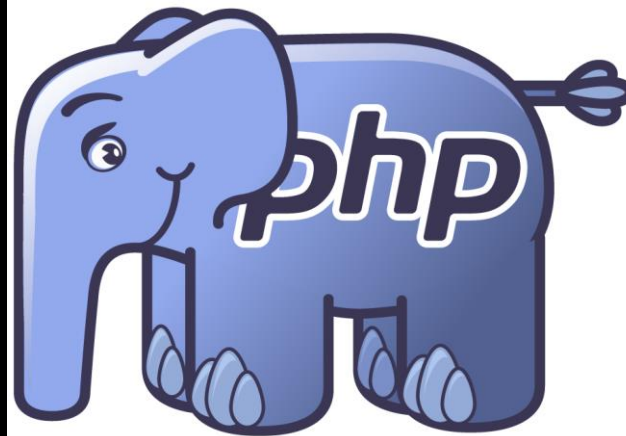




Estácio

DESENVOLVIMENTO WEB

HTML, CSS, JavaScript



Professor: Dr. **Rodrigo Fujioka**

<https://www.fujideia.com.br/fuji> | [linkedin/rodrigofujioka](https://www.linkedin.com/in/rodrigofujioka) | [@rodrigofujioka](https://www.instagram.com/rodrigofujioka)

ESSE MATERIAL TRATA DO PHP

- Tipo e estrutura de controles.

OBJETIVOS DO CURSO



***Apresentar as variáveis em
PHP***



Realizar Prática.

TIPOS EM PHP

O PHP suporta
oito tipos
primitivos.

São quatro tipos
básicos:

* boolean

* integer

* float (número
de ponto
flutuante, ou
também double)

* string

TIPOS EM PHP

O PHP suporta oito tipos primitivos.

Dois tipos compostos:

* array

* object

E finalmente dois tipos especiais:

* resource

* NULL

BOOLEANOS

Booleanos

Este é o tipo mais simples. Um booleano expressa um valor verdade. Ele pode ser TRUE ou FALSE.

Nota: O tipo booleano foi introduzido no PHP 4.

Sintaxe

Para especificar um literal booleano, use as palavras-chave TRUE ou FALSE. Ambas são case-insensitive.

```
<?php
$foo = True; // atribui o valor True para $foo
?>
```

INTEIROS

Inteiros

Um inteiro é um número do conjunto $Z = \{..., -2, -1, 0, 1, 2, ...\}$.

Veja também: [Inteiros de tamanho arbitrário / GMP](#), [Números de ponto flutuante](#) e [Precisão arbitrária / BCMath](#).

Sintaxe

Inteiros podem ser especificados em notação decimal (base 10), hexadecimal (base 16) ou octal (base 8), opcionalmente precedido de sinal (- ou +).

Para usar a notação octal, você precisa preceder o número com um 0 (zero). Para utilizar a notação hexadecimal, preceda número com 0x.

Exemplo #1 Literais inteiras

```
<?php
$a = 1234; // número decimal
$a = -123; // um número negativo
$a = 0123; // número octal (equivalente a 83 em decimal)
$a = 0x1A; // número hexadecimal (equivalente a 26 em decimal)
?>
```

PONTO FLUTUANTE

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:

```
<?php  
$a = 1.234;  
$b = 1.2e3;  
$c = 7E-10;  
?>
```


STRING

Strings

Uma [string](#) é uma série de caracteres. Antes do PHP 6, 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. Veja [utf8_encode\(\)](#) e [utf8_decode\(\)](#) para algumas funcionalidades básicas de Unicode.

Nota: Não é problema para uma [string](#) ser bastante longa. PHP não impõe limite de tamanho de uma [string](#); o único limite é o de memória disponível do computador no qual o PHP está sendo executado.

STRING

Sintaxe

Uma string literal pode ser especificada de quatro formas diferentes.

- apóstrofo
- aspas
- sintaxe heredoc
- sintaxe nowdoc (desde o PHP 5.3.0)

- http://www.php.net/manual/pt_BR/language.types.string.php

ARRAY

```
<?php
$arr = array("foo" => "bar", 12 => true);

echo $arr["foo"]; // bar
echo $arr[12];    // 1
?>
```

OBJECT

```
<?php
class foo
{
    function do_foo()
    {
        echo "Doing foo.";
    }
}

$bar = new foo;
$bar->do_foo();
?>
```

RESOURCE

Resource

Um recurso é uma variável especial, que mantém uma referência a um recurso externo. Recursos são criados e usados por funções especiais. Veja o [apêndice](#) para uma lista de todas essas funções e seus tipos correspondentes.

Nota: O tipo [resource](#) foi incluído no PHP 4.

- http://www.php.net/manual/pt_BR/language.types.resource.php

NULL

Sintaxe

Há apenas um único valor do tipo null, e é a palavra-chave case-insensitive NULL.

```
<?php  
$var = NULL;  
?>
```

- http://www.php.net/manual/pt_BR/language.types.resource.php

```
<?php
```

```
// Nomes de constantes válidos
```

```
define("FOO", "alguma coisa");
```

```
define("FOO2", "alguma outra coisa");
```

```
define("FOO_BAR", "alguma coisa mais");
```

```
// Nomes de constantes inválidas
```

```
define("2FOO", "alguma coisa");
```

```
// Isto é válido, mas deve ser evitado:
```

```
// O PHP pode vir a fornecer uma constante mágica
```

```
// que danificará seu script
```

```
define("__FOO__", "alguma coisa");
```

```
?>
```

```
<form action="welcome.php" method="get">
Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
```

- Crie um Form

```
Welcome <?php echo $_GET["fname"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
```

Crie um script PHP para receber os dados via get.

\$_REQUEST

```
Welcome <?php echo $_REQUEST["fname"]; ?>!  
You are <?php echo $_REQUEST["age"]; ?> years old.
```

CONSIDERAÇÕES FINAIS

Dúvidas?



PRÁTICA

Calculadora

Soma
Subtracao
Multiplicacao
Divisao

Operação



- <http://www.w3schools.com/php/>
- http://php.net/manual/pt_BR/index.php

PERGUNTAS ?

<https://github.com/rodrigofujioka/estaciodevweb>



OBJETIVOS DO CURSO

- *Operadores de Atribuição*
- *Estrutura de Controle.*

OPERADORES ARITMÉTICOS

Operador	Função	Exemplo
+	Adição	$\$a + \b
-	Subtração	$\$a - \b
*	Multiplicação	$\$a * \b
/	Divisão	$\$a / \b
%	Módulo (resto da divisão)	$\$a \% \b

OPERADORES DE ATRIBUIÇÃO

Operador	Função	Exemplo	Equivalente a
<code>+=</code>	Atribuição e adição	<code>\$a += \$b</code>	<code>\$a = \$a + \$b</code>
<code>-=</code>	Atribuição e Subtração	<code>\$a -= \$b</code>	<code>\$a = \$a - \$b</code>
<code>*=</code>	Atribuição e Multiplicação	<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>
<code>/=</code>	Atribuição e Divisão	<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>
<code>%=</code>	Atribuição e Módulo	<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>
<code>.=</code>	Atribuição e concatenação	<code>\$a .= \$b</code>	<code>\$a = \$a . \$b</code>

OPERADORES DE INCREMENTO

Operador	Nome	Função
<code>++\$a</code>	Pré-incremento	Incrementa <code>\$a</code> em um, e então retorna <code>\$a</code> .
<code>\$a++</code>	Pós-incremento	Retorna <code>\$a</code> , e então incrementa <code>\$a</code> em um.
<code>--\$a</code>	Pré-decremento	Decrementa <code>\$a</code> em um, e então retorna <code>\$a</code> .
<code>\$a--</code>	Pós-decremento	Retorna <code>\$a</code> , e então decrementa <code>\$a</code> em um.
<code>-\$a</code>	Inverter o sinal	Inverte o sinal de <code>\$a</code> e retorna <code>\$a</code> .

OPERADORES

- <http://www.aprenderphp.com.br/artigo/operadores-no-php>

IF

```
if (expr)  
    statement
```

- http://www.php.net/manual/pt_BR/control-structures.if.php

IF

```
$nota = 6;

if ($nota >= 7) {

    print "Parabéns, você foi aprovado !";
    print "<P>";

}

print "Sua nota final foi : ";
print $nota;
print "<p>";
```

IF-ELSE

```
$nota = 6;

if ($nota >= 7) {

    print "Parabéns, você foi aprovado !";
    print "<P>";

} else {

    print "Você terá que fazer aulas de recuperação.";
    print "<P>";

}

print "Sua nota final foi : ";
print $nota;
print "<p>";
```

IF-ELSE

```
$nota = 6;
```

```
if ($nota >= 7) {
```

```
    print "Parabéns, você foi aprovado !";  
    print "<P>";
```

```
} else {
```

```
    print "Você terá que fazer aulas de recuperação.";   
    print "<P>";
```

```
}
```

```
print "Sua nota final foi : ";  
print $nota;  
print "<p>";
```

SWITCH

```
// Declarando variáveis
$feriado = "carnaval";

// Determinando através do switch que mensagem enviar
switch ($feriado){
    case "carnaval":
        print "feliz carnaval !";
        break;
    case "natal":
        print "feliz natal.";
        break;
    default:
        print "tenha um bom dia.";
}
```


WHILE

```
$contador = 1;

while ($contador <= 6) {

    print "<H$contador>Hello World !</H$contador>\n";

    $contador++;

}
```


WHILE

```
$contador = 0;

while ($contador < 20) {

    $contador++;

    if ($contador == 5 || $contador == 10 || $contador == 15) {

        continue;

    }

    print "contador = $contador<br>";

}
```

DO WHILE

```
$contador = 1;

do{

    print "<H$contador>Hello World !</H$contador>\n";

    $contador++;

} while ($contador<=6) ;
```

DO WHILE

```
$contador = 1;
```

```
do{
```

```
    print "<H$contador>Hello World !</H$contador>\n";
```

```
    $contador++;
```

```
} while ($contador<=6) ;
```

DO WHILE

Linha 1 Coluna 1	Linha 1 Coluna 2	Linha 1 Coluna 3	Linha 1 Coluna 4	Linha 1 Coluna 5
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3	Linha 2 Coluna 4	Linha 2 Coluna 5
Linha 3 Coluna 1	Linha 3 Coluna 2	Linha 3 Coluna 3	Linha 3 Coluna 4	Linha 3 Coluna 5
Linha 4 Coluna 1	Linha 4 Coluna 2	Linha 4 Coluna 3	Linha 4 Coluna 4	Linha 4 Coluna 5
Linha 5 Coluna 1	Linha 5 Coluna 2	Linha 5 Coluna 3	Linha 5 Coluna 4	Linha 5 Coluna 5
Linha 6 Coluna 1	Linha 6 Coluna 2	Linha 6 Coluna 3	Linha 6 Coluna 4	Linha 6 Coluna 5
Linha 7 Coluna 1	Linha 7 Coluna 2	Linha 7 Coluna 3	Linha 7 Coluna 4	Linha 7 Coluna 5
Linha 8 Coluna 1	Linha 8 Coluna 2	Linha 8 Coluna 3	Linha 8 Coluna 4	Linha 8 Coluna 5
Linha 9 Coluna 1	Linha 9 Coluna 2	Linha 9 Coluna 3	Linha 9 Coluna 4	Linha 9 Coluna 5
Linha 10 Coluna 1	Linha 10 Coluna 2	Linha 10 Coluna 3	Linha 10 Coluna 4	Linha 10 Coluna 5

DO WHILE

```
<table border=1 cellspacing=5 width=100%>

<?php
    do{
        ?>

        <tr>

            <?php
            |
            $coluna = 1;

            while($coluna<=5){

                print "<td align=center valign=middle>Linha $linha Coluna $coluna</td>";

                $coluna++;

            }

            ?>
        </tr>

        <?php

        $linha++;

    }while($linha <= 10);

?>

</table>
```

DO WHILE

```
<table border=1 cellspacing=5 width=100%>

<?php
do{
    ?>
    <tr>

    <?php
    $coluna = 1;

    while($coluna<=5){

        print "<td align=center valign=middle>Linha $linha Coluna $coluna</td>";

        $coluna++;

    }

    ?>
    </tr>

    <?php
    $linha++;

}while($linha <= 10);

?>

</table>
```

FOR

```
for ($contador=1; $contador <=6; $contador++){  
    print "<h$contador>Utilizando o comando for</h$contador>\n";  
    |  
}
```

```
for ($contador = 6; $contador > 0; $contador--){  
    print "<h$contador>Utilizando o comando for</h$contador>\n";  
}
```

BREAK

```
for ($contador = 1; $contador <= 10; $contador++) {  
    print "contador = $contador<br>";  
    if ($contador = 5) {  
        break;  
    }  
}
```


CONTINUE

```
$contador = 0;

while ($contador < 20) {

    $contador++;

    if ($contador == 5 || $contador == 10 || $contador == 15) {

        continue;

    }

    print "contador = $contador<br>";

}
```

FOREACH

```
1  <?php
2  $linguagens = array("PHP", "Java", "Python", "C++", "C", "Smaltalk");
3
4  foreach($linguagens as $nomeLinguagem) {
5      print "$nomeLinguagem <br>";
6  }
7  ?>
```

CONSIDERAÇÕES FINAIS

Dúvidas?





DEFINIR PROJETO

<http://www.freewebsitetemplates.com/>

REFERÊNCIAS

- Manual PHP, https://www.php.net/manual/pt_BR/index.php.