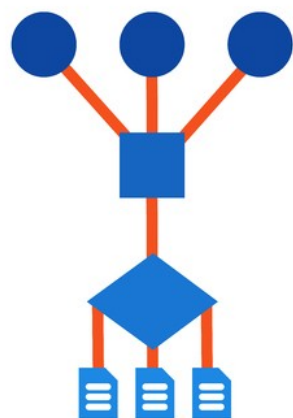


Linguagem PHP

Prof. Daniel Di Domenico



Introdução e
sintaxe dos comandos



PHP



- Acrônimo de **PHP**: **H**ypertext **P**rocessor
 - Originalmente era **P**ersonal **H**ome **P**age
- Vastamente utilizada nos dias atuais para implementação de páginas e sistemas Web
 - É utilizada por grandes empresas, como Facebook, Wikipedia, WordPress
- Gratuita e livre
- Multiplataforma: uma aplicação PHP pode rodar em diversos sistemas operacionais (Windows, Linux e Mac)
- **Site oficial: <https://www.php.net/>**

- **Características da linguagem de programação PHP:**
 - Suporta programação estruturada e **orientada a objetos**
 - É uma linguagem de script e interpretada
 - Script: pode-se executá-la diretamente pelo terminal
 - Interpretada: não requer compilação
 - Facilidade de aprendizado



PHP: programação



- Tipagem dos dados:
 - Dinâmica: variáveis não precisam ser declaradas e assumem o tipo do valor atribuído
 - Fraca: pode-se alterar o tipo de uma variável
- PHP é **case-sensitive** (para praticamente tudo)
 - **Diferencia letras maiúsculas de minúsculas**
 - **Cuidado**: os programas ficarão com erro se o case-sensitive não for considerado
 - Boa prática: utilize o padrão camelCase

PHP: requisitos de software

- **Linguagem**

- PHP (versão 7 ou superior)
 - Deve ser acessada pelo terminal

- **Visual Studio Code (VS Code):** editor de texto

- Instalar extensões:
 - PHP Intelephense (Ben Mewburn)
 - Path Intellisense (Christian Kohler)
 - PHP Getters & Setters (Christophe VERGNE)
- Desativar extensão:
 - PHP Language Features (digitar @builtin php)




PHP: executando um programa

- Nossos programas PHP serão executados através do terminal
 - `php programa.php`



- Como fazer e executar um programa **Olá Mundo** em PHP?

A black rectangular area representing a terminal window. Inside, the text "Hello World." is displayed in a green monospaced font.

```
Hello World.
```

PHP: programação

- Um programa PHP deve ser codificado dentro de um arquivo com extensão **.php**, devendo ser iniciado como um bloco
 - Início do bloco: **<?php**

```
<?php
```

```
//Comandos PHP aqui  
echo "Olá mundo!\n";
```

PHP: programação

- Exemplos de tipos de dados:

```
$var = "Aula em PHP"; //String
$var = 10; //Inteiro
$var = 3.1415; //Float
$var = true; //Booleano

//Identificar o tipo da variável
$tipo = gettype($var); //Retorna uma String com o tipo
```

- **Variáveis**
 - Identificadas pelo caractere \$ na frente do nome
- **Atribuição de valores à variáveis/objetos**
 - Caractere =
- **Final de instrução**
 - O PHP requer o uso de ; ao final de cada instrução
- **Strings**
 - Requer o uso " " ou ' ' de para identificar as Strings
 - Podem ser concatenadas com o operador . (ponto)

PHP: programação

- Verificação de tipos de dados:

```
is_null($var); //Retorna true, pois a variável é nula/vazia

$var = "Aula em PHP";
is_string($var); //Retorna true, pois a variável é uma string

$var = 10; //Inteiro
is_integer($var); //Retorna true, pois a variável é um inteiro

$var = true; //Booleano
is_bool($var); //Retorna true, pois a variável é um booleano

//Demais: is_float(), is_array(), is_object()
```

PHP: comandos

- A linguagem PHP possui **comandos** para controlar o fluxo de execução do código, sendo:
 - Comando **IF** (se) para condições
 - Comando **ELSE** (senão) para condições compostas
 - Comando **SWITCH** (escolha) para condições com casos fixos com uma única entrada
 - Comando **WHILE** (enquanto) para repetições controladas por condições
 - Comando **FOR** (para) para repetições considerando um intervalo numérico pré-definido

IF e ELSE (se e senão)

- Se a **condição** for verdadeira, o bloco é executado
 - ELSE é o caso contrário, sendo executado se as demais condições testadas nos IFs forem falsas

//IF simples

```
$n1 = 1;  
$n2 = 2;
```

Condição

```
if($n1 == 1 && $n2 >= 2) {  
    echo "n1 = 1 e n2 >= 2";  
}
```

Em PHP, as **chaves** para abrir e fechar o bloco são opcionais caso exista apenas uma linha a ser executada

//IF composto

```
$n1 = 1;
```

```
if($n1 == 1) {  
    echo "n1 = 1";  
} else if($n1 == 2) {  
    echo "n1 = 2";  
} else {  
    echo "n1 != 1 e n1 != 2";  
}
```

Pode utilizar
else if ou
elseif

PHP: programação

- Operadores relacionais:
 - `==` (igual)
 - `===` (idêntico – igual e de mesmo tipo)
 - `!=`, `<>` (diferente)
 - `!==` (não identico)
 - `>` (maior)
 - `<` (menor)
 - `>=` (maior ou igual)
 - `<=` (menor ou igual)
- Operadores lógicos
 - **and**, `&&` (E)
 - **or**, `||` (OU)
 - **xor** (OU EXCLUSIVO)
 - **!** (NEGAÇÃO - not)

SWITCH (escolha)

- Se a **variável** for igual ao **caso**, são executados os comandos do bloco
 - DEFAULT é o caso contrário. O uso do BREAK é obrigatório para finalizar o fluxo do comando SWITCH

Variável de
teste para os
casos

```
$n1 = 1;  
switch($n1) {  
    case 1:  
        echo "n1 = 1";  
        break;  
    case 2:  
        echo "n1 = 2";  
        break;  
    default:  
        echo "n1 != 1 e n1 != 2";  
}
```

WHILE (enquanto)

- **Repete** a execução de um bloco de comandos enquanto a condição for **verdadeira**

```
$n1 = 1;
```

//while com condição avaliada no INÍCIO

```
while($n1 <= 5) {  
    echo $n1 . "\n";  
    $n1 = $n1 + 1;  
}
```

Condição

```
$n1 = 1;
```

//while com condição avaliada no FINAL

```
do {  
    echo $n1 . "\n";  
    $n1 = $n1 + 1;  
} while($n1 <= 5);
```

Condição

Executa **ao menos uma vez**,
mesmo a condição sendo
inicialmente FALSA

Saída na tela:

1
2
3
4
5

FOR (para)

- **Repete** a execução de um bloco de comandos enquanto a condição for **verdadeira**
 - Após cada iteração, ocorre a execução de uma expressão de incremento

2- Condição de parada da repetição

$n1++$ equivale a
 $n1 = n1 + 1$

```
for( $n1=1; $n1<=5; $n1++ ) {  
    echo $n1 . "\n";  
}
```

1- Declaração da variável de controle da repetição

Saída na tela:

1
2
3
4
5

3- Incremento da variável de controle

Exercícios

- **1-** Faça um programa em PHP que imprima como saída os números de 1 a 10 de três formas:
 - Utilizando o comando WHILE
 - Utilizando o comando DO-WHILE
 - Utilizando o comando FOR
- **2-** Faça um programa que percorra os números de 1 a 100 e imprima a soma de todos eles como saída
- **3-** Faça um programa que percorra os números de 1 a 50 e imprima como saída apenas os números que sejam divisíveis por 3
 - Utilize o operador de resto de divisão: %
 - Exemplo:
 - $6 \% 3$ é igual a 0 – Então, 6 é divisível por 3
 - $7 \% 3$ é igual a 1 (1 é o resto da divisão) - Então, 7 NÃO é divisível por 3