6) Web/HTTP, Sessões e Cookies

header, http_response_code

Envia cabeçalhos HTTP e define o código de resposta.

```
// redirecionar
header("Location: /login.php");
http_response_code(302);
// Sempre chamar header() antes de qualquer saída (echo)
?>
```

Dica: Cabeçalhos devem ser enviados antes de qualquer HTML/echo.

setcookie, $_COOKIE

Define cookies no cliente e lê via superglobal.

```
setcookie("tema", "escuro", time()+3600, "/");
echo $_COOKIE["tema"] ?? "padrão";
?>
```

Dica: Cookies são enviados na próxima requisição após o setcookie.

session_start, $_SESSION, session_destroy

Inicia sessão, manipula dados de sessão e finaliza.

```
session_start();
$_SESSION["usuario"] = "ana";
echo $_SESSION["usuario"];
session_destroy(); // encerra a sessão
?>
```

Dica: Chame session_start() no topo do script, antes de HTML.

7) Segurança & Validação

password_hash, password_verify

Gera hash seguro de senha e verifica no login.

```
$hash = password_hash("minha_senha", PASSWORD_DEFAULT);
if (password_verify("minha_senha", $hash)) {
  echo "Senha ok";
}
?>
```

Dica: Nunca salve senhas em texto plano. Use password_hash.

filter_var (VALIDATE_EMAIL, VALIDATE_INT, VALIDATE_URL...)

Valida e filtra dados de entrada de forma simples.

```
$email = "ana@example.com";
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
  echo "Email válido";
}
?>
```

Dica: Evite filtros obsoletos. Consulte a documentação do PHP ao escolher filtros.

htmlspecialchars

Escapa caracteres especiais para evitar XSS ao exibir conteúdo do usuário.

```
$nome = '<b>Ana</b>';
echo htmlspecialchars($nome, ENT_QUOTES, 'UTF-8'); // exibe os < > como texto
?>
```

8) JSON e Expressões Regulares

json_encode, json_decode

Convertem dados entre PHP e JSON (texto).

```
$dados = ["nome"=>"Ana","idade"=>20];
$json = json_encode($dados);        // {"nome":"Ana","idade":20}
$refaz = json_decode($json, true);  // array associativo
?>
```

Dica: json_decode($json, true) retorna array; sem o true, retorna objeto.

preg_match, preg_replace

Casamento e substituição com expressões regulares (regex).

```
$txt = "Código: ABC-1234";
if (preg_match("/[A-Z]{3}-\d{4}/", $txt)) { echo "Formato ok"; }
echo preg_replace("/\s+/", " ", "um   texto   com   espaços");
?>
```

Dica: Regex é poderoso; teste suas expressões em ferramentas online.

9) Utilidades & Depuração

isset, empty, unset

Checa existência, se está vazio e destrói variáveis.

```
$valor = "0";
echo isset($valor) ? "existe" : "não";
echo empty($valor) ? "vazio" : "cheio"; // cuidado: "0" é empty
unset($valor);
?>
```

Dica: empty('0') é true — atenção ao validar strings numéricas.

print_r, var_dump, error_log

Exibem estrutura/valores para debug e gravam no log do servidor.

```
$arr = ["a"=>1,"b"=>2];
print_r($arr);
var_dump($arr);
error_log("Algo aconteceu aqui...");
?>
```

Observações: Algumas 'funções' citadas (ex.: include/require, echo) são construtos da linguagem, não funções. Este guia prioriza funções amplamente usadas no PHP moderno.