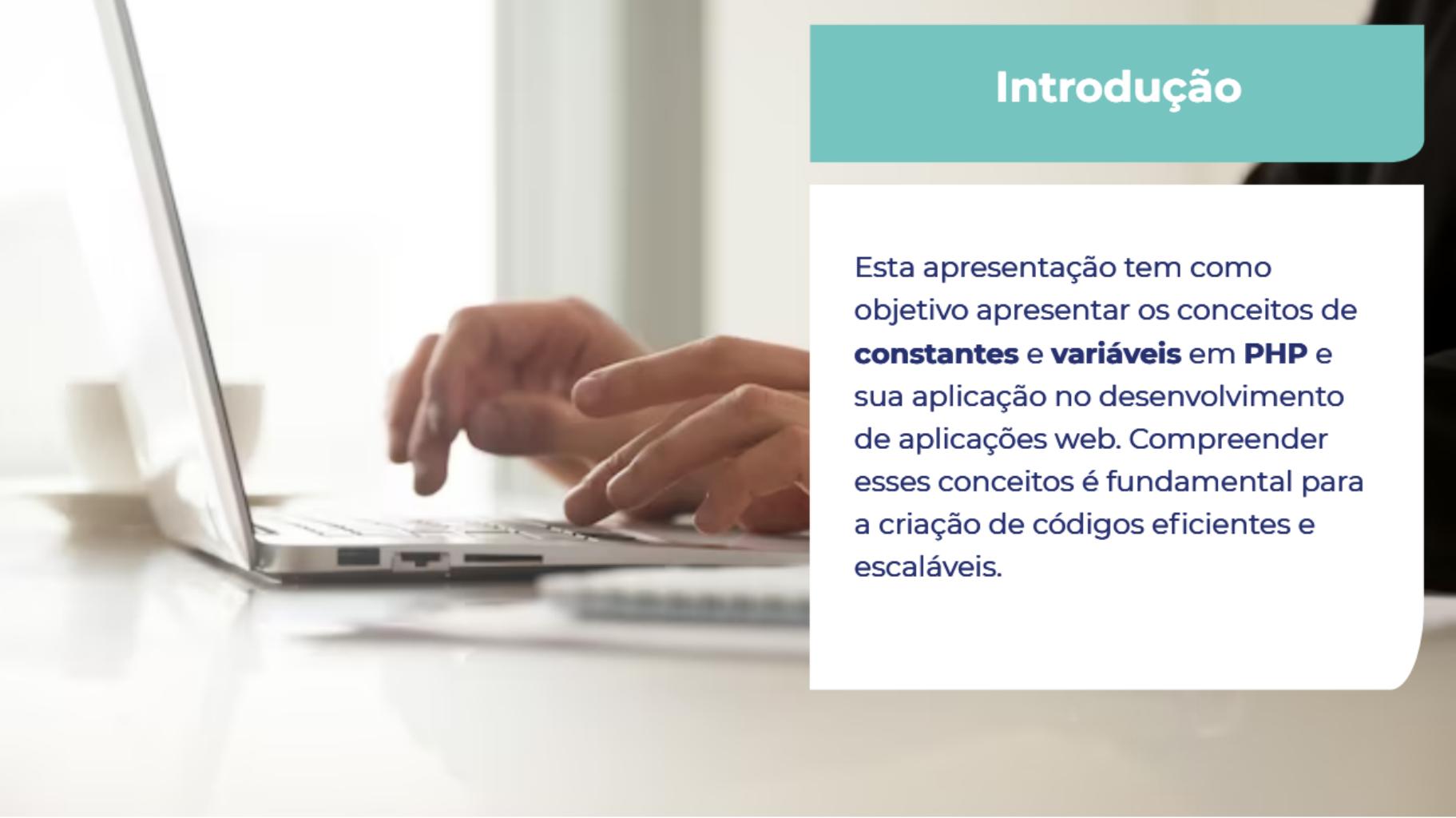




CONSTANTES E VARIÁVEIS EM PHP

Prof. Wanderlei Silva do Carmo
wander.silva@gmail.com



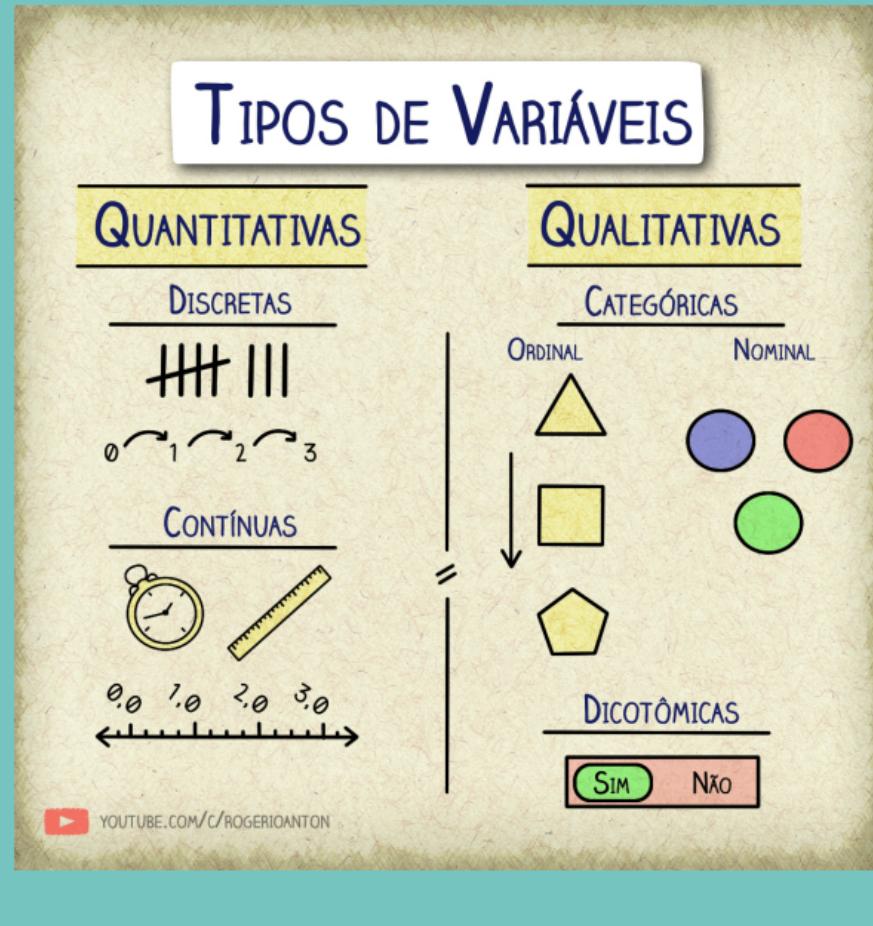


Introdução

Esta apresentação tem como objetivo apresentar os conceitos de **constantes e variáveis** em PHP e sua aplicação no desenvolvimento de aplicações web. Compreender esses conceitos é fundamental para a criação de códigos eficientes e escaláveis.

O QUE SÃO VARIÁVEIS?

Variáveis são espaços na memória que armazenam informações que podem ser utilizadas em diferentes partes do código. Em **PHP**, a declaração de variáveis é feita com o símbolo **\$** seguido do nome da variável. É importante entender como trabalhar com variáveis para a criação de códigos mais dinâmicos.



Constantes e Variáveis em PHP

Antes de começarmos, é importante entendermos o que são constantes e variáveis em PHP.

Constantes

As constantes são valores fixos que não podem ser alterados durante a execução do programa. Elas são definidas usando a função `define()` e têm a seguinte sintaxe:

php

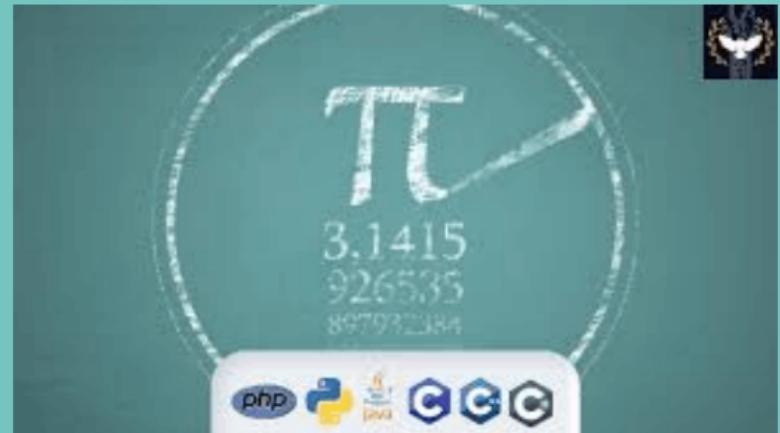
 Copy code

```
define('NOME_DA_CONSTANTE', valor_da_constante);
```

Por convenção, os nomes das constantes são escritos em letras maiúsculas.

O QUE SÃO CONSTANTES?

Constantes são valores fixos que não podem ser alterados durante a execução do código. Em **PHP**, a declaração de constantes é feita com a função **define()**. As constantes são úteis para armazenar informações que não mudam, como por exemplo, dados de conexão com banco de dados.



<https://wallky.com.br/tech/blog/imgs/folders/o-que-sao-constantes-programacao.webp>



CONSTANTES FÍSICAS

| Constante | Símbolo | Valor para cálculo | Valor + (incerteza) + unidade |
|--|--------------|---|---|
| Velocidade da luz no vácuo | c | $3 \cdot 10^8 \text{ m/s}$ | 2,997 924 58 (exato) |
| Carga elementar | e | $1,6 \cdot 10^{-19} \text{ C}$ | 1,602 177 33(49) $\cdot 10^{-19} \text{ C}$ |
| Número de Avogadro | N_A | $6,02 \cdot 10^{23}$ | 6,022 136 7(36) $\cdot 10^{23}$ |
| Constante da gravitação universal | G | $6,67 \cdot 10^{-11} \frac{\text{N} \cdot \text{m}^2}{\text{kg}^2}$ | 6,672 59(85) $\cdot 10^{-11} \frac{\text{N} \cdot \text{m}^2}{\text{kg}^2}$ |
| Permeabilidade elétrica do vácuo | ϵ_0 | $8,8 \cdot 10^{-12} \frac{\text{C}^2}{\text{Nm}^2}$ | 8,854 187 817 $\cdot 10^{-12} \frac{\text{C}^2}{\text{Nm}^2}$ (exato) |
| Permeabilidade magnética do vácuo | μ_0 | $4\pi \cdot 10^{-7} \frac{\text{T} \cdot \text{m}}{\text{A}}$ | $4\pi \cdot 10^{-7} \frac{\text{T} \cdot \text{m}}{\text{A}}$ (exato) |
| Constante eletrostática do vácuo ou Constante de Coulomb | k_0 | $9 \cdot 10^9 \frac{\text{N} \cdot \text{m}^2}{\text{C}^2}$ | 8,987 551 787 $\cdot 10^9 \frac{\text{N} \cdot \text{m}^2}{\text{C}^2}$ (exato) |
| Unidade de massa atômica | u | $1.66 \cdot 10^{-12} \text{Kg}$ | 1.660 540 2(10) $\cdot 10^{-12} \text{Kg}$ |
| Constante dos gases | R | $8,31 \frac{\text{J}}{\text{mol} \cdot \text{K}}$ | 8.314 510(70) $\frac{\text{J}}{\text{mol} \cdot \text{K}}$ |
| Constante de Planck | \hbar | $6.63 \cdot 10^{-34} \text{ J} \cdot \text{s}$ | 6.626 075(40) $10^{-34} \text{ J} \cdot \text{s}$ (exato) |

VARIÁVEIS

na programação



TIPOS DE VARIÁVEIS

Em **PHP**, existem diferentes tipos de variáveis, cada uma com uma função específica. Os principais tipos são: **string**, **integer**, **float**, **boolean** e **array**. É importante conhecer as características de cada tipo para utilizar corretamente em seu código.



Variáveis

As variáveis são valores que podem ser alterados durante a execução do programa. Elas são definidas usando o símbolo `\$` e têm a seguinte sintaxe:

```
php
```

 Copy code

```
$nome_da_variavel = valor_da_variavel;
```

Por convenção, os nomes das variáveis são escritos em letras minúsculas e podem ser compostos por letras, números e underscores (_).

Exemplo 1: Usando Constantes em PHP

```
php
```

 Copy code

```
define('PI', 3.14);
define('NOME', 'João');

echo 'O valor de PI é: ' . PI . '<br>';
echo 'O nome é: ' . NOME . '<br>';
```

Neste exemplo, definimos duas constantes: PI e NOME. Então, imprimimos o valor de PI e o nome na tela.



Exemplo 2: Usando Variáveis em PHP

php

Copy code

```
$idade = 20;
$nome = 'Maria';

echo 'O nome é: ' . $nome . '<br>';
echo 'A idade é: ' . $idade . '<br>';

$idade = 25;

echo 'A nova idade é: ' . $idade . '<br>';
```

Neste exemplo, definimos duas variáveis: `idade` e `nome`. Então, imprimimos o nome e a idade na tela. Depois, alteramos o valor da `idade` e imprimimos novamente.



Exemplo 3: Calculando a Média com Variáveis em PHP

php

Copy code

```
$nota1 = 8;  
$nota2 = 9;  
$nota3 = 7;  
  
$media = ($nota1 + $nota2 + $nota3) / 3;  
  
echo 'A média é: ' . $media . '<br>';
```

Neste exemplo, definimos três variáveis: nota1, nota2 e nota3. Então, calculamos a média das notas e imprimimos na tela.



Exemplo 4: Usando Constantes em Condições em PHP

```
php

define('IDADE_MINIMA', 18);

$idade = 20;

if ($idade >= IDADE_MINIMA) {
    echo 'Você é maior de idade.';
} else {
    echo 'Você é menor de idade.';
}
```

 Copy code

Neste exemplo, definimos uma constante IDADE_MINIMA com o valor 18. Então, verificamos se a idade é maior ou igual a IDADE_MINIMA e imprimimos uma mensagem na tela.

Espero que essa aula sobre constantes e variáveis em PHP possa ajudar a entender melhor esses conceitos e como utilizá-los em seus programas. Lembre-se que a prática é fundamental para fixar o conhecimento, então continue praticando e explorando esses recursos em seus próprios projetos!

Tipos de variáveis



1. Variáveis Numéricas

As variáveis numéricas são usadas para armazenar valores numéricos, como inteiros e números de ponto flutuante. Em PHP, não é necessário definir o tipo de variável ao declará-la, pois o PHP determina o tipo automaticamente.

php

 Copy code

```
$numero_inteiro = 10;  
$numero_flutuante = 3.14;
```

2. Variáveis de Texto

As variáveis de texto são usadas para armazenar sequências de caracteres. Em PHP, as strings podem ser declaradas usando aspas simples ou aspas duplas.

php

 Copy code

```
$nome = "João";  
$sobrenome = 'Silva';
```

3. Variáveis Booleanas

As variáveis booleanas são usadas para armazenar valores booleanos, como `'true'` ou `'false'`. Elas são comumente usadas em estruturas condicionais.

php

 Copy code

```
$verdadeiro = true;  
$falso = false;
```

4. Variáveis de Array

As variáveis de array são usadas para armazenar uma coleção de valores relacionados. Em PHP, os arrays podem ser declarados usando a função `array()` ou usando colchetes.

php

 Copy code

```
$nomes = array("João", "Maria", "Pedro");
$idades = [20, 25, 30];
```

5. Variáveis de Objeto

As variáveis de objeto são usadas para armazenar referências a objetos. Em PHP, os objetos são instâncias de classes definidas pelo usuário.

php

 Copy code

```
class Pessoa {  
    public $nome;  
    public $idade;  
}  
  
$pessoal = new Pessoa();  
$pessoal->nome = "João";  
$pessoal->idade = 20;
```

6. Variáveis de Recurso

As variáveis de recurso são usadas para armazenar referências a recursos externos, como conexões de banco de dados ou manipuladores de arquivos.

php

 Copy code

```
$arquivo = fopen("exemplo.txt", "r");
```

7. Variáveis Nulas

As variáveis nulas são usadas para indicar que uma variável não possui nenhum valor. Em PHP, as variáveis nulas são representadas pelo valor `null`.

php

 Copy code

```
$valor_nulo = null;
```



ESCOPO DE VARIÁVEIS

O escopo de uma variável determina a sua disponibilidade em diferentes partes do código. Em **PHP**, existem três tipos de escopo: **global**, **local** e **estático**. É importante entender como funciona o escopo das variáveis para evitar erros e criar códigos mais organizados.

O escopo de uma variável é a área do seu programa em que a variável pode ser acessada e usada. Em outras palavras, é a parte do seu código em que a variável é "visível" e pode ser utilizada.

Em PHP, existem quatro níveis de escopo de variáveis:

1. Escopo Local
2. Escopo de Função
3. Escopo Global
4. Escopo de Variável Estática

1. Escopo Local

Variáveis declaradas dentro de uma função só podem ser acessadas dentro daquela função. Essas variáveis são consideradas "locais" porque só existem enquanto a função está em execução.

php

 Copy code

```
function calcularValor($quantidade, $preco) {  
    $total = $quantidade * $preco;  
    return $total;  
}  
  
echo calcularValor(5, 10); // retorna 50  
echo $total; // vai gerar um erro porque a variável $total está fora do escopo
```

2. Escopo de Função

Variáveis declaradas fora de uma função, mas dentro do escopo global, são consideradas "globais" e podem ser acessadas em todo o programa. No entanto, é importante ter cuidado ao usar variáveis globais, pois elas podem ser facilmente alteradas em qualquer lugar do código, tornando o programa difícil de entender e depurar.

php

 Copy code

```
$taxa = 0.1; // variável global

function calcularValor($quantidade, $preco) {
    global $taxa; // define que a variável $taxa é global dentro da função
    $subtotal = $quantidade * $preco;
    $total = $subtotal + ($subtotal * $taxa);
    return $total;
}

echo calcularValor(5, 10); // retorna 55
```

3. Escopo Global

As variáveis declaradas dentro de uma classe são chamadas de propriedades da classe e podem ser acessadas por qualquer método dentro da classe. As propriedades da classe são consideradas "globais" dentro da classe.

php

 Copy code

```
class Pessoa {
    public $nome; // propriedade da classe
    public $idade; // propriedade da classe

    public function apresentar() {
        echo "Olá, meu nome é " . $this->nome . " e tenho " . $this->idade .
    }
}

$pessoal = new Pessoa();
$pessoal->nome = "João";
$pessoal->idade = 20;
$pessoal->apresentar(); // retorna "Olá, meu nome é João e tenho 20 anos."
```

4. Escopo de Variável Estática

Variáveis estáticas são aquelas que mantêm o seu valor entre chamadas de função. Elas são declaradas usando a palavra-chave `'static'`. As variáveis estáticas têm escopo de função, mas mantêm o seu valor mesmo depois que a função termina.

php

 Copy code

```
function contarChamadas() {
    static $contador = 0; // variável estática
    $contador++;
    echo "Esta função foi chamada " . $contador . " vezes.";
}

contarChamadas(); // retorna "Esta função foi chamada 1 vezes."
contarChamadas(); // retorna "Esta função foi chamada 2 vezes."
contarChamadas(); // retorna "Esta função foi chamada 3 vezes."
```



Escopo de uma variável é a área do programa em que ela pode ser acessada e usada. Em PHP, existem quatro níveis de escopo de variáveis: local, de função, global e de variável estática.

É importante entender o escopo de uma variável para evitar problemas de conflito de nomes e para garantir que as variáveis sejam usadas corretamente em todo o programa. É uma boa prática usar variáveis locais sempre que possível e evitar o uso excessivo de variáveis globais, pois elas podem dificultar a manutenção do código. Além disso, as variáveis estáticas podem ser úteis em situações em que você precisa manter o valor de uma variável entre chamadas de função.

CONCLUSÃO

Nesta apresentação, vimos os conceitos de **constantes** e **variáveis** em **PHP** e sua importância no desenvolvimento de aplicações web. Compreender como trabalhar com esses conceitos é fundamental para criar códigos eficientes e escaláveis. Espero que esta apresentação tenha sido útil para você!

Dúvidas?

<https://www.php.net>

