

Strings

Strings

- Uma string é uma série de caracteres.
- Um caracter é o mesmo que um byte, ou seja, há exatamente 256 caracteres diferentes possíveis. Isto implica que o PHP não tem suporte nativo ao Unicode.
- Sintaxe:
 - Apóstrofo (aspas simples)
 - Aspas (aspas duplas)
 - Heredoc

Apóstrofo

```
echo 'isto é uma string comum';  
echo 'Você pode incluir novas linhas em strings,  
dessa maneira que estará  
tudo bem';  
// Imprime: Arnold disse uma vez: "I\'ll be back"  
echo 'Arnold once said: "I\'ll be back"';  
// Imprime: Você tem certeza em apagar C:\*.*?  
echo 'Você tem certeza em apagar C:\\*.*?';  
// Imprime: Você tem certeza em apagar C:\*.*?  
echo 'Você tem certeza em apagar C:\\*.*?';  
// Imprime: Isto não será substituído: \n uma nova linha  
echo 'Isto não será substituído: \n uma nova linha';  
// Imprime: Variáveis $também não $expandem  
echo 'Variáveis $também não $expandem';
```

Aspas

- Caracteres que necessitam ser escapados

Seqüência	Significado
\n	fim de linha (linefeed ou LF ou 0x0A (10) em ASCII)
\r	retorno de carro (carriage return ou CR ou 0x0D (13) em ASCII)
\t	TAB horizontal (HT ou 0x09 (9) em ASCII)
\\	contra barra ou barra invertida
\\$	sinal de cifrão
\"	aspas
\[0-7]{1,3}	a seqüência de caracteres batendo a expressão regular dos caracteres em notação octal
\x[0-9A-Fa-f]{1,2}	a seqüência de caracteres batendo a expressão regular de um caracter em notação hexadecimal

Heredoc

```
$str = <<<EOD
```

```
Exemplo de uma string  
distribuída em várias linhas  
utilizando a sintaxe heredoc.
```

```
EOD;
```

```
echo $str;
```

```
function teste () {
```

```
    $var = <<<TEXT0
```

```
    Exemplo de uma string  
    distribuída em várias linhas  
    utilizando a sintaxe heredoc.
```

```
TEXT0;
```

```
}
```

Interpretação de variáveis

- Quando uma string é especificada dentro de aspas ou heredoc, variáveis são interpretadas dentro delas.
- Há dois tipos de sintaxe: um simples e um complexo . A sintaxe simples é a mais comum e conveniente, provendo uma maneira de interpretar uma variável, um valor de array ou uma propriedade de object.
- A sintaxe completa foi introduzida no PHP 4, e pode ser reconhecida por chaves ({}) envolvendo a expressão.

Interpretação de variáveis

- Sintaxe complexa

```
$bom = 'fantastico';  
echo "Isto é { $bom}";  
echo "Isto é {$bom}";  
echo "Isto é ${bom}";  
echo "Este quadrado tem {$square->width}00 centímetros de lado.";  
echo "Isto funciona: {$arr[4][3]}";  
echo "Isto funciona: {$arr['foo'][3]}";  
echo "Você pode escrever também {$obj->values[3]->name}";  
  
$nome = "teste";  
$teste = "conteudo";  
echo "Este é o valor da variável chamada $nome: ${${$nome}}";
```


Modificação de caracteres

- Caracteres nas strings podem ser acessados e modificados apenas especificando o deslocamento baseado em zero do caracter desejado depois da string dentro de chaves.
- Para manter a compatibilidade, você ainda pode usar colchetes de array para o mesmo propósito. Entretanto, esse sintaxe é obsoleta desde o PHP 4.

Modificação de caracteres

```
// Pega o primeiro caracter da string
```

```
$str = 'Isto é um teste.';
```

```
$primeiro = $str{0};
```

```
// Pega o terceiro caracter da string
```

```
$terceiro = $str{2};
```

```
// Pega o último caracter da string
```

```
$str = 'Isto ainda é um teste.';
```

```
$ultimo = $str{strlen($str)-1};
```

```
// Modifica o ultimo caracter da string
```

```
$str = 'Olhe o mal';
```

```
$str{strlen($str)-1} = 'r';
```

Escapando valores literais

```
echo 'Esta é \'minha\' string';
```

```
$a = 10;
```

```
echo "O valor de $a é \"$a\".";
```

```
echo "Escapando a contra-barra: - \ -";
```

```
echo "Escapando um literal chaves + cifrão : {\$";
```

Funções de string

- strlen
 - Tamanho da string
 - Cuidados devem ser tomados com os caracteres especiais que serão contabilizados

```
$str = 'abcdef';  
echo strlen($str); // 6
```

```
$str = ' ab cd ';  
echo strlen($str); // 7
```

```
$str = 'Teste de espaço';  
echo strlen($str); // 15
```

Funções de string

- `strtr`
 - Substitui ocorrências de strings por outras

```
$addr = strtr($addr, "ääö", "ao");
```

```
echo $addr;
```

```
$trans = array("hello" => "hi", "hi" => "hello");
```

```
echo strtr("hi all, I said hello", $trans);
```

Funções de string

- strcmp, strcasecmp, strncasecmp
 - Comparação segura de string para binário

```
$str = "Hello World";  
if (strcmp($str, "hello world") === 0) {  
    // Não entra aqui porque é case-insensitive  
}  
if (strcasecmp($str, "hello world") === 0) {  
    // Entra aqui porque é case-sensitive  
}
```

Funções de string

- strpos, stripos
 - Entra a posição da primeira ocorrência de uma string

```
$mystring = 'abc';  
$findme   = 'a';  
$pos = strpos($mystring, $findme);  
  
if ($pos === false) {  
    echo "A string '$findme' não foi encontrada na string '$mystring';"  
} else {  
    echo "A string '$findme' foi encontrada na string '$mystring';"  
    echo " e existe na posição $pos";  
}
```

Funções de string

- `strrpos`, `stripos`
 - Encontra a posição da última ocorrência de um caractere em uma string
 - A string de busca pode ser uma string de mais de um caractere no PHP 5, e números são convertidos para uma string ao invés de serem aplicados como valor ordinal de um caractere no PHP 4.

```
$mystring = "Curso bom";  
$pos = strrpos($mystring, "b");  
if ($pos !== false) { // note: três sinais iguais  
    print "ok";  
}
```


Funções de string

- strstr
 - Encontra a primeira ocorrência de uma string e retorna o restante da string original a partir da primeira posição em que foi encontrada a string de busca.

```
$email = 'user@example.com';  
$domain = strstr($email, '@');  
print $domain; // mostra @example.com
```

Funções de string

- `strspn`, `strcspn`
 - Encontra o comprimento do segmento inicial combinando com a máscara

```
$var = strspn("42 is the answer", "1234567890");  
print $var;
```

```
$var = strcspn("Number 42 is the answer", "1234567890");  
print $var;
```

Funções de string

- `str_replace`, `str_ireplace`, `substr_replace`
 - Substitui todas as ocorrências da string de procura com a string de substituição

```
$bodytag = str_replace("%body%", "black", "<body text='%body%'" );
```

```
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
```

```
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");
```

```
$frase = "você comeria frutas, vegetais, e fibra todos os dias.";
```

```
$saudavel = array("frutas", "vegetais", "fibra");
```

```
$saboroso = array("pizza", "cerveja", "sorvete");
```

```
$novafrase = str_replace($saudavel, $saboroso, $frase);
```

```
$str = str_replace("ll", "", "good golly miss molly!", $count);
```

```
echo $count; // 2
```

```
echo substr_replace("Canned tomatoes are good", "potatoes", 7, 8);
```

Funções de string

- substr
 - Retorna parte de uma string

```
$rest = substr("abcdef", 1);      // retorna "bcdef"  
$rest = substr("abcdef", 1, 3);  // retorna "bcd"  
$rest = substr("abcdef", -2);    // retorna "ef"  
$rest = substr("abcdef", -3, 1); // retorna "d"  
$rest = substr("abcdef", 0, -1); // retorna "abcde"  
$rest = substr("abcdef", 2, -1); // retorna "cde"  
$rest = substr("abcdef", -3, -1); // retorna "de"
```

// Outra opção é acessar através de chaves

```
$string = 'abcdef';
```

```
echo $string{0};           // retorna a
```

```
echo $string{3};          // retorna d
```

Funções de string

- Formatação

```
/* Define o local para o Português(Brasil) ) */  
setlocale (LC_ALL, 'pt_BR', 'ptb_bra');  
  
echo strftime ("%A, %d %B %Y", time());  
  
echo 10.5;  
  
echo number_format("100000.698"); // Mostra 100,001  
  
// Mostra 100 000,698  
echo number_format("100000.698", 3, ",", " ");  
  
echo money_format('%.2n', "100000.698"); //LINUX
```

Formatação

- printf, sprintf, vsprintf
 - Retorna uma string produzida de acordo com uma string de formatação.

```
$num = 15;  
$location = "zoo";  
$format = "There are %d monkeys in the %s";  
printf($format, $num, $location);
```

```
$format = "The %s contains %d monkeys";  
printf($format, $num, $location);
```

```
$format = "The %2\${s} contains %1\${d} monkeys.  
That's a nice %2\${s} full of %1\${d} monkeys.";   
printf($format, $num, $location);
```

Formatação

```
$year = 2007; $month = 8; $day = 2;  
$isodate = sprintf("%04d-%02d-%02d", $year, $month, $day);  
print $isodate;
```

```
$money1 = 68.75;  
$money2 = 54.35;  
$money = $money1 + $money2;  
// echo $money irá mostrar "123.1";  
$formatted = sprintf("%01.2f", $money);  
// echo $formatted irá mostrar "123.10"  
print $formatted;
```


Formatação

- % - Um caractere por cento. Não é requerido nenhum argumento.
- b - O argumento é tratado com um inteiro, e mostrado como um binário.
- c - O argumento é tratado como um inteiro, e mostrado como o caractere ASCII correspondente.
- d - O argumento é tratado como um inteiro, e mostrado como um número decimal com sinal.
- u - O argumento é tratado com um inteiro, e mostrado como um número decimal sem sinal.
- f - O argumento é tratado como um float, e mostrado como um número de ponto flutuante.
- o - O argumento é tratado com um inteiro, e mostrado como um número octal.
- s - O argumento é tratado e mostrado como uma string.
- x - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras minúsculas).
- X - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras maiúsculas).

Formatação de entrada

- sscanf

```
// Pegando o número serial
$serial = sscanf("SN/2350001", "SN/%d");
// e a data de criação
$mandate = "January 01 2000";
list($month, $day, $year) = sscanf($mandate, "%s %d %d");
echo "O Item {$serial[0]} foi criado em:
      $year-".substr($month, 0, 3). "-$day\n";

// pega informação do autor e gera uma entrada de DocBook
$auth = "24\tLewis Carroll";
$ln = sscanf($auth, "%d\t%s %s", $id, $first, $last);
echo "<author id='$id'>
      <firstname>$first</firstname>
      <surname>$last</surname>
</author>\n";
```

Formatação de entrada

- fscanf
 - Interpreta a leitura de um arquivo de acordo com um formato

```
$handle = fopen ("users.txt", "r");  
while ($userinfo = fscanf ($handle, "%s\t%s\t%s\n")) {  
    list ($name, $profission, $countrycode) = $userinfo;  
    //... fazer algo com os valores  
}  
fclose($handle);
```

```
javier   argonauta      pe  
hiroshi  escultor      jp  
robert   tradutor      us  
luigi    florista      it
```

Expressões Regulares (PCRE)

- Perl-compatible Regular Expression (PCRE) oferece um poderoso mecanismo de comparação e substituição de string
- Metacaracteres:

.	Match any character
^	Match the start of the string
\$	Match the end of the string
\s	Match any whitespace character
\d	Match any digit
\w	Match any “word” character

Expressões Regulares (PCRE)

- Quantificadores

*	The character can appear zero or more times
+	The character can appear one or more times
?	The character can appear zero or one times
{n,m}	The character can appear at least n times, and no more than m. Either parameter can be omitted to indicated a minimum limit with no maximum, or a maximum limit without a minimum, but not both.

Expressões Regulares (PCRE)

- Exemplos:

/ab[cd]e/

/ab[c-e\d]/

/ab?c/

/ab{1,3}c/

/a(bc.)e/

/a(bc.)+e/

Expressões Regulares (PCRE)

- preg_match
 - Extração e busca simples

```
$name = "Davey Shafik";  
// Simple match  
$regex = "[a-zA-Z\s]";  
if (preg_match($regex, $name)) {  
    print "found";  
}  
// Match with subpatterns and capture  
$regex = '/^(\\w+)\\s(\\w+)/';  
$matches = array();  
if (preg_match ($regex, $name, $matches)) {  
    var_dump ($matches);  
}
```

Expressões Regulares (PCRE)

- preg_match_all
 - Extração e busca de todas a ocorrências encontradas

```
$string = "a1bb b2cc c2dd";  
$regex = "#([abc])\d#";  
$matches = array();  
if (preg_match_all ($regex, $string, $matches)) {  
    var_dump ($matches);  
}
```

Expressões Regulares (PCRE)

- preg_replace
 - Executa uma expressão regular para busca e substituição

```
$body = "[b]Make Me Bold![/b]";  
print $body;  
$regex = "@\[b\](.*?)\[\/b\]@i";  
$replacement = '<b>$1</b>';  
$body = preg_replace($regex, $replacement, $body);  
print $body;
```