



Operações Aritméticas

Academia Indiana

Operadores Aritméticos



São aqueles que estudamos na escola: somar, subtrair, multiplicar, dividir, etc.

Neste caso se os operandos são números(INTEGER e FLOAT) podemos usá-los normalmente, se forem de outro tipo, seus valores serão convertidos antes da operação.

Com estes operadores podemos fazer qualquer operação matemática com tipo de dados numéricos no PHP:

- Adição: **(+)**
- Subtração: **(-)**
- Multiplicação: **(*)**
- Divisão: **(/)**
- Módulo - resto da divisão: **(%)**

Operadores de decremento e incremento



São operadores aritméticos que trabalham apenas com um operando, os chamados operadores unários e que servem para: trocar sinais, incrementar ou decrementar valores:

- `-operando`: troca o sinal do operando.
- `++operando`: pré-incremento, primeiro incrementa depois realiza a operação.
- `--operando`: pré-decremento, primeiro decrementa depois realiza a operação.
- `operando++`: pós-incremento, primeiro faz a operação depois incrementa.
- `operando--`: pós-decremento, primeiro faz a operação depois decrementa.



Operadores de atribuição

O PHP possui um operador básico de atribuição e diversos derivados. Nesses casos a operação é feita entre os dois operandos, e o resultado é atribuído ao primeiro:

- Atribuição simples(=)
- Atribuição com adição (+=)
- Atribuição com subtração (-=)
- Atribuição com multiplicação (*=)
- Atribuição com divisão (/=)
- Atribuição com módulos (%=)



Precedência de Operadores

Quando usamos mais de um operador em uma expressão o PHP verifica a precedência de cada um deles. Ou seja determinados operadores possuem menor, igual ou maior prioridade do que os demais.

Os parênteses podem ser utilizados para forçar a precedência, se necessário.

```
<?php
```

```
echo 20 + 4 * 2; // 28
```

```
echo (8 - 5 ) * 3; // 9
```

```
?>
```



Operadores de Comparação

Operadores de comparação ou condicionais, são aqueles capazes de fazer comparações entre variáveis, com eles podemos saber se uma variável é maior que a outra, diferente, etc

- `(==)` Igual a
- `(===)` Idêntico, e do mesmo tipo de dados
- `(!=),(<>)` Diferente de
- `(!==)` Não idêntico
- `(<)` Menor que
- `(>)` Maior que
- `(<=)` Menor ou igual a
- `(>=)` Maior ou igual a
- `(<=>)` Define um inteiro(-1,0,1) caso o resultado da comparação for menor que, igual ou maior que, respectivamente



Operadores lógicos

Os operadores lógicos trabalham com valores booleanos, com o objetivo de avaliar expressões cujo valor pode ser verdadeiro ou falso, ou seja, implementando a lógica booleana.

- **and(e)** – O resultado verdadeiro só será obtido quando ambos dos dados comparados forem verdadeiros.
- **or(ou)** – O resultado verdadeiro só será obtido quando pelo menos um dos dados comparados for verdadeiro.
- **xor** – O resultado verdadeiro só será obtido quando pelo menos um dos dados comparados for verdadeiro, mas não ambos.
- **!(não)** – O resultado verdadeiro só será obtido se o valor dado não for verdadeiro.
- **&&(e)** – Igual ao operador AND.
- **||(ou)** – Igual ao operador OR.

Informação importante:

- **&& e ||** = precedência alta;
- **and e or** = precedência baixa.



Estruturas condicionais

Um dos conceitos mais importantes da programação é o de estruturas de controle de fluxo.

São estruturas utilizadas para modificar o funcionamento da aplicação em certo ponto, fazendo com que decisões sejam tomadas e que trechos diferentes de código sejam executados de acordo com uma condição.

São parte fundamental em qualquer linguagem de programação, e basicamente determinam as condições que guiam por onde o código deve prosseguir.

As estruturas condicionais do PHP são: **IF**, **ELSE IF**, **ELSE** e **SWITCH**.



Operador if

O operador **IF** é utilizado para avaliar o valor de uma condição booleana, ou seja, que pode assumir apenas dois valores distintos: VERDADEIRO (true) ou FALSO (false).

A sintaxe do operador if é a seguinte:

```
if (condição)
{
    //código
}
```

As chaves são utilizadas para delimitar um bloco de código, caso fosse necessário executar apenas uma linha, as chaves poderiam ser removidas.



If Else...

O **operador IF** pode ser complementado com o operador ELSE, que significa SENÃO. Ou seja, **SE** uma condição for atendida, fazer uma coisa. **SENÃO**, fazer outra. O ELSE só precisa ser utilizado quando houver algo específico a ser feito se a condição não for atendida.

```
if (condição)
{
    //código 1
}
else
{
    //código 2
}
```



If else if...

Caso existam várias condições específicas a serem validadas, você pode inserir outros IFs após o else, conforme a sintaxe abaixo:

```
if(condição1)
{
    //código 1
}
else if(condição2)
{
    //código 2
}
else
{
    //código 3
}
```



Exemplo de uso

```
$hora = 7;  
  
if($hora <= 12){  
  
echo "Bom dia";  
  
} else if($hora > 12 && $hora <= 18){  
  
    echo "Boa tarde";  
  
} else {  
  
    echo "Boa noite";  
  
}
```



Operador Switch

o **operador SWITCH**, avalia uma expressão com base em vários valores predeterminados.

Se a expressão não possuir nenhum dos valores válidos, um valor padrão pode ser definido. Abaixo temos a sintaxe básica dessa estrutura.

```
switch($opcao)
{
    case 1:
        //código 1
        break;
    case 2:
        //código 2
        break;
    default:
        //nenhuma das opções
        break;
}
```



Exemplo de uso do switch

```
switch(4){  
    case 1:  
        $result = 'O valor é 1';  
        break;  
    case 2:  
        $result = 'O valor é 2';  
        break;  
    case 3:  
        $result = 'O valor é 3';  
        break;  
    case 4:  
        $result = 'O valor é 4';  
        break;  
    default:  
        $result = 'Não foi encontrado';  
}  
echo $result;
```

Exercícios



1- Termine o script abaixo, somando o valor das variáveis \$valor1 e \$valor2, e multiplicando o resultado pelo valor de \$valor1 exibindo o resultado na tela.

```
<?php  
$valor1 = 2;  
$valor2 = 7;  
  
...
```

Exercícios



2- Termine o script abaixo, calculando a média dos valor de \$valor1, \$valor2 e \$valor3, armazenando o resultado na variável \$media e exibindo na tela a mensagem: Média calculada: [valor]

```
<?php  
$valor1 = 7;  
$valor2 = 3;  
$valor3 = 9;  
  
...
```


Exercícios



3- Termine o script abaixo. A variável \$preco contém o valor da venda. Acrescente 12% ao preço do produto, divida em 10 parcelas e mostre na tela o valor de cada parcela e o valor total da compra.

```
<?php
```

```
$preco = 320;
```

```
...
```

Exercícios



4- Termine o script abaixo, verificando se a variável \$numero é par ou ímpar e exiba na tela a mensagem “número par” ou “numero impar”.

```
<php
```

```
$numero = 7;
```

```
...
```

Exercícios



5- Finalize o script abaixo verificando se valor da variável \$a é maior ou menor que o valor da variável \$b. A mensagem a ser impressa deve ser “A maior que B” ou “A menor que B”.

```
<?php
```

```
$a = 7;
```

```
$b = 21;
```

```
...
```

Exercícios



6- Termine o script abaixo, verificando se a variável \$numero valor positivo, negativo ou igual a zero. Exiba na tela a mensagem: "Valor Positivo", "Valor Negativo" ou "Igual a Zero".

```
<?php
```

```
$numero = 0;
```

Exercícios



7- Termine o script abaixo, exibindo o nome do mês correspondente à variável \$numMes e caso o valor de \$numMes esteja fora do intervalo entre 1 e 12, exibir a mensagem “Mês inválido”

```
<?php
```

```
$numMes = 8;
```

```
...
```