INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Programação Web

Aula 04 – Manipulação e Tipos de Dados em PHP

Prof. Pedro Baesse
pedro.baesse@ifrn.edu.br

Roteiro

- Tipo de Dados
- Constantes
- Manipulação de dados
- Variáveis
- Operadores

Tipos de Dados

- Variáveis podem guardar diversos tipos de dados
  - Booleano
  - Numérico
  - Array
  - Objeto
  - Recurso
  - Misto
  - Null

Tipo Booleano

- Expressa um valor lógico que pode ser verdadeiro ou falso
  - TRUE ou FALSE podem ser atribuídos

```php
<?php

        //Declara variável com valor TRUE
        $exibirFrase = TRUE;

        //Testa se $exibirFrase é verdadeiro (TRUE)

        if($exibirFrase)
        {
                echo "A variável booleana é VERDADEIRA!";

        }

?>
```

Tipo Booleano

- Mais um exemplo:

```php
<?php

    //Declara variável numérica
    $umidade = 91;

    //Testa se $umidade maior que 90. Retorna um boolean
    $vaiChover = ($umidade > 90);

    //Testa se $vaiChover é verdadeiro
    if($vaiChover)
    {

        echo "Vai chover com toda certeza absoluta da terra!";

    }

?>
```

Tipo Booleano

- Também são considerados valores falsos:
  - Inteiro 0
  - Ponto Flutuante 0.0
  - Uma string vazia “” ou “0”
  - Um array vazio
  - Um objeto sem elementos
  - Tipo NULL
- Vamos Testar??

Tipo Numérico

- Os números podem ser especificados
  - Decimal(base 10)
  - Hexadecimal (base 16)
  - Octal(base 8)
  - Opcionalmente usando sinal (- ou +)

Tipo Numérico

- Os números podem ser especificados
  - Decimal(base 10)
  - Hexadecimal (base 16)
  - Octal(base 8)
  - Opcionalmente usando sinal (- ou +)

Tipo Numérico

```php
<?php
        //Número Inteiro
        $num = 537;
        echo $num, " = Número Inteiro <br><br>";

        //Número Negativo
        $num = -13;
        echo $num, " = Número Negativo <br><br>";

        //Número Octal (Equivalente a 83 em decimal)
        $num = 0123;
        echo $num, " = Número Octal <br><br>";

        //Número Hexadecimal (Equivalente a 26 em decimal)
        $num = 0x1A;
        echo $num, " = Número Hexadecimal <br><br>";

        //Ponto Flutuante
        $num = 1.234;
        echo $num, " = Ponto Flutuante <br><br>";

        //Notação Científica
        $num = 4e23          ;
        echo $num, " = Notação Científica <br><br>";
?>
```

Tipo String

- Cadeia de caracteres alfanuméricos
  - Podem ser usados aspas simples (‘’), aspas duplas (“”), ou aspas invertidas (’) porém existem diferenças (abordado posteriormente)

```php
<?php

        $string = 'Esse texto foi armazenado usando um variável do tipo string com aspas simples <BR>';
        echo $string;

        $string = "Essa string foi armazenada usando aspas duplas";
        echo $string;

?>
```

Manipulação de Dados – Strings

- O PHP as strings de maneira de acordo com o delimitador usado
- Aspas simples (‘’)
  - Uso bastante similar ao das aspas duplas (“”)
  - Para usar uma aspas simples como string usar o caractere de controle contra barra (\ ). Indica ao PHP que a próxima aspa deve ser lida como texto
  - Para adicionar uma quebra de linha bastar usar o ENTER no meio do texto. Em caso de formato HTML usar a tag

Manipulação de Dados – Strings

- Aspas Duplas (“”)
  - Principal diferença entre as aspas simples (‘’) é interpolação de variáveis
  - Também é o possível usar as aspas simples (‘’) sem o caractere de controle

Tipo Vetor (Array)

- Uma lista de valores armazenados na memória
  - Podem ser de diferentes tipo (números, strings, objetos)
  - Acessados a qualquer momentos através de chave, índice associado
    - Strings ou números
    - Identificado por [ ]
  - Pode crescer dinamicamente com adição de novos itens

Tipo Vetor (Array)

```php
<?php

        //Criação do Array
        $carros = array('Palio','Corsa','Gol', "siena" =>"Siena");

        echo $carros[1] . "<br>";  //Resultado Corsa
        echo $carros["siena"] . "<br>";  //Resultado Siena

?>
```

- Chaves dos Arrays inicia-se 0!

Tipo Vetor (Array)

```php
//Adicionando novos elementos
$carros[4] = "Clio";
$carros[11] = "Versa";
$carros[ ] = "Sandero";

echo $carros[4] . "<br>";  //Resultado Clio
echo $carros[5] . "<br>";  //Sem Resultado
echo $carros[11] . "<br>";  //Resultado Versa
echo $carros[12] . "<br>";  //Resultado Sandero
```

- Usando o colchetes sem índice ([ ]) o PHP procurará o último índice utilizado e o incrementará

Tipo Vetor (Array)

```php
$carros["popular"] = "Fusca";
$carros["quantidade"] = 7;

echo $carros["popular"] . "<br>";  //Resultado Fusca
echo $carros["quantidade"] . " carros no array <br>";  //Resultado 7
```

- Também pode ser usada um string como índice, neste caso chamada de chave associativa

- É possível utilizar arrays com 2 tipos de índice sem ocorrer erro

Matrizes

- São arrays multidimensionais
- Também possuem um único identificador mas formado por um ou mais índices

```
$brasil["RN"][1] = "Natal";
$brasil["RN"][2] = "Caicó";
$brasil["RN"][3] = "Pau do Ferros";
$brasil["PB"][1] = "João Pessoa";
$brasil["PB"][2] = "Campina Grande";
$brasil["MG"][1] = "Belo Horizonte";
$brasil["MG"][2] = "Araguari";
$brasil["GO"][1] = "Goiânia";

echo $brasil["RN"][1] . "<br>"; //Resultado Natal
echo $brasil["MG"][2] . "<br>"; //Resultado Araguari
echo $brasil["GO"][1] . "<br>"; //Resultado Goiânia
```

Tipo Objeto

- Um objeto é uma entidade definida por:
  - Propriedades (atributos)
  - Métodos (ações)
- Para criar um objeto usa-se o operador **new**

Tipo Objeto

- Um objeto é uma entidade definida por:
  - Propriedades (atributos)
  - Métodos (ações)
- Para criar um objeto usa-se o operador **new**

Tipo Objeto

```php
class Computador
{
        var $cpu;

        function ligar()
        {
                echo "Ligando computador a {$this->cpu}...";
        }
}

$obj = new Computador;
$obj->cpu = "500Mhz";
$obj->ligar();
```

Tipo Recurso

- Recurso (resource) é uma referência recurso externo
  - Criados e utilizados por funções especiais
    - A função **mysql_connect()** retorna, ao se conectar a um banco de dados, uma referência do tipo recurso
    - resource mysql_connect(...)

Tipo Misto

- Representa múltiplos (não necessariamente todos) tipos de dados de um mesmo parâmetro
  - Um parâmetro do tipo mixed indica que a função aceita diversos tipos de dados
  - Um exemplo é a função get typpe(), que obtém o tipo do parâmetro passado (pode ser integer, string, array, objeto...)

Tipo Misto

- string gettype(mixed var)
  - Pode retornar
    - Booleano
    - Integer
    - Double
    - String
    - Array
    - Object
    - Resource
    - NULL

Tipo NULL

- Indica que a variável não possui nenhum valor
- NULL é o único valor possível

Constantes

- Um valor que não sofre modificações durante a execução do programa
- Representada por um identificador, como as variáveis, porém só aceita valores escalares (booleano, inteiro, ponto flutuante, string)
- Valor escalar é aquele que não é composto por outros valores, como array ou objetos

Constantes

- A nomenclatura segue as mesmas regras das variáveis
- Não utiliza o cifrão ($)
- Geralmente escrita em MAIÚSCULO
- MAXIMO_CLIENTE

Constantes

- A função define() defini o valor da constante
  - Bool define (string *nome,* misto *valor,* bool case_insentive)

```
//Define o valor máximo de clientes
define("MAXIMO_CLIENTE",100);

echo  MAXIMO_CLIENTE;
```

- O que acontece caso se defina a constante novamente??

Constantes – Pré-definidas em PHP

- TRUE
  - Valor Verdadeiro
- FALSE
  - Valor Falso
- __FILE__
  - Contém o nome do script que está sendo executado
- __LINE__
  - Contém o nome da linha do script que está sendo executado
- PHP_VERSION
  - Contém a versão corrente do PHP

Constantes – Pré-definidas em PHP

- PHP_OS
  - Nome do sistema operacional no qual o PHP está rodando
- E_ERROR
  - Exibe um erro ocorrido em um script. A execução é interrompida
- E_WARNING
  - Exibe uma mensagem de aviso do PHP. A execução continua
- E_PARSE
  - Exibe um erro de sintaxe. A execução é interrompida
- E_NOTICE
  - Mostra que ocorreu algo, mas não necessariamente um erro. A execução não pára

Escopo das Variáveis – Função

- Funciona exclusivamente dentro de um função

```
$num = 5000;
        function escopo_funcao()
        {
                $num+=999;

                echo $num . "<br>";
        }

echo $num . "<br>";

escopo_funcao();
```

Escopo das Variáveis – Global

- Pode ser usada no programa principal e em uma função
- Definida como global no início da função
- Utilizar o array predefinido $GLOBALS, que usa os nomes das variáveis como índice

Escopo das Variáveis – Global

```php
$num = 5000;

function escopo_global()
{
        global $num;

        $num+=999;

        echo $num . "<br>";
}

echo $num . "<br>";

escopo_global();
```

- Uso da variável global

Escopo das Variáveis – Global

```php
$num = 5000;

function escopo_global()
{
        global $num;

        $num+=999;

        echo $num . "<br>";
}

echo $num . "<br>";

escopo_global();
```

- Uso da variável global

Escopo das Variáveis – $GLOBALS

```php
$num = 5000;

function  escopo_global_array()
{
        echo $GLOBALS["num"] . "<br>";

        $GLOBALS["num"]++;
}

escopo_global_array();

echo $num . "<br>";
```

- Uso do Array $Globals

Conversão de Variáveis

- O PHP soma normalmente duas variáveis do tipo numérico
- Se houver números e textos em uma String o PHP somará apenas os números

Conversão de Variáveis

- O PHP soma normalmente duas variáveis do tipo numérico
- Se houver números e textos em uma String o PHP somará apenas os números

Conversão de Variáveis

- O PHP soma normalmente duas variáveis do tipo numérico
- Se houver números e textos em uma String o PHP somará apenas os números

Conversão de Variáveis

```php
$string = "5";

$numero = 3;

$texto = "3 vezes campeão";

echo $string+$numero+$texto;
```

Conversão de Variáveis

- Em alguns momentos é necessário converter manualmente para realizar alguns cálculos
  - Converte o número imediatamente depois dele

- (int),(integer)
  - Converte para inteiro
- (real),(float),(double)
  - Converte para ponto flutuante
- (string)
  - Converte em string
- (array)
  - Converte em array (vetor)
- (object)
  - Converte em objeto

Conversão de Variáveis

```php
$x = 50;
$y = 2.35;

$soma = (int) $y + $x;

echo  $soma . "<br>";

$soma = (float)($y + $x);

echo  $soma . "<br>";
```

Conversão de Variáveis

- Vamos criar várias variáveis, converte-las e verificar seu tipo antes e depois usando a função gettype()

Interpolação de variáveis

- Escrever o valor de uma ou mais variáveis dentro da *string* que será mostrada na tela ou atribuida a outra variável

```
$cidade = "Pau dos Ferros";

$regiao = "Alto Oeste";

$estado = "Rio Grande do Norte";

echo "A cidade de $cidade fica no $regiao do $estado";
```

Interpolação de variáveis

- Tomar cuidado ao escrever uma variável ao lado de um ou mais caracteres. Isso pode causar um erro!

```
$p = "Penta";
echo "O Brasil é $pcampeão";
```

- Solução

```
echo "O Brasil é ${p}campeão";

echo "O Brasil é". $p ." campeão";
```

Variáveis em tempo execução

- É útil criar variáveis dinamicamente, ou seja, em tempo de execução do programa

```
$texto = "Natal";

$futuro_identificador = "cidade";

$$futuro_identificador = $texto;

echo "$cidade é no litoral do RN“
```

- $cidade foi criada dinamicamente a partir do valor que continha e o uso de $$

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA RIO GRANDE DO NORTE

Operadores

- Informa ao PHP o que deve ser executado.
  - Ex: Atribuir um valor a uma variável, realizar operações aritméticas, comparações de valores, testar...

- Tipos
  - Operadores Aritméticos
  - Operadores Binários
  - Operadores de Comparação
  - Operadores de atribuição
  - Operadores lógicos
  - Operador ternário

Operadores

- Informa ao PHP o que deve ser executado.
  - Ex: Atribuir um valor a uma variável, realizar operações aritméticas, comparações de valores, testar...

- Tipos
  - Operadores Aritméticos
  - Operadores Binários
  - Operadores de Comparação
  - Operadores de atribuição
  - Operadores lógicos
  - Operador ternário

Operadores Aritméticos

Exemplo	Nome	Resultado
-$a	Negação	Oposto de $a.
$a + $b	Adição	Soma de $a e $b.
$a - $b	Subtração	Diferença entre $a e $b.
$a * $b	Multiplicação	Produto de $a e $b.
$a / $b	Divisão	Quociente de $a por $b.
$a % $b	Módulo	Resto de $a dividido por $b.

Operadores Aritméticos

Exemplo	Nome	Efeito
-$a	Troca Sinal	Trocar o sinal de $a
++$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.

- Faz em um linha de código o que levaria duas ou mais linhas caso não fossem usadas. Assim o código fica mais simples e claro

Operadores Aritméticos

```php
echo "<h3>Pós-incremento</h3>";
$a = 5;
echo "Deve ser 5: " . $a++ . "<br />\n";
echo "Deve ser 6: " . $a . "<br />\n";

echo "<h3>Pré-incremento</h3>";
$a = 5;
echo "Deve ser 6: " . ++$a . "<br />\n";
echo "Deve ser 6: " . $a . "<br />\n";

echo "<h3>Pós-decremento</h3>";
$a = 5;
echo "Deve ser 5: " . $a-- . "<br />\n";
echo "Deve ser 4: " . $a . "<br />\n";

echo "<h3>Pré-decremento</h3>";
$a = 5;
echo "Deve ser 4: " . --$a . "<br />\n";
echo "Deve ser 4: " . $a . "<br />\n";
```

Operadores Binários

- Trabalham diretamente com os bits

Exemplo	Nome	Resultado
$a & $b	E	Os bits que estão ativos tanto em $a quanto em $b são ativados.
$a | $b	OU	Os bits que estão ativos em $a ou em $b são ativados.
$a ^ $b	XOR	Os bits que estão ativos em $a ou em $b, mas não em ambos, são ativados.
~ $a	NÃO	Os bits que estão ativos em $a não são ativados, e vice-versa.
$a << $b	Deslocamento à esquerda	Desloca os bits de $a $b passos para a esquerda (cada passo significa "multiplica por dois")
$a >> $b	Deslocamento à direita	Desloca os bits de $a $b passos para a direita (cada passo significa "divide por dois")

Operadores de Atribuição

- Usado para colocar um valor em uma outra variável

Exemplo	Efeito
$a = $b	$a recebe o valor de $b
$a += $b	Equivalente a $a = $a + $b
$a -= $b	Equivalente a $a = $a - $b
$a *= $b	Equivalente a $a = $a * $b
$a /= $b	Equivalente a $a = $a / $b
$a %= $b	Equivalente a $a = $a % $b
$a .= $b	Concatenação: equivalente a $a = $a . $b
$a &= $b	Equivalente a $a = $a & $b
$a |= $b	Equivalente a $a = $a | $b
$a ^= $b	Equivalente a $a = $a ^ $b
$a <<= $b	Equivalente a $a = $a << $b
$a >>= $b	Equivalente a $a = $a >> $b

Exercício

```php
$soma = ($valor1 = 4) + 5; // $soma é igual a 9 agora e $valor1 foi configurado como 4.
$valor2 = 20;
$valor3 = 30;

$soma += 1; //$soma fica com 10
$soma += $valor2; //$soma fica com 10+20=30
$soma *= $valor3; //$soma fica com 30*30=900
$soma %= 100;

$saudacao = "Bom ";
$saudacao .= "Dia!"; // configura $saudacao para "Bom Dia!", como em $saudacao = $saudacao . "Dia!";

echo $saudacao . " Hoje vai ter " . $soma . " % tristeza!";
```

Operadores de Comparação

- Também chamados de condicionais. Executam comparações entre duas variáveis, ou uma variável e um texto ou uma variável e um número

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 (introduzido no PHP4).
$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.

Exercício

```
var_dump(0 == "a"); // 0 == 0 -> true
var_dump("1" == "01"); // 1 == 1 -> true
var_dump("1" == "1e0"); // 1 == 1 -> true
```

- Uma String é convertida para número quando comparada com um número

Operadores Lógicos

- São aqueles que retornam o valor verdadeiro ou falso

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

Operadores Lógicos

- Precedência dos Operadores Lógicos
  - II e &&
  - AND e OR

```php
// foo() nunca será chamada como estes operadores são short-circuit
$a = (false  && foo());
$b = (true  || foo());
$c = (false  and foo());
$d = (true  or  foo());

// "||" tem maior precedência que "or"
$e = false || true; // $e vai receber (false || true) que é verdadeiro
$f = false  or true; // $f vai receber falso
var_dump($e, $f);

// "&&" tem maior precedência que "and"
$g = true && false; // $g vai receber (true && false) que é falso
$h = true and false; // $h vai receber verdadeiro
var_dump($g, $h);
```

Operadores Lógicos

- Muito usados para verificação de campos obrigatórios em um formulário

```
if(empty($nome) OR empty($email) OR empty($cpf)){

        echo "Você deve preencher os campos nome, e-mail e CPF!";
        exit;
}
```

- O método empty() retorna verdadeiro caso a variável for vazia

- É testado se os campos nome, email e cpf são vazios

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Operadores Lógicos

- Operador AND (E)

EXP1	EXP2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

- Operador OR (OU)

EXP1	EXP2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

Operadores Lógicos

- Operador XOR (OU Exclusivo)

EXP1	EXP2	Resultado
V	V	F
V	F	V
F	V	V
F	F	F

- Operador ! (NOT)

EXP1	Resultado
V	F
F	V

Operador Ternário

- Uma forma abreviada de usar o comando if, que será visto adiante.
- Sintaxe cond ? exp1 : exp2
- Se a condição for verdadeira a exp1 é executada e se for falsa a exp2 é executada

Operador Ternário

- Uma forma abreviada de usar o comando if, que será visto adiante.
- Sintaxe cond ? exp1 : exp2
- Se a condição for verdadeira a exp1 é executada e se for falsa a exp2 é executada

Operador Ternário

```php
$nota = 8;

if($nota>=6){
echo "Você passou por média!";
}else{
echo "Você ficou de recuperação...!";
}

echo "<BR><BR> Mesma verificação usano o operador ternário
        <BR><BR>";

echo ($nota>=6) ? "Você passou por média!" : "Você ficou de
                                    recuperação...!";
```

Precedência de Operadores

Operador	Informação adicional
++ --	incremento/decremento
!	lógico
* / %	aritmético
+ - .	aritmético e string
<< >>	Bit-a-bit
< <= > >= <>	comparação
== != === !==	comparação
&	Bit-a-bit e referências
^	Bit-a-bit
|	Bit-a-bit
&&	lógico
||	lógico
? :	ternário
= += -= *= /= .= %= &= |= ^= <<= >>=	atribuição
and	lógico
xor	lógico
Or	lógico

Dúvidas

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Referências

- PHP Manual:
  - http://www.php.net/manual/pt_BR/index.php
- Desenvolvendo Websites com PHP
  - De Juliano Niederauer