

## TEMA 2 - Variáveis e Tipos de dados

### Habilidades

- Compreender a função de uma variável e seu tratamento no contexto do PHP.
- Compreender os conceitos de tipagem de variáveis e seu funcionamento no código.

O PHP que vamos estudar como já mencionado, vai ser a versão 6.0, na sintaxe procedural, toda linguagem de programação tem a sua documentação diante disso vamos falar sobre a documentação do PHP, que é muito importante para nós programadores. Para acessar a documentação do PHP vamos entrar nesse link <https://www.php.net/>.

Nele vamos encontrar as informações atuais, dicas de novos comandos, expressões, funções e até API, a dica é que toda vez que surgir uma dúvida acesse esse site e procure sobre o que você precisa.

Na parte superior da tela à direita tem a caixa de procura, basta você digitar o comando que você tem dúvida que o site vai lhe direcionar para uma ajuda objetiva e rápida.

### 2.1 Sintaxe do PHP

Como vimos no conteúdo passado entendemos que a sintaxe é a organização de qualquer linguagem, agora vamos nos aprofundar mais, sobre a sintaxe do PHP. Para comentar **no PHP usamos // para uma única linha e /\* comentário em bloco \*/** Exemplo de comentário:

```
// Este é um comentário de uma linha em PHP.
```

```
/*  
    Este é um comentário  
    de várias linhas em PHP.  
    Pode ocupar quantas linhas forem necessárias.  
*/
```

Agora de uma maneira mais clara, estamos usando php com as tags de html, junto do seu escopo;

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Olá Mundo em PHP</title>  
    <style>  
        body {  
            font-family: Arial, sans-serif;  
            background-color: #f0f0f0;  
            margin: 0;  
            padding: 0;  
            display: flex;  
            justify-content: center;  
            align-items: center;  
            height: 100vh;  
        }  
        .container {
```

```

        text-align: center;
    }
    h1 {
        color: #333;
        font-size: 3rem;
        margin-bottom: 20px;
    }
    p {
        color: #666;
        font-size: 1.2rem;
    }
</style>
</head>
<body>
    <div class="container">
        <?php
            $mensagem = "Olá Mundo! oppppa";
            echo "<h1>$mensagem</h1>";
        ?>
        <p>Este é um exemplo de página PHP estilizada com HTML e CSS.</p>
    </div>
</body>
</html>

```

## 2.2 Tipos de variáveis

Variável como vimos nas aulas de lógica é um local temporário, para guardar dados de vários tipos, exemplo inteiro, decimal, texto e lógico. Agora vamos falar desses tipos só usando o PHP. Os tipos de dados usados no PHP, são básicos e compostos. Para declarar uma variável no PHP, temos que usar o \$ do dinheiro sempre na frente do nome da variável, lembrando que seguimos os exemplos básicos para declarar qualquer tipo de variável.

Vamos falar sobre os tipos de variável de dados básicos;

### String

String é um tipo de variável que só suporta texto, podendo ser uma letra uma palavra ou até um texto, conhecida também como caracter. Só que aqui no PHP é chamada string.

```

<?php
// Declaração e atribuição de uma variável de string.
$nome = "João";

// Concatenação de strings usando o operador ".".
$saudacao = "Olá, " . $nome . "! Bem-vindo!";

// Interpolação de strings dentro de uma string delimitada por aspas duplas.
$saudacao_interpolada = "Olá, $nome! Bem-vindo!";
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">

```

```

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Saudação</title>
</head>
<body>
    <h1>Saudação</h1>
    <p><?php echo $saudacao; ?></p>
    <p><?php echo $saudacao_interpolada; ?></p>
</body>
</html>

```

## Inteiro

Inteiro ou também chamado de int é um tipo de variável que só recebe números inteiros, não aceita números quebrados exemplo {... -4 -3 -2 -1 0 1 2 3 4 ...}.

```

<?php
// Declaração e atribuição de uma variável do tipo inteiro.
$idade = 25;

```

```

// Mensagem baseada na idade.
if ($idade >= 18) {
    $mensagem = "Você é maior de idade.";
} else {
    $mensagem = "Você é menor de idade.";
}
?>

```

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Verificação de Idade</title>
</head>
<body>
    <h1>Verificação de Idade</h1>
    <p><?php echo $mensagem; ?></p>
    <p>Sua idade: <?php echo $idade; ?></p>
</body>
</html>

```

## Float

Float é um tipo de variável que recebe números inteiros e quebrados, conhecido como decimal. Também falado ponto flutuante:

```

<?php
// Declaração e atribuição de uma variável do tipo float.
$peso = 68.5;

// Mensagem baseada no peso.
if ($peso >= 70.0) {
    $mensagem = "Seu peso está acima do recomendado.";
}

```

```

} elseif ($peso >= 50.0 && $peso < 70.0) {
    $mensagem = "Seu peso está dentro do recomendado.";
} else {
    $mensagem = "Seu peso está abaixo do recomendado.";
}
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Verificação de Peso</title>
</head>
<body>
    <h1>Verificação de Peso</h1>
    <p><?php echo $mensagem; ?></p>
    <p>Seu peso: <?php echo $peso; ?> kg</p>
</body>
</html>

```

## Boolean

Booleano ou lógico é um tipo de variável que recebe somente dois valores verdadeiro (true) e Falso (false), também 1 para verdadeiro e 0 para falso.

```

<?php
// Declaração e atribuição de uma variável do tipo booleano.
$temConta = true;

```

```

// Mensagem baseada na variável booleana.
if ($temConta) {
    $mensagem = "Você possui uma conta cadastrada.";
} else {
    $mensagem = "Você não possui uma conta cadastrada.";
}
?>

```

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Verificação de Conta</title>
</head>
<body>
    <h1>Verificação de Conta</h1>
    <p><?php echo $mensagem; ?></p>
</body>
</html>

```

## Tipos de variáveis compostas

## Array

Um array é um tipo de variável que pode armazenar mais de um valor ao mesmo tempo, possui um identificador que chamamos de índice.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Array com HTML</title>
</head>
<body>
  <h1>Lista de Frutas</h1>

  <?php
  // Criando um array de frutas.
  $frutas = array("Maçã", "Banana", "Laranja", "Morango", "Abacaxi");

  // Acessando elementos do array usando índices.
  echo "<p>A primeira fruta é: " . $frutas[0] . "</p>";
  echo "<p>A segunda fruta é: " . $frutas[1] . "</p>";

  // Adicionando um novo elemento ao array.
  $frutas[] = "Uva";

  // Exibindo o array atualizado com print_r.
  echo "<p>Lista de frutas atualizada: </p>";
  echo "<pre>";
  print_r($frutas);
  echo "</pre>";
  ?>
</body>
</html>
```

O elemento <pre> em HTML (preformatado) é útil quando você deseja exibir texto formatado exatamente como está no código-fonte, respeitando espaços em branco e quebras de linha. Isso é particularmente útil ao exibir conteúdo de arrays usando a função print\_r(), pois o <pre> preserva a formatação do array e o torna mais legível.

Sem o uso do <pre>, o texto gerado pela função print\_r() seria exibido em uma única linha, o que poderia dificultar a leitura e compreensão, especialmente em arrays com múltiplos elementos.

## Objeto

Objeto é um tipo de variável que recebe valores de classes, usado no paradigma POO, que vamos estudar em outro módulo, segue um exemplo de como instanciar uma variável.

Vamos supor que temos uma classe chamada Pessoa com os atributos nome e idade, e um método apresentar() que exibe uma mensagem de apresentação da pessoa:

```
<?php
// Definição da classe Pessoa
```

```

class Pessoa {
    public $nome;
    public $idade;

    public function apresentar() {
        return "Olá! Meu nome é {$this->nome} e tenho {$this->idade} anos.";
    }
}

```

```

// Instanciando um objeto da classe Pessoa

```

```

$ Pessoa = new Pessoa();
$ Pessoa->nome = "João";
$ Pessoa->idade = 30;
?>

```

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Exemplo de Objeto com HTML</title>
</head>
<body>
    <h1>Apresentação da Pessoa</h1>
    <p><?php echo $ Pessoa->apresentar(); ?></p>
</body>
</html>

```

Neste exemplo, definimos a classe Pessoa com os atributos nome e idade, e o método apresentar() que retorna uma mensagem com esses atributos.

Em seguida, instanciamos um objeto da classe Pessoa usando o operador new, atribuímos valores aos seus atributos e, por fim, utilizamos o método apresentar() para exibir uma mensagem de apresentação no HTML.

## 2.2 Comando de exibição print e echo

### Print

Comando para exibir uma string na tela, por não ser uma função real não precisa usar parênteses com ele.

### Echo

Comando echo tem a mesma função do print, sendo o mais usado quando usamos o PHP, sendo assim ele também não é uma função real, não precisa usar parênteses.

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Exemplo de Print e Echo com HTML</title>

```

```

</head>
<body>
  <?php
    // Definição de uma variável
    $mensagem = "Olá, mundo!";?>
    Usando print para imprimir a mensagem:
    <?php print $mensagem;?><br>
    Usando echo para imprimir a mesma mensagem:
    <?php echo $mensagem;?>
  </body>
</html>

```

## 2.3 Tipos de dados com funções ambiente

Funções são comandos feitos pelo desenvolvedor, criando uma espécie de rotinas, ao chamar essas rotinas será executado o conjunto todo. Podemos também ter uma função da própria linguagem que chamamos de função de ambiente.

### Exemplo de função ambiente:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de PHP com HTML</title>
</head>
<body>
  <h1>Manipulação de Strings e Data</h1>

  <?php
    // String de exemplo
    $texto = "Oi, Mundo!";

    // Exibindo o comprimento da string
    $comprimento = strlen($texto);
    echo "<p>Comprimento da string: $comprimento</p>";

    // Convertendo a string para minúsculas
    $minusculas = strtolower($texto);
    echo "<p>String em minúsculas: $minusculas</p>";

    // Convertendo a string para maiúsculas
    $maiusculas = strtoupper($texto);
    echo "<p>String em maiúsculas: $maiusculas</p>";

    // Substituindo parte da string
    $nova_string = str_replace("Mundo", "Planeta", $texto);
    echo "<p>String com substituição: $nova_string</p>";

  ?>
</body>
</html>

```

## 2.4 Concatenação

Este elemento é muito usado em qualquer linguagem de programação, sua finalidade é unir um conjunto de string, para exibir na tela. Usamos o ponto (.) para concatenar no PHP. Segue um exemplo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Concatenação de Strings em PHP</title>
</head>
<body>
  <?php
    // Definindo as variáveis
    $aluno = "João";
    $curso = "Programação";

    // Concatenando as strings para formar uma mensagem
    $mensagem = "O aluno " . $aluno . " está matriculado no curso de " . $curso . ".";

    // Exibindo a mensagem
    echo "<p>$mensagem</p>";
  ?>
</body>
</html>
```

## 2.5 Conversão de dados

As variáveis declaradas no PHP, não tem logo na sua criação um tipo de dados específico ela vai se adaptar ou virar um tipo de dados conforme os dados que se atribui a ela, isso vale também para variáveis que vem derivadas de outras, chamamos de coerção.

Veja alguns exemplos:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Manipulação de Tipos de Dados em PHP</title>
</head>
<body>
  <h2>Manipulação de Tipos de Dados em PHP</h2>

  <?php
    // Inicializando uma variável como inteiro
    $valor = 10;

    // Exibindo informações sobre a variável antes da conversão
    echo "<p><strong>Antes da Conversão:</strong></p>";
    echo "<p>Valor: $valor</p>";
    echo "<p>Tipo de Dados: " . gettype($valor) . "</p>";
```



```
// Convertendo a variável para string usando settype
settype($valor, "string");

// Exibindo informações sobre a variável depois da conversão
echo "<p><strong>Depois da Conversão:</strong></p>";
echo "<p>Valor: $valor</p>";
echo "<p>Tipo de Dados: " . gettype($valor) . "</p>";

// Exibindo informações detalhadas sobre a variável usando var_dump
echo "<p><strong>Informações Detalhadas:</strong></p>";
var_dump($valor);
?>
</body>
</html>
```

#### Neste exemplo:

Inicializamos a variável \$valor como um inteiro com o valor 10.  
 Usamos gettype para obter o tipo de dados da variável antes e depois da conversão.  
 Usamos settype para converter a variável para string.  
 Finalmente, usamos var\_dump para exibir informações detalhadas sobre a variável.

Convertendo o resultado float para um valor inteiro.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Divisão Inteira em PHP</title>
</head>
<body>
  <h2>Divisão Inteira em PHP</h2>
  <?php
  // Calculando a divisão entre 5 e 3
  $resultado = 5 / 3;
  $resultado_formatado = number_format($resultado, 2);
  echo "<p>Resultado da divisão entre 5 e 3: $resultado_formatado</p>";

  // Convertendo o resultado para inteiro
  settype($resultado, "integer");

  // Exibindo o resultado da divisão inteira
  echo "<p>Resultado da divisão inteira entre 5 e 3: $resultado</p>";
  ?>
</body>
</html>
```

Neste exemplo, a divisão entre 5 e 3 resultará em um número decimal. Em seguida, usamos settype para converter esse resultado para um valor inteiro, resultando em uma divisão inteira.

## 2.6 Tipos de erros

Erros de lógica é o mais comum entre os desenvolvedores, quando isso acontece é quando você e a máquina não consegue ter o mesmo pensamento, nesse caso é bom descansar a mente e depois voltar a programar.

Outro erro comum é o de sintaxe, quando o programador digita errado na tela, nesse caso vai aparecer essa mensagem.

**Parse error:** syntax error, unexpected '\$idade' (T\_VARIABLE), expecting ';' or ',' in **C:\xampp\htdocs\aula02\index.php** on line **16**

Veja que "Parse error: Syntax", vai ajudar você entender que foi digitado um comando errado que não existe, no final dessa mensagem está aparecendo "line 16" significa que o problema está na linha 16.

Outros erros que vão aparecer na sua aplicação.

Constante	Descrição
E_ALL	Todos os erros e avisos são exibidos.
E_ERROR	Erros fatais de tempo de execução. Eles indicam erros que não podem ser recuperados, como um problema de alocação de memória.
E_RECOVERABLE_ERROR	Erro fatal captável. Indica que ocorreu um erro provavelmente perigoso, mas não deixou a Engine em um estado instável.
E_WARNING	Avisos em tempo de execução (erros não fatais). A execução do script não é interrompida.
E_PARSE	Erros de parse em tempo de compilação. Os erros de parse devem ser gerados apenas pelo parser.
E_NOTICE	Notificação de tempo de execução. Indica que o script encontrou algo que poderia indicar um erro, mas também poderia ocorrer no curso normal da execução de um script.

### ATIVIDADES:

1. Liste e explique brevemente os tipos de variáveis básicas no PHP, incluindo string, inteiro, float e boolean.
2. O que são variáveis compostas em PHP? Dê exemplos de quando e por que você usaria arrays e objetos.
3. Explique as principais diferenças entre os comandos de exibição "print" e "echo" no PHP. Em que situações você escolheria um em vez do outro?
4. Cite dois tipos de erros comuns na programação mencionados no texto e explique brevemente cada um deles.
5. O que é a variável de ambiente "REMOTE\_USER" e em que contexto ela pode ser útil para um desenvolvedor PHP?
6. Quais os tipos básicos de variáveis que temos no PHP?
7. O que é concatenar?
8. Descreva a diferença entre um erro de lógica e um erro de sintaxe em programação, e como você abordaria cada um deles durante o desenvolvimento de um programa.
9. Explique a importância da documentação oficial do PHP, e forneça um exemplo de como você a utilizaria para encontrar informações sobre uma função específica.
10. Qual é a função de uma variável?

- 
1. Os tipos de variáveis básicas no PHP são:
    - **\*\*String\*\***: Variável que armazena texto.

- **\*\*Inteiro\*\***: Variável que armazena números inteiros.
- **\*\*Float\*\***: Variável que armazena números decimais.
- **\*\*Boolean\*\***: Variável que armazena valores verdadeiro (true) ou falso (false).

2. Variáveis compostas em PHP são estruturas que podem armazenar múltiplos valores. Um exemplo são os arrays, que podem conter uma lista de elementos do mesmo tipo. Por exemplo, um array pode armazenar uma lista de números ou uma lista de nomes. Já os objetos são instâncias de classes que podem conter propriedades e métodos. Eles são úteis para modelar entidades complexas e definir comportamentos específicos para essas entidades.

3. As principais diferenças entre os comandos de exibição "print" e "echo" no PHP são:

- **\*\*Print\*\***: É uma construção da linguagem, não uma função. Pode imprimir uma ou mais expressões e sempre retorna 1.
- **\*\*Echo\*\***: É uma função que pode imprimir uma ou mais expressões e não tem valor de retorno.

4. Dois tipos de erros comuns mencionados no texto são:

- **\*\*Erro de lógica\*\***: Ocasionado quando a lógica do programa não está correta, levando a resultados inesperados ou incorretos.
- **\*\*Erro de sintaxe\*\***: Ocasionado por erros de digitação ou estruturação incorreta do código, resultando em falhas na interpretação pelo interpretador da linguagem.

5. A variável de ambiente "REMOTE\_USER" contém o nome de usuário fornecido pelo usuário na autenticação HTTP. Ela pode ser útil para um desenvolvedor PHP quando é necessário acessar informações sobre o usuário autenticado em um ambiente de autenticação baseado na web.

6. Os tipos básicos de variáveis no PHP são: String, Inteiro, Float e Boolean.

7. Concatenar é o processo de unir duas ou mais strings em uma única string. No PHP, isso é feito usando o operador de concatenação ".", que junta os valores das variáveis ou strings.

8. Um erro de lógica ocorre quando a lógica do programa não está correta, resultando em comportamento inesperado. Para corrigir esse tipo de erro, é necessário revisar e ajustar a lógica do código. Um erro de sintaxe ocorre quando o código não está estruturado corretamente e viola as regras gramaticais da linguagem de programação. Para corrigir esse tipo de erro, é necessário identificar e corrigir os erros de digitação ou a estrutura incorreta do código.

9. A documentação oficial do PHP é importante para os desenvolvedores, pois fornece informações detalhadas sobre os recursos da linguagem, funções, classes e métodos disponíveis. Um exemplo de como utilizá-la seria acessar o site oficial do PHP em <https://www.php.net/> e usar a caixa de pesquisa para encontrar informações sobre uma função específica, como "strlen". Isso fornecerá a documentação oficial da função `strlen` com descrição, parâmetros e exemplos de uso.

10. A função de uma variável é armazenar e manipular dados dentro de um programa. Elas permitem que os desenvolvedores armazenem informações temporárias ou dinâmicas durante a execução do programa, facilitando o processamento e a interação com os dados.