

Linguagens de Script e PHP

Fatec
Campinas

Roteiro

- ▶ Origem
- ▶ Linguagem Script
- ▶ Linguagem Interpretada
- ▶ PHP

Origem

- ▶ Raízes no Job Control da IBM usado para controle de mainframes nos anos 50;
- ▶ Nos anos 60 surge o Shell Script, usado para controlar os sistemas operacionais;
- ▶ Foram surgindo diversas outras linguagens de script com diferentes propósitos;
- ▶ O nome vem do termo Script do cinemas.

O que é Linguagem de Script??

- ▶ Uma linguagem de programação executada dentro de um programa;
- ▶ Usada para automatizar comandos que seriam feitos por uma pessoa;
- ▶ São “interpretadas”, ou seja, um interpretador traduz o código para linguagem de máquina;

Linguagem Interpretada

► Vantagens

- Independente de Plataforma
- Tipagem dinâmica

► Desvantagens

- Desempenho mais lento que código de máquina compilado

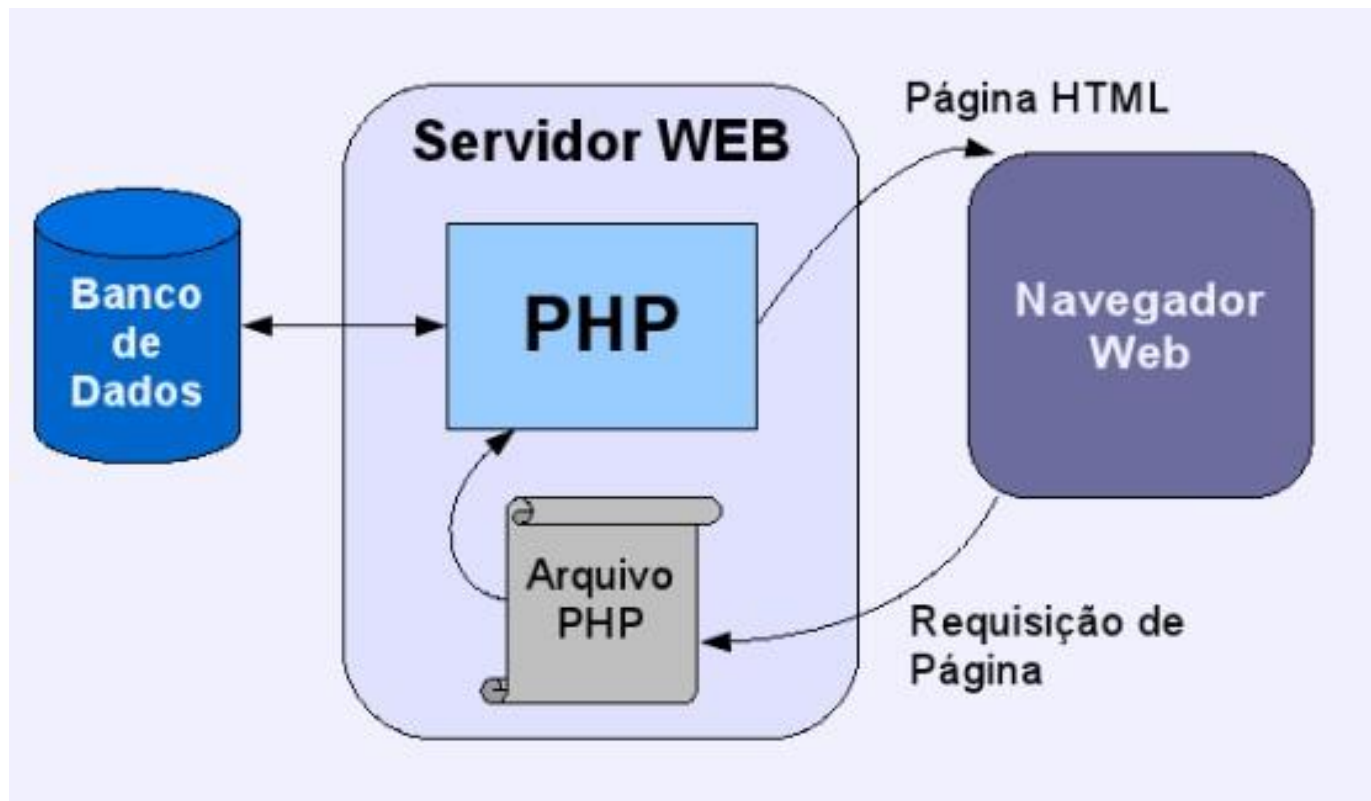
PHP

- ▶ PHP significa PHP: Hypertext Processor. Criado em por 1994 por Rasmus Lerdof;
- ▶ Código-Aberto;
- ▶ Cliente – Servidor
- ▶ Modularizado;
- ▶ Estruturado ou POO;
- ▶ Segunda linguagem interpretada mais usada no mundo;

PHP – Possibilidades

- ▶ Script no lado do servidor (server-side)
 - Pode funcionar somente com um interpretador em sistemas operacionais
- ▶ Script no lado do servidor (server-side)
 - Maior utilização
 - Interpretador PHP + Servidor Web + Navegador
- ▶ Aplicações GUI no lado cliente
 - Criação de Interface de Janelas
 - Independente de SO

Como funciona



Extensão de Arquivos

- ▶ .php
 - Arquivo PHP contendo um programa;
- ▶ class.php
 - Arquivo PHP contendo uma classe;
- ▶ inc.php
 - Arquivo a ser incluído, pode incluir constantes ou configurações;

Executando um código em PHP

- Para executar um código em PHP é necessário que o servidor da aplicação seja iniciado.
- Se estiver usando o XAMPP, abra o local de instalação (por padrão fica em C:\Xampp)
- Execute o arquivo xampp-control e verifique se o Apache está inicializado
- Seu arquivo deve ser salvo na pasta C:\Xampp \htdocs
- O arquivo principal da aplicação deve-se chamar **index**

Executando um código em PHP

- É possível executar um programa em php de duas formas,
1) abra o terminal, navegue até a pasta **c:/xampp/htdocs**

```
C:\Users\Janny>cd c:/xampp/htdocs  
c:\xampp\htdocs>_
```

- Digite o comando **php nomedoarquivo.php**

```
c:\xampp\htdocs>php teste.php
```

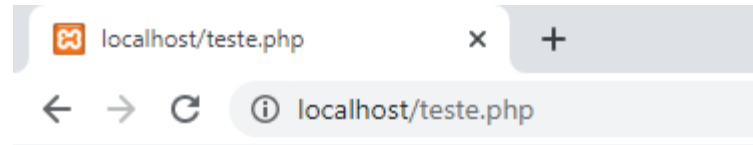
- Verifique o resultado

```
c:\xampp\htdocs>php teste.php  
Digite um nome:  
Fatec  
O nome digitado é Fatec  
c:\xampp\htdocs>_
```

Executando um código em PHP

2) abra o navegador, na URL digite localhost + o caminho da sua aplicação.

Exemplo:



Delimitadores de Código

- ▶ O código de um programa do php deve ser delimitado:

```
<?php
```

```
    //código ;
```

```
    //código ;
```

```
    //código ;
```

```
?>
```

- ▶ Todos os comandos são delimitado por ponto-e-vírgula ;

Comentários

- ▶ Para comentar uma única linha de código

```
// echo "a" ;
```

```
# echo "a" ;
```

- ▶ Para comentar várias linhas de código

```
/* echo "a" ;  
   echo "b" ; */
```

Comentários

- Facilita o entendimento de terceiros

```
<?php
```

```
/*
```

```
 * Classe para auxiliar cálculos
```

```
 *
```

```
 *@author Fatec Campinas
```

```
 *@version 1.0
```

```
 *
```

```
*/
```

Variáveis

- ▶ Identificadores utilizados para valores mutáveis e voláteis;
- ▶ São armazenadas em memória RAM;
- ▶ Sempre iniciado com \$;

Variáveis

```
<?php
```

```
// Define o nome da variável
```

```
$variavel = "nome";
```

```
?>
```

Variáveis

- Nunca inicie a nomenclatura de variáveis com números;
- Nunca utilize espaços no meio do identificador da variável;
- Nunca utilize caracteres especiais:
 - ! @ # \$ % ^ & * / [] { }

Variáveis

- ▶ Regras e boas práticas
 - Evite criar variáveis com mais de 15 caracteres para manter a clareza do código;
 - Nomes das variáveis devem ser significativos e transmitir a idéia de seu conteúdo;
 - Utilize preferencialmente palavras em minúsculo (separadas por _) ou somente a primeira letra em caso de ter mais de uma palavra;

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.

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.
$a \wedge b$	XOR	Os bits que estão ativos em a ou em b , mas não em ambos, são ativados.
$\sim a$	NÃO	Os bits que estão ativos em a não são ativados, e vice-versa.
$a \ll b$	Deslocamento à esquerda	Desloca os bits de a b passos para a esquerda (cada passo significa "multiplica por dois")
$a \gg b$	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 \wedge = \b	Equivalente a $\$a = \$a \wedge \$b$
$\$a << = \b	Equivalente a $\$a = \$a << \$b$
$\$a >> = \b	Equivalente a $\$a = \$a >> \$b$

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 se \$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.

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

► 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

Entrada de dados

- A entrada de dados pode ser feita através de linha de comando ou através de um formulário.
- O comando para entrada através da linha de comando é o
`fgets(STDIN);`

Exemplo:

```
<?php

    echo "Digite um nome:\n";

    $nome = fgets(STDIN);

    echo "O nome digitado é $nome";

?>
```

Saída de dados

- A saída de dados pode ser feita através de linha de comando ou através de um formulário.
- O comando para saída através da linha de comando é o

Echo “mensagem”; ou

Print “mensagem”

Exemplo:

```
<?php
    echo "Digite um nome:\n";

    $nome = fgets(STDIN);

    echo "O nome digitado é $nome";

?>
```

Entrada de dados

- Para fazer a entrada através de linha de comando, abra o terminal, navegue até a pasta **c:/xampp/htdocs**

```
C:\Users\Janny>cd c:/xampp/htdocs  
c:\xampp\htdocs>_
```

- Digite o comando **php nomedoarquivo.php**

```
c:\xampp\htdocs>php teste.php
```

- Verifique o resultado

```
c:\xampp\htdocs>php teste.php  
Digite um nome:  
Fatec  
O nome digitado é Fatec  
c:\xampp\htdocs>_
```


Exemplo

- Crie um programa que leia 2 números(entrada), calcule(processamento):

- a) A soma
- b) A subtração
- c) Multiplicação
- d) Divisão

E mostre o resultado (saída)

Exercícios

- 1) Crie um programa que escreva o seu nome completo, colocando uma palavra em cada linha
- 2) Crie um programa que solicite o nome, o endereço e a idade do usuário e depois mostra os dados digitados em uma única linha.
Exemplo:
 - <nome> mora no endereço: <endereço> e tem <idade> anos.
- 3) Crie um programa que leia uma idade de uma pessoa e calcule quantos dias essa pessoa já viveu.
- 4) Crie um programa que leia um número na tela, mostra seus 2 sucessores e 2 antecessores.
 - Exemplo: numero = 5, antecessores 4,3, sucessores 6,7
- 5) Crie um programa que lê o salário de um funcionário, e o percentual de reajuste. O programa deve calcular o salário reajustado e mostrar o resultado.
- 6) Crie um programa que lê um número real em dólar, e converte o valor para reais. Considere que a cotação é US\$ 1 = R\$ 5,47.
- 7) Crie um programa que leia as variáveis inteiras x e y e troque o valor destas variáveis. Isto é, x deve ficar com o valor de y e y deve ficar com o valor de x. Mostre os valores depois da troca
- 8) Faça um programa que solicite ao usuário o valor do litro de combustível (ex. 4,19) e qual valor ele deseja abastecer (ex. 50,00). Calcule quantos litros de combustível o usuário obterá com esses valores.
- 9) Sabendo-se que um veículo faz 12km/lt. Faça um programa que calcule a média de consumo de combustível de um veículo. O usuário deve informar o KM inicial (ex. 12500 km), o KM final (ex. 12700 km) e quantos litros foram gastos no percurso.
- 10) Faça um programa que calcule o valor a ser pago por uma dívida em atraso. O usuário deve informar o valor original da dívida (ex. R\$ 50,00), a quantidade de dias em atraso (ex. 35 dias) e o percentual da multa por dia de atraso (ex. 1%)

Obrigada!