**Linguagem de Programação – PHP**

Universidade Federal de Sergipe – UFS

Departamento de Sistemas de Informação – DSI

Alesandro da Silva Santos

Eliane Dantas de Jesus

Natalia da Silva Costa

**1. Introdução**

A linguagem de programação PHP foi criada em 1994 por Rasmus Lerdorf, a qual significava Personal Home Page Tools. No início era formada por um conjunto de scripts escritos em linguagem C, os quais tinham o objetivo a criação de páginas dinâmicas, que eram utilizadas por Rasmus para verificar os acessos ao seu currículo na internet. Como mais pessoas passaram a utilizar esta linguagem, em 1995 seu código fonte foi liberado, com isso, novos desenvolvedores se juntaram ao projeto.

É uma linguagem de código-aberto, e que é compatível com diversos servidores Web, a exemplo do Apache. O PHP permite englobar pedaços de código em páginas de HTML normais. Além disso, facilita a conexão de suas páginas Web com o banco de dados do lado servidor. Em síntese, PHP possui inúmeras qualidades, a exemplo do fato de ser livre, ter código-aberto – como já mencionado anteriormente –, tem vários recursos, pode ser usado em diversas plataformas, é estável, rápido e fácil de aprender.

**2. Variáveis do PHP**

Para realizar a declaração de variáveis é necessário seguir algumas regras:

* + Para iniciar o programa deve-se utilizar a nomenclatura <?php, e em seguida é possível realizar a declaração das variáveis;
  + Para indicar o início usa-se o símbolo $ (cifrão), em seguida o nome da variável e depois é atribuído um valor à variável. Ex: $nome = ‘Maria’;
  + No PHP não é possível declarar variável sem atribuir um valor, além disso, por se tratar de uma linguagem de tipagem dinâmica, não é possível definir o tipo de uma variável na declaração;
  + O tipo de dado da variável é automaticamente definido através do tipo de dado atribuído a variável no PHP;
  + É uma linguagem case sensitive, ou seja, diferencia maiúsculas de minúsculas. Ex: $nome é diferente de $Nome;
  + Um nome de variável válido deve iniciar com um sublinhado OU uma letra, nunca deve ser iniciado com um número ou caractere especial;
  + Seguindo as regras acima, são válidos caracteres de a-z, números e caracteres das tabelas ASCII de 128 a 255.

**3. Comentários**

O PHP permite fazer comentários no estilo das linguagens C/C++ e Unix shell.

* + Comentário de apenas uma linha: // Exemplo de comentário
  + Comentário de várias linhas: /\* Exemplo de comentário \*/
  + Comentário de apenas uma linha no estilo Shell: # Exemplo de comentário

Os comentários de estilo "uma linha" apenas comentam até o final da linha ou do bloco PHP de código corrente, o que ocorrer primeiro. Isto significa que o código HTML após // ... ?> ou # ... ?> SERÁ impresso: ?> interrompe o modo PHP e retorna para o modo HTML, e // ou # não podem influenciar isto.

**4. Palavras reservadas**

Em todas as linguagens de programação existem palavras reservadas, as quais não podem ser usadas como identificadores. No PHP existem algumas dessas palavras, estas palavras têm um significado especial no PHP. Algumas delas representam coisas que parecem funções, algumas parecem constantes, e assim por diante - mas na verdade não o são realmente: elas são constructos da linguagem. As palavras seguintes não podem ser usadas como constantes, nomes de classes, ou nome de funções. Estas palavras-chave são permitidas em propriedades, constantes e nomes de métodos em classes, interfaces e traits, com exceção de class, que não pode ser utilizada como nome de constante. A seguir é apresentada algumas dessas palavras reservadas.

* + **And:** é um operador lógico;
  + **Break:** finaliza a execução da estrutura for, foreach, while, do- while ou switch atual;
  + **Case:** é utilizado no comando condicional switch;
  + **Const:** define constantes no programa;
  + **Continue:** é utilizado em estruturas de laço para pular o resto da iteração atual, e continuar a execução na validação da condição e, então, iniciar a próxima iteração;
  + **Declare:** é usada para definir diretivas de execução para um bloco de código;
  + **Default:** utilizado no comando switch;
  + **Do:** é usado em comandos de repetição, como o do while;
  + **Echo:** exibe uma ou mais strings;
  + **Else/Else if:** comandos condicionais;
  + **Empty:** determina se a variável é vazia;
  + O PHP oferece uma sintaxe alternativa para algumas estruturas de controle; a saber, if, while, for, foreach e switch. Em cada caso, basicamente a sintaxe alternativa é trocar a chave de abertura por dois pontos (:) e a chave de fechamento por **endif;, endwhile;, endfor;, endforeach;,** ou **endswitch**;, respectivamente;
  + **Exit:** mostra uma mensagem e termina o script atual;
  + **For:** comando de repetição;
  + **Foreach:** fornece uma maneira fácil de iterar sobre arrays. O foreach funciona somente em arrays e objetos, e emitirá um erro ao tentar usá-lo em uma variável com um tipo de dado diferente ou em uma variável não inicializada;
  + **Function:** usada na criação de funções;
  + **Global:** cria uma variável visível em todo o programa;
  + **If:** comando condicional;
  + **Implements:** para implementar uma interface, o operador implements é utilizado;
  + **Include:** inclui e avalia o arquivo informado;
  + **Include\_once:** inclui e avalia o arquivo informado durante a execução do script. Este é um comportamento similar a declaração include, com a única diferença que, se o código do arquivo já foi incluído, não o fará novamente, e o include\_once retornará true. Como o nome sugere, o arquivo será incluído somente uma vez;
  + **Isset:** informa se a variável foi iniciada;
  + **List:** cria variáveis como se fossem arrays;
  + **Match:** ramifica a avaliação baseada em uma verificação de identidade de um valor. Semelhante a uma declaração switch, uma expressão match possui uma expressão sujeito que é comparada com múltiplas alternativas. Ao contrário da switch, ela irá avaliar para um valor assim como as expressões ternárias. Diferente da switch, a comparação é uma verificação de identidade (===) em vez de uma comparação de equalidade fraca (==);
  + **Namespace:** são uma forma de encapsular itens;
  + **Or:** é um operador lógico;
  + **Print:** mostra uma string;
  + **Private:** está ligado à visibilidade da variável;
  + **Protected:** está ligado à visibilidade da variável;
  + **Public:** está ligado à visibilidade da variável;
  + **Return:** retorna o controle do programa para o módulo que o chamou. A execução continuará na expressão seguinte à invocação do módulo;
  + **Static:** está ligada ao escopo da variável;
  + **Switch:** é similar a uma série de declarações IF na mesma expressão;
  + **Throw:** lança uma exceção;
  + **Traits:** é um mecanismo para reutilização de código em linguagens de herança única, como PHP;
  + **Unset:** destrói a variável especificada;
  + **While:** é um comando de repetição;
  + **Xor:** é um operador lógico;
  + **Yield:** fornece um valor para o código de loop sobre o generator e pausa a execução da função do generator.

**5. Literais reservados**

* + **null:** representa uma variável sem valor;
  + **true:** valor lógico que representa verdadeiro;
  + **false:** valor lógico que representa falso.

**6. Operadores e delimitadores**

Em PHP os operadores são:

## **Operadores Aritméticos**

|  |  |  |
| --- | --- | --- |
| **Exemplo** | **Nome** | **Resultado** |
| +$a | Identidade | Conversão  de $a para int ou float conforme apropriado. |
| -$a | Negação | Oposto de $a |
| $a + Sb | Adição | Soma de $a e $b |
| $a - $b | Subtração | Diferença de $a e $b |
| $a \* $b | Multiplicação | Produto de $a e Sb |

|  |  |  |
| --- | --- | --- |
| $a / $b | Divisão | Quociente de $a e $b |
| $a % $b | Módulo | Resto de $a dividido por $b |
| $a \*\* $b | Exponenciação | Resultado de *$a* elevado a *$b*.  Introduzido no PHP 5.6. |

* + **Operador de Atribuição**  utiliza o “=” para atribuir um valor a um operando. Além disso, para todos os operandos aritméticos há “operadores combinados” como: $a += 5; //soma 5 ao valor atribuído em a.

$b = “dia!”; //adiciona dia! Ao que está armazenado em b.

* + **Operadores de Comparação**

|  |  |  |
| --- | --- | --- |
| **Exemplo** | **Nome** | **Resultado** |
| $a == $b | Igual | Verdadeiro (**true**) se $a é igual a  $b. |
| $a === $b | Idêntico | Verdadeiro (**true**) se $a é igual a  $b, e eles são do mesmo tipo. |
| $a != $b | Diferente | Verdadeiro se $a não é igual a $b. |
| $a <> $b | Diferente | Verdadeiro se $a não é igual a $b. |
| $a !== $b | Não idêntico | Verdadeiro de $a não é igual a $b, ou eles não são do mesmo tipo (introduzido no PHP4). |
| $a < $b | Menor que | Verdadeiro se $a é estritamente  menor que $b. |
| $a > $b | Maior que | Verdadeiro se $a é estritamente  maior que $b. |
| $a <= $b | Menor ou igual | Verdadeiro se $a é menor ou igual a $b. |
| $a >= $b | Maior ou igual | Verdadeiro se $a é maior ou igual  a $b. |

* + **Operadores de controle de erro**  utiliza um operador de controle de erro: o sinal 'arroba' (@). Quando ele precede uma expressão em PHP, qualquer mensagem de erro que possa ser gerada por aquela expressão será ignorada.

## **Operadores de Incremento/Decremento**

|  |  |  |
| --- | --- | --- |
| **Exemplo** | **Nome** | **Efeito** |
| ++$a | Pré-incremento | Incrementa $a em um, e então  retorna $a. |
| $a++ | Pós-incremento | Retorna $a, e então incrementa $a em um. |
| --$a | Pré-decremento | Decrementa $a em um, e então  retorna $a. |
| $a-- | Pós-decremento | Retorna $a, e então decrementa $a  em um. |

* + **Operadores Lógicos**

|  |  |  |
| --- | --- | --- |
| **Exemplo** | **Nome** | **Resultado** |
| $a and $b  $a & $b  $a && $b | E | Verdadeiro se tanto $a quanto $b são verdadeiros. |
| $a or $b  $a | $b  $a || $b | OU | Verdadeiro se $a ou $b são verdadeiros. |
| $a xor $b  $a ^ $b | XOR | Verdadeiro se $a ou $b são verdadeiros, mas não ambos. |
| !$a | NÃO | Verdadeiro se $a não é verdadeiro. |

* + **Operadores de String**  Há dois operadores de string. O primeiro é o operador de concatenação ('.'), que retorna a concatenação dos seus argumentos direito e esquerdo. O segundo é o operador de atribuição de concatenação ('.='), que acrescenta o argumento do lado direito no argumento do lado esquerdo. Exemplo:

<?php

$a = "Olá ";

$b = $a . "mundo!"; // agora $b contém "Olá mundo!"

$a = "Olá ";

$a .= "mundo!"; // agora $a contém "Olá mundo!"

?>

## **Operadores de Arrays**

|  |  |  |
| --- | --- | --- |
| **Exemplo** | **Nome** | **Resultado** |
| $a + $b | União | União de $a e $b. |
| $a == $b | Igualdade | true se $a e $b tem os mesmos pares de chave/valor. |
| $a === $b | Identidade | true se $a e $b tem os mesmos pares de chave/valor na mesma  ordem e do mesmo tipo. |
| $a != $b | Desigualdade | true se $a não é igual a $b. |
| $a <> $b | Desigualdade | true se $a não é igual a $b. |
| $a !== $b | Não identidade | true se $a não é identico a $b. |

* **Precedência dos operadores**

|  |  |  |
| --- | --- | --- |
| **Associação** | **Operadores** | **Informação Adicional** |
| Esquerda | [ | Array() |
| Direita | \*\* | Aritmética |
| Direita | ++ -- (int) (float) (string) (array) (bool) @ | types e incremento/decremento |
| Direita | ! | Lógicos |
| Esquerda | \* / % | Aritmética |
| Esquerda | + - . | Aritmética e string |
| Não associativo | < <= > >= | Comparação |
| Não associativo | == != === !== <> | Comparação |
| Esquerda | & && and ^ xor | || or | Lógicos |
| Direita | ?? | Comparação |
| Esquerda | ? : | Ternário |
| Direita | = += -  = \*= \*\*= /= .= %= &= |  = ^= <<= >>= | Atribuição |

# Literais String

Uma string é uma série de caracteres, onde um caractere é o mesmo que um byte. Isso significa que o PHP possui suporte a um conjunto de apenas 256 caracteres, e, portanto, não possui suporte nativo a Unicode.

Uma string literal pode ser especificada de duas formas diferentes, as quais são:

* **Aspas simples:** $nome = ‘Maria da Silva’;
* **Aspas duplas:** $nome = “Maria da Silva”;

# Literais Numéricos

## **Inteiros**

Inteiros podem ser especificados em notação decimal (base 10), hexadecimal (base 16), octal (base 8) ou binária (base 2). O operador de negação pode ser usado para indicar um inteiro negativo. A partir do PHP 7.4.0, inteiros literais podem conter sublinhados (\_) entre os dígitos, para melhorar a sua leitura.

Ex: $a = 1200; // base 10

$b = 0123; // base 8 (equivale ao número 83 em decimal)

$c = 0x1A; // base 16 (equivale ao número 26 em decimal)

$d = 0b11111111; // base 2 (equivale ao número 255 em decimal)

$e = 1\_234\_567; // número decimal a partir do PHP 7.4.0

## **Números de ponto flutuante**

Números de ponto flutuante (também conhecidos como "floats", "doubles" ou "números reais"), podem ser especificados utilizando qualquer uma das seguintes sintaxes:

$a = 1.267;

$b = 1.2e3;

$c = 7E-10;

$d = 1\_234.567;

## **Booleanos**

Um bool expressa um valor de verdade. Ele pode ser true ou false. Para especificar o tipo booleano deve-se usar as palavras reservadas true ou false, sendo que ambas são case sensitive.

Para converter explicitamente um valor para bool, é utilizado os modificadores (bool) ou (boolean). Entretanto, na maioria dos casos, o modificador não será necessário, já que qualquer valor será convertido automaticamente se um operador, função ou estrutura de controle requerer um argumento bool.

Ao converter para bool, os seguintes valores são considerados false:

* + - o próprio booleano false
    - o inteiro 0 (zero)
    - os pontos flutuantes 0.0 e -0.0 (zero)
    - uma string vazia e a string "0"
    - um array sem elementos
    - o tipo especial NULL (incluindo variáveis não definidas)
    - objetos SimpleXML criados a partir de elementos vazios sem atributos, ou seja, elementos que não têm filhos nem atributos.

Com exceção dos valores acima, qualquer outro é considerado true.

## **Referências:**

CONVERSE, Tim; PARK, Joyce. **PHP: a bíblia**. Gulf Professional Publishing, 2003.

DALL’OGLIO, Pablo. **PHP Programando com Orientação a Objetos 3ª Edição**. Novatec Editora, 2015.

**Variáveis no PHP. Acesse e aprenda como declarar!** Disponível em: <https://www.treinaweb.com.br/blog/variaveis-no-php-acesse-e-aprenda-como-declarar/>. Acesso em: 4 fev. 2023.

**Manual do PHP.** Disponível em: <https://www.php.net/manual/pt_BR/index.php>. Acesso em: 4 fev. 2023