

Funções para *strings*

- strtoupper

Converte para maiúsculas

- ▶ Formato

```
strtoupper(string)
```

- ▶ Exemplo

```
<?php
//imprime MAIÚSCULO
echo strtoupper('maíusculo');
?>
```

No exemplo anterior, a letra "ú" permanece em minúsculo. Para funcionar com letras acentuadas, usar mb strtoupper

```
<?php
//imprime MAIÚSCULO
echo mb_strtoupper('maíusculo', 'utf-8');
?>
```

- strtolower

Converte para minúsculas

- ▶ Formato

```
strtolower(string)
```

- ▶ Exemplo

```
<?php
//imprime minÚsculo
echo strtolower('MINÚSCULO');
?>
```

No exemplo anterior, a letra "Ú" permanece em maiúsculo. Para funcionar com letras acentuadas, usar mb strtolower

```
<?php
//imprime minúsculo
echo mb_strtolower('MINÚSCULO', 'utf-8');
?>
```

Funções para *strings* (continuação)

○ substr

Retorna uma parte de uma *string*

► Formato

```
substr(string, começo[, tam])
```

► Exemplo

```
<?php
//retorna "f"
$rest = substr("abcdef", -1);

//retorna "ef"
$rest = substr("abcdef", -2);

//retorna "d"
$rest = substr("abcdef", -3, 1);
?>
```

○ str_replace

Substitui todas ocorrências

► Formato

```
str_replace(procura, substitui, texto)
```

► Exemplo

```
<?php
$txt = 'O Rato roeu a roupa';

//substitui Rato por Leão
echo str_replace('Rato', 'Leão', $txt);
?>
```

Funções para *strings* (continuação)

○ strpos

Encontra a posição da primeira ocorrência de uma *string*

► Formato

```
strpos(string, procurar_por)
```

► Exemplo

```
<?php
$txt = 'O Rato roeu a roupa';
$encontrar = 'roupa';

//informa a posição de 'roupa' (14)
echo "Posição de $encontrar: " .
strpos($txt, $encontrar);
?>
```

○ trim

Retira espaço no início e final de uma *string*

► Formato

```
trim(string)
```

► Exemplo

```
<?php
$str = '    olá mundo ';

echo trim($str);

echo strlen(trim($str));
echo strlen($str);
?>
```

Exercício 24

24) Incremente o arquivo `biblioteca.inc.php`, incluindo:

- a) Uma função para transformar todas as vogais da *string* fornecida por '?':

`oculta_vogais(string)`

Exemplo:

```
oculta_vogais("IFGoiano");  
  
//imprime: ?FG???n?
```

- b) Uma função para retornar uma *string* em ordem inversa:

`inverte(string)`

Exemplo:

```
inverte("Oi, pessoal!");  
  
//imprime: !laossep ,iO
```

Importante: não vale usar a função interna `strrev`

Inclua o arquivo `biblioteca.inc.php` no arquivo `ex24.php` e faça pelo menos três chamadas das funções recentemente criadas

Arrays

- ▶ *Arrays* (ou vetores) podem ser usados para armazenar
 - ▶ números
 - ▶ *strings*
 - ▶ objetos
 - ▶ outros *arrays*
 - ▶ etc...
- ▶ O PHP oferece uma série de funções internas úteis para o tratamento de *arrays*

Arrays

Criando um *array*

- ▶ Arrays são acessados por meio de sua posição (índice)
- ▶ Para criar um *array*, pode-se utilizar a função `array()`

```
$cores = array('vermelho', 'azul', 'verde');
```

ou

```
$cores = array(0=>'vermelho', 1=>'azul', 2=>'verde');
```

- ▶ Outra forma é adicionar valores diretamente com a sintaxe:

```
$nomes[] = 'maria';  
$nomes[] = 'joão';  
$nomes[] = 'carlos';  
$nomes[] = 'josé';
```

Arrays

Criando um *array* (continuação)

- De qualquer forma, para acessar o *array*, basta indicar seu índice entre colchetes:

```
echo $cores[0];    //resultado = vermelho
echo $cores[1];    //resultado = azul
echo $cores[2];    //resultado = verde
```

```
echo $nomes[0];    //resultado = maria
echo $nomes[1];    //resultado = joão
echo $nomes[2];    //resultado = carlos
echo $nomes[3];    //resultado = josé
```

Arrays

Arrays associativos

- ▶ Além de poder utilizar índices numéricos, também é possível usar uma *string* como chave de uma posição do *array*

```
$cores = array('vermelho' => '#FF0000', 'azul' => '#0000FF', 'verde' => '#00FF00');
```

Para facilitar a leitura, pode-se utilizar quebras de linha e espaços adicionais:

```
$cores = array('vermelho' => '#FF0000',  
              'azul'      => '#0000FF',  
              'verde'     => '#00FF00');
```

- ▶ Outra forma é simplesmente adicionar valores, com a sintaxe:

```
$pessoa['nome'] = 'Maria da Silva';  
$pessoa['rua']  = 'São João';  
$pessoa['bairro'] = 'Cidade Alta';  
$pessoa['cidade'] = 'Porto Alegre';
```


Arrays

Arrays associativos (continuação)

- De qualquer forma, para acessar o *array*, basta indicar sua chave entre colchetes:

```
echo $cores['vermelho']; //resultado = #FF0000
echo $cores['azul'];     //resultado = #0000FF
echo $cores['verde'];    //resultado = #00FF00
```

```
echo $pessoa['nome'];    //resultado = Maria da Silva
echo $pessoa['rua'];     //resultado = São João
echo $pessoa['bairro'];  //resultado = Cidade Alta
echo $pessoa['cidade'];  //resultado = Porto Alegre
```

Arrays

Iterações

- Os *arrays* podem ser iterados pelo operador FOREACH

```
<?php
    $frutas['cor']      = 'vermelha';
    $frutas['sabor']    = 'doce';
    $frutas['formato']  = 'redonda';
    $frutas['nome']     = 'maçã';

    foreach($frutas as $chave => $valor){
        echo "$chave => $valor<br>";
    }
?>
```



Resultado:

```
cor => vermelha
sabor => doce
formato => redonda
nome => maçã
```

Arrays

Acesso

- As posições de um *array* podem ser acessadas a qualquer momento e sobre elas operações podem ser realizadas

```
<?php
$minha_multa['carro'] = 'Pálio';
$minha_multa['valor'] = 178.00;

//alteração de valores
$minha_multa['carro'] .= ' ED 1.0';
$minha_multa['valor'] += 20;

//exibe o array
var_dump($minha_multa);
?>
```



Resultado:

```
array(2) {
    ["carro"]=>
    string(12) "Pálio ED 1.0"
    ["valor"]=>
    float(198)
}
```

Dica

Através da função `print_r` ou `var_dump` é possível imprimir toda a estrutura de um *array*.

Arrays

Arrays multidimensionais

- Arrays multidimensionais ou matrizes são *arrays* nos quais suas posições podem conter outros *arrays*

```
<?php
$carros = array('Palio' => array('cor' => 'azul',
                                'potência' => '1.0',
                                'opcionais' => 'Ar Cond.'),
               'Corsa' => array('cor' => 'cinza',
                                'potência' => '1.3',
                                'opcionais' => 'MP3'),
               'Gol'   => array('cor' => 'branco',
                                'potência' => '1.0',
                                'opcionais' => 'Metálica')
               );

echo $carros['Palio']['opcionais']; //resultado = Ar Cond.
?>
```

Arrays

Arrays multidimensionais (continuação)

- Outra forma de criar um *array* multidimensional é simplesmente atribuindo-lhe valores

```
<?php
$carros['Palio']['cor']          = 'azul';
$carros['Palio']['potência']    = '1.0';
$carros['Palio']['opcionais']   = 'Ar Cond.';

$carros['Corsa']['cor']         = 'cinza';
$carros['Corsa']['potência']    = '1.3';
$carros['Corsa']['opcionais']   = 'MP3';

$carros['Gol']['cor']           = 'branco';
$carros['Gol']['potência']      = '1.0';
$carros['Gol']['opcionais']     = 'Metálica';

echo $carros['Palio']['opcionais']; //resultado = Ar Cond.
?>
```

Arrays

Arrays multidimensionais (continuação)

- Para realizar iterações em um *array* multidimensional, é preciso observar quantos níveis ele possui

```
<?php
//...considerando que os arrays anteriores já estão no código

foreach($carros as $modelo => $caracts){
    echo "=> modelo $modelo<br>";

    foreach($caracts as $caract => $valor){
        echo "característica $caract => $valor<br>";
    }
}
?>
```

Resultado:

```
=> modelo Palio
característica cor => azul
característica potência => 1.0
característica opcionais => Ar Cond.
=> modelo Corsa
característica cor => cinza
característica potência => 1.3
característica opcionais => MP3
=> modelo Gol
característica cor => branco
característica potência => 1.0
característica opcionais => Metalica
```

Importante

Usar um laço de repetição e não `print_r` ou `var_dump` para imprimir o conteúdo dos arrays.

Exercícios 25, 26 e 27

- 25) Crie um *array* de 100 posições utilizando um laço de repetição (FOR, por exemplo), preenchendo cada posição com números aleatórios de 0 a 1000 (função `rand`). Em seguida, imprima o conteúdo do *array* utilizando um laço de repetição (FOREACH, por exemplo).

Salve seu arquivo como `ex25.php`

- 26) Crie um *array* chamado `$alunos` e utilize-o para armazenar o nome de pelo menos 5 alunos. Em seguida, altere o nome dos três primeiros alunos e, então, imprima o conteúdo do *array* utilizando um laço de repetição (FOREACH, por exemplo).

Salve seu arquivo como `ex26.php`

- 27) Crie um *array* multidimensional chamado `$clientes` e, em cada posição deste *array*, armazene nome, CPF, RG e telefone de pelo menos 5 clientes. Em seguida, imprima o conteúdo do *array* utilizando laços de repetição (FOREACH, por exemplo).

Salve seu arquivo como `ex27.php`