



Centro Paula Souza



FATEC PRESIDENTE PRUDENTE

DISCIPLINA: ELETIVA I

AULA: FUNÇÕES INTERNAS E FUNÇÕES DEFINIDAS PELO USUÁRIO

FUNÇÕES INTERNAS

- Também conhecidas como funções *built-in*
- O PHP vem por padrão com muitas funções e construtores.
- Lista de funções internas:
https://www.php.net/manual/pt_BR/indexes.functions.php

EXEMPLOS

Funções de Data e Hora

Para mostrar um campo Data e Hora o caminho mais prático é a utilização da função DATE

```
<?php  
  
    echo date("d/m/Y");  
  
?>
```

Funções de Data e Hora

Letras chaves para usar com o comando date	
d	Dia do mês, numérico (01...31)
j	Dia do mês, numérico (1...31)
w	Dia da semana (0=Domingo...6=Sábado)
D	Nome do dia da semana abreviado (Sun...Sat)
M	Nome do mês abreviado (Jan...Dec)
F	Nome do mês completo (January...December)
Y	Ano, numérico, 4 dígitos
y	Ano, numérico, 2 dígitos
H ou g	Hora (00...23)
h	Hora (01...12)
i	Minutos, numérico (00...59)
s	Segundos (00...59)
A	AM ou PM
a	am ou pm

Função checkdate()

Verifica a validade de uma data

```
<?php
    if (checkdate('12', '25', '2016'))
        echo 'Válido';
    else
        echo 'Inválido';
?>
```

Função strlen()

Obtém a quantidade de caracteres de uma string

```
<?php
```

```
$palavra = "Análise e Desenvolvimento de Sistemas";  
echo "A variável contém:".strlen($palavra) . "caracteres";
```

```
?>
```

Função substr()

Extraí partes de uma string

```
<?php
    $data = date("Y-m-d");
    $dia = substr($data,-2);
    $mês = substr($data,5,2);
    $ano = substr($data,0,4);
    $novadata = $dia . "/" . $mês . "/" . $ano;
    echo $novadata;
?>
```


Funções strtoupper() e strtolower()

strtoupper() converte todos os caracteres contidos em uma string em letras maiúsculas.

strtolower() converte todos os caracteres contidos em uma string em letras minúsculas.

```
<?php
    $A = "Olá pessoal! ";
    $B = strtoupper($A);
    $C = strtolower($A);
    echo $B . "<br>";
    echo $C;
?>
```

Função sort()

Ordena os dados de um vetor

```
<?php  
    $array = ["Vanessa", "Priscila", "Patricia"];  
    sort($array);  
?>
```

Função header()

Envia uma solicitação de alteração no HEADER HTTP

```
<?php  
    header('Location: http://www.example.com/');  
    exit;  
?>
```

Função isset()

Informa se uma variável foi iniciada

```
<?php
    $var = “;
    if (isset($var)) {
        echo "Essa variável existe.";
    }
?>
```

Função mail()

Envia um email

```
<?php  
    mail('caffinated@example.com', 'My Subject', 'Message');  
?>
```

Função number_format()

Formata um número decimal

```
<?php  
    $valor = 10001.67890;  
    number_format($valor,2,',','.');  
?>
```

FUNÇÕES DEFINIDAS PELO USUÁRIO

- A ideia básica de uma função, implementada em alguma linguagem de programação, é encapsular um código que poderá ser invocado/chamado por qualquer outro trecho do programa.
- Seu significado e uso são muito parecidos com o de funções matemáticas, ou seja, existe um nome, uma definição e posterior invocação à função.

EXEMPLO

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Exemplo de função.\n";
    return $valor_retornado;
}
?>
```


EXEMPLO

- Funções definidas condicionalmente

```
<?php

$makefoo = true;

/* Nos nao podemos chamar foo() daqui
   porque ela ainda não existe,
   mas nos podemos chamar bar() */

bar();

if ($makefoo) {
    function foo()
    {
        echo "Eu não existo até que o programa passe por aqui.\n";
    }
}

/* Agora nos podemos chamar foo()
   porque $makefoo foi avaliado como true */

if ($makefoo) foo();

function bar()
{
    echo "Eu existo imediatamente desde o programa começar.\n";
}

?>
```

EXEMPLO

- Funções recursivas

```
<?php
function recursion($a)
{
    if ($a < 20) {
        echo "$a\n";
        recursion($a + 1);
    }
}
?>
```

PASSAGEM DE ARGUMENTOS

- Passagem por valor

```
<?php
function takes_array($input)
{
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
?>
```

PASSAGEM DE ARGUMENTOS

- Passagem por referência

```
<?php
function add_some_extra(&$string)
{
    $string .= ' e alguma coisa mais.';
}
$str = 'Isto é uma string,';
add_some_extra($str);
echo $str;    // imprime 'Isto é uma string, e alguma coisa mais.'
?>
```

VALORES PADRÃO DE ARGUMENTOS

```
<?php
function makecoffee($type = "cappuccino")
{
    return "Fazendo uma xícara de café $type.\n";
}
echo makecoffee();
echo makecoffee(null);
echo makecoffee("espresso");
?>
```

O exemplo acima irá imprimir:

```
Fazendo uma xícara de café cappuccino.
Fazendo uma xícara de café .
Fazendo uma xícara de café espresso.
```

VALORES PADRÃO DE ARGUMENTOS

```
<?php  
function bar(A $a = null, $b) {} // Ainda permitido  
function bar(?A $a, $b) {}      // Recomendado  
?>
```

NÚMERO VARIÁVEL DE ARGUMENTOS

```
<?php
function sum(...$numbers) {
    $acc = 0;
    foreach ($numbers as $n) {
        $acc += $n;
    }
    return $acc;
}

echo sum(1, 2, 3, 4);
?>
```

PESQUISA

- O que são argumentos nomeados? (Conceito introduzido no PHP 8.0)

LISTA DE EXERCÍCIOS 3

1. Faça um programa que leia uma data e verifique se ela é válida. Além disso informe ao usuário seu respectivo dia da semana.
2. Faça um programa que leia o nome de 10 alunos de uma turma e que ao enviar esses nomes eles sejam apresentados em ordem alfabética
3. Escreva um programa que leia dois números inteiros e uma função que retorne o menor número entre eles.
4. Escreva um programa que leia 5 números inteiros positivos. Para cada número informado, utilize uma função que escreva a soma de seus divisores (exceto ele mesmo).

Exemplo: divisores de 8: $1+2+4 = 7$

5. Faça um programa que leia a altura e o peso de uma pessoa, e que tenha uma função que calcule o IMC baseando-se nos valores lidos. Exiba o IMC e a classificação dele na tela.

Cálculo do IMC: $\text{peso} / \text{altura}$

IMC	Classificação
abaixo de 18,5	abaixo do peso
entre 18,6 e 24,9	Peso ideal (parabéns)
entre 25,0 e 29,9	Levemente acima do peso
entre 30,0 e 34,9	Obesidade grau I
entre 35,0 e 39,9	Obesidade grau II (severa)
acima de 40	Obesidade III (mórbida)



Obrigada