

Introdução ao PHP

Olá,

Bem-vindos à matéria de Desenvolvimento Voltado à Web I. Para iniciar os nossos estudos, vamos mostrar alguns conceitos introdutórios para que vocês se situem dentro do tema.

Primeiramente, vamos estudar o que é o PHP e, em seguida, apresentaremos a sintaxe básica, os operadores e estruturas de controle da linguagem.

Leiam atentamente esta aula e, se tiverem alguma dúvida, usem os recursos que estão na sua área do aluno.

⚡ Bons estudos!

Objetivos de aprendizagem

Ao término desta aula, vocês serão capazes de:

- entender os conceitos básicos do PHP;
- executar códigos básicos no PHP.

Seções de estudo

- 1 – O que é PHP
- 2 – Sintaxe Básica
- 3 – Operadores
- 4 – Estruturas de Controle

1- O que é PHP

O PHP, acrônimo recursivo para *Hypertext Preprocessor* (Pré-Processador de Hipertexto), é uma linguagem *open source*, amplamente utilizada no desenvolvimento WEB (SOARES, 2014).

Através dessa linguagem é possível a implementação de sistemas web mais dinâmicos. Ela permite, desde a recuperação e armazenamento de informações em bancos de dados, até a manipulação de arquivos e envio de e-mails (SILVA, 2014).

O PHP é processado no lado do servidor e pode alterar e gerar códigos HTML, que serão enviados para o lado do cliente. O navegador que recebe o HTML nem precisa saber que essa página passou pelo PHP (BEIGHLEY; MORRISON, 2010).

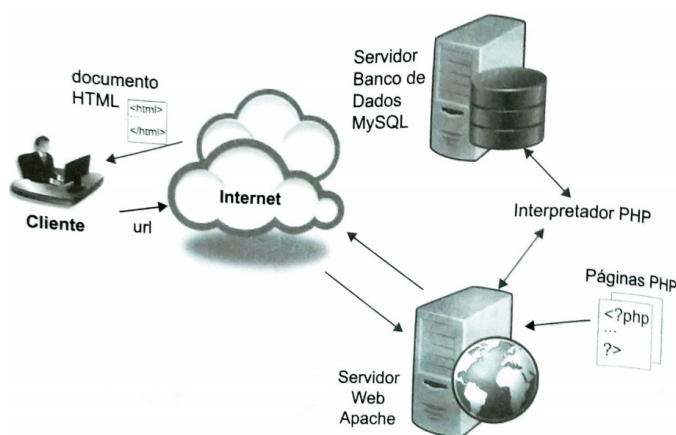


Figura 1 – Arquitetura PHP. Fonte: SOARES, 2014.

Soares (2014) apresenta, na Figura 1, a arquitetura do pré-processamento no PHP. Assim que o cliente (navegador) faz uma requisição a uma página (www.unigran.br/index.php, por exemplo) o servidor web atende a essa requisição, executando cada passo que segue:

1. O servidor identifica a extensão do documento;
2. Aciona então o interpretador PHP;
3. O interpretador PHP processa a página executando cada comando PHP que for encontrado nela;
4. O PHP retorna então para o servidor web um documento HTML com o resultado;
5. O servidor web responde ao cliente com o documento HTML gerado;
6. O navegador processa e exibe o documento HTML recebido.

Silva (2014) ressalta que a interface de apresentação para o usuário é de responsabilidade da linguagem de marcação HTML e das folhas de estilo CSS. E que algumas interações com o usuário podem ser realizadas pela linguagem JavaScript. Para o PHP fica a função de personalização de conteúdo.

Na disciplina de Desenvolvimento voltado à Web I foram expostas as características da arquitetura Cliente/Servidor e do Web Server HTTP. Foi demonstrada também a instalação do Apache HTTP Server (Xampp) que instala também o PHP. Caso você não possua o Xampp instalado na plataforma, serão disponibilizados vídeos e materiais que possam lhe ajudar.

1.1 – Características do PHP

A principal característica do PHP é sua capacidade de integração com o HTML, que facilita a geração de páginas web mais dinâmicas (SOARES, 2014).

Assim como o HTML, os arquivos PHP são textos que necessitam de um editor para serem criados/editados. O mais importante é não esquecer de atribuir ao arquivo salvo a extensão .php, que como foi visto, é a forma de o servidor entender que esse arquivo precisa ser interpretado pelo PHP (SILVA, 2014).

2- Sintaxe Básica

Dentro de um arquivo PHP, então, é possível colocar tags HTML misturados com códigos PHP, porém, é imprescindível que se faça a distinção deles. Para indicar ao interpretador onde começam e onde terminam os códigos em PHP, utilizamos as tags delimitadoras “<?php” e “?>” respectivamente (SILVA, 2014).

Não há uma limitação de quantas vezes podemos iniciar e fechar o PHP. Mas todo cuidado é necessário, já que o interpretador não tentará ler os trechos fora das tags delimitadoras como código PHP, o que pode gerar certos incômodos na hora do desenvolvimento (BENTO, 2014).

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Primeira      Página</
title>
  </head>
  <body>
    <?php
      echo "Primeira  página  em
PHP";
    ?>
  </body>
</html>
```

O resultado do documento acima será a exibição na página da frase “Primeira página em PHP”. Além de conter trechos de HTML e PHP juntos, que já vimos que era possível, é apresentada a primeira função do PHP o echo que exibe strings.

Subseção A – Comentários

Como na maioria das linguagens de programação, no PHP também é possível inserir comentários. O interpretador aceita comentários de única linha, identificados pelos marcadores “//”, ou de múltiplas linhas identificados pelos marcadores “/*” para início e “*/” para fim (SILVA, 2014).

```
<?php
/*
    Programa:    comentarios.php
    Autor:       Felipe Perez
    Data:        20/10/2019
    */
    //exibe a data e hora do servidor
    echo date("d-m-Y H:i:s");
?>
```

Subseção B – Variáveis

Variáveis são localidades, na memória, que podem armazenar valores e podem ser chamadas dentro do código. O PHP não exige que as variáveis sejam declaradas, ou atribuído um tipo. Quando utilizada pela primeira vez, automaticamente ela já é criada (SILVA, 2014).

São representadas pelo sinal de cifrão (\$), acompanhado de um identificador único, que deve seguir algumas regras para sua criação (SOARES, 2014).

1. Seu identificador deve iniciar por uma letra ou pelo símbolo underline (_);
2. Pode conter letras, números ou underline (_);
3. Não pode conter caracteres especiais;
4. O identificador é *case sensitive*, então o PHP diferencia as variáveis \$var, \$Var, \$vAR, \$VAR;
5. O PHP possui alguns nomes reservados, tais como: \$_GET, \$_POST, \$_FILES, \$_SESSION, entre outros.

Soares (2014) apresenta oito tipos de variáveis e separa em três grupos, os escalares, os compostos e os especiais. Dentro dos escalares estão os Inteiros, Ponto Flutuantes, Textos e Booleanos, nos compostos estão os Arrays e Objetos E nos especiais estão os recursos e o NULL (Nulo).

```
<?php
    $i = 10; //Inteiro
    $nome = "Felipe"; //String
    $falso = FALSE; //Booleano
    $valor = 100.50 //Ponto Flutuante
?>
```

As variáveis do tipo texto (string), no PHP, suportam quatro tipos de delimitadores, segundo Soares (2014). Veremos os principais, as aspas duplas (“”) e as aspas simples (‘’). Existem também os tipos especiais HEREDOC e NOWDOC utilizados, em geral, para textos longos.

As aspas simples suportam apenas texto e não traduzem nenhum caractere especial com exceção ao caractere “\”.

```
<?php
    $s = 'Texto simples';
    echo $s;
?>
```

As aspas duplas possibilitam a inclusão de caracteres especiais, além de incluir referência a outras variáveis dentro dela.

```
<?php
    $i = 10;
    $s = "valor de I = $i";
    echo $s;
?>
```

Subseção C – Constantes

Como visto na seção anterior, o PHP possui as variáveis cujos valores podem ser alterados de acordo com a programação. O PHP possui também constantes que, após serem especificadas, não podem ser alteradas ou removidas. As constantes somente aceitam tipos escalares (inteiros, texto, ponto flutuante e booleanos) (SOARES, 2014).

Para se definir uma constante, utilizamos o comando:

```
define(<nome>, <valor>, [<case
insensitive>])
```

O primeiro parâmetro define o nome da constante, que deve começar com uma letra ou com o caractere underline (_), no segundo o valor da constante é atribuído a ela. Soares (2014) afirma que a utilização do terceiro parâmetro não é aconselhada.

```
<?php
    define("VALOR", 10);
    define("FRUTA", "Manga");
    echo "Fruta = ".FRUTA;
    echo "Valor = ".VALOR;
?>
```

No exemplo acima, o valor das constantes VALOR e FRUTA são definidos. Em seguida, o conteúdo dessas constantes pode ser obtido por meio da concatenação no PHP que é feita utilizando o ponto (.), dois valores são concatenados quando entre eles o ponto for colocado.

O PHP possui diversas constantes predefinidas que podem ser auxiliares em atividades específicas. O manual online do PHP apresenta algumas constantes Mágicas do PHP, tais como __FILE__ que contém o caminho completo e nome do arquivo que está sendo executado e __DIR__ que retorna o diretório do arquivo (PHP, 2019).

3- Operadores

Os operadores no PHP são responsáveis pela transformação de valores. Soares (2014) descreve três grupos de operadores, os unários, os binários e os ternários. O primeiro manipula apenas um único valor, o segundo dois valores e o ternário avalia dois valores e retorna um terceiro valor ou expressão.

Os operadores são divididos em categorias, as mais utilizadas serão apresentadas nas subseções abaixo.

Subseção A – Operadores aritmético

Operações matemáticas básicas.

Operador	Função	Exemplo
+	Adição	\$valor + 10
-	Subtração	\$valor - 10
*	Multiplicação	\$valor * 10
/	Divisão	\$valor / 5
%	Módulo	\$valor % 2

Subseção B – Operadores de atribuição

O operador básico de atribuição é o sinal de = que atribui o resultado à direita à expressão que está à esquerda, \$valor = 12, o valor 12 é atribuído à variável \$valor.

A combinação de operações pode ser realizada, \$valor = (\$total = 12) - 7. Ao final da execução, a variável \$total tem o valor 12, enquanto a variável \$valor tem o valor 5.

O operador básico pode ser combinado também com operações aritméticas e de string, resultando nos seguintes operadores:

Operador	Função	Exemplo
=	Atribuição	\$valor = \$total + 5
=&	Atribuição por referência	\$valor = &\$total
+=	Atribuição e adição	\$valor += 5 (\$valor = \$valor + 5)
-=	Atribuição e subtração	\$valor -= 2 (\$valor = \$valor - 2)
*=	Atribuição e multiplicação	\$valor *= 10 (\$valor = \$valor * 10)
/=	Atribuição e divisão	\$valor /= 3 (\$valor = \$valor / 3)
%=	Atribuição e módulo	\$valor %= 2 (\$valor = \$valor % 2)
.=	Atribuição concatenação	\$valor .= "a" (\$valor = \$valor . "a")

Subseção C – Operadores de comparação

A função desses operadores é retornar verdadeiro ou falso de acordo com as expressões comparadas.

Operador	Função	Exemplo (\$valor = 2 e \$a = "2")
==	Igualdade de valor	\$valor == \$a (verdadeiro)
===	Idênticos de tipo e valor	\$valor === \$a (falso)
!=	Diferente	\$valor != \$a (falso)
<>	Diferente	\$valor <> \$a (falso)
!==	Não idêntico	\$valor !== \$a (verdadeiro)
<	Menor	\$valor < \$a (falso)
<=	Menor ou igual	\$valor <= \$a (verdadeiro)
>	Maior	\$valor > \$a (falso)

Subseção D – Operadores lógicos

Os operadores lógicos também retornam verdadeiro ou falso de acordo com a comparação feita entre os operandos. Com exceção da negação (!), que é unário, todos os outros são binários.

Operador	Função	Exemplo
AND	E	TRUE AND FALSE (falso)
&&	E (== AND)	TRUE && TRUE (verdadeiro)
OR	Ou	TRUE OR FALSE (verdadeiro)
	Ou (== OR)	FALSE OR FALSE (falso)
XOR	Ou exclusivo	TRUE XOR TRUE (falso)
!	Negação	!TRUE (falso)

Subseção E – Operadores de incremento/decremento

São operadores que permitem a adição e subtração diretamente na variável a que eles estão ligados, são sempre operações unárias.

Operador	Função
++\$valor	Incrementa antes de retornar o valor
\$valor++	Incrementa depois de retornar o valor
--\$valor	Decrementa antes de retornar o valor
\$valor--	Decrementa depois de retornar o valor

4 Estruturas de Controle

Presentes em todas as linguagens, as estruturas de controle direcionam o fluxo de execução do programa para realizar tarefas específicas, seja direcionando para um fluxo quando a instrução for verdadeira ou pela repetição de instruções que devem ser executadas várias vezes.

Nesta seção, serão exemplificadas algumas estruturas de controle do PHP.

Subseção A – If, elseif, else

Essa estrutura de controle possibilita que a partir de uma determinada condição, trechos de código possam ser executados. Caso a condição seja verdadeira ou falsa trechos específicos para essas condições serão executados.

O construtor if é responsável por criar esses grupos de comando.

```
<?php
    if ($valor==10) {
        //Bloco de comandos se $valor
        for igual a 10
    }elseif ($valor<10) {
        //Bloco de comandos se $valor for
        diferente e menor que 10
    }else{
        //Bloco de comandos se $valor for
        maior que 10
    }
?>
```

No exemplo acima, o if verifica a comparação da variável \$valor com 10. Caso essa comparação seja verdadeira, os comandos que estiverem delimitados pelas chaves {} que seguem o if serão executados. O opcional elseif fica responsável por criar mais um fluxo, caso a comparação inicial do if seja falsa, porém ele pode criar uma nova condição. Já o opcional else delimita e executa quando todas as outras condições forem falsas.

Subseção B – while, do while

As estruturas while e do while executam um bloco de comandos repetidas vezes. A cada iteração da estrutura, uma expressão é avaliada, e caso a expressão seja verdadeira, uma nova iteração é executada, caso seja falsa, a repetição é interrompida.

```
<?php
    $valor=0;
    while($valor<10){
        echo $valor++; //resultado:
0123456789
    }
?>
```

A variável valor é inicializada com 0. A estrutura while, a cada iteração, verifica se o valor da variável \$valor é menor que 10. Enquanto essa afirmação for verdadeira, mais uma rodada é iniciada. Dentro do bloco de instruções que serão repetidas deve-se criar as condições necessárias para que a estrutura pare, caso contrário, o loop nunca se encerrará. No exemplo acima, a cada iteração do laço, o valor da variável é incrementado, o que fará com que esse bloco se repita por 10 vezes.

```
<?php
    $valor=10;
    do{
        echo $valor++; //resultado:
10
    }while($valor<10)
?>
```

A diferença entre essas estruturas é que a segunda executa pelo menos uma vez, mesmo que a expressão já tenha valor falso. No primeiro exemplo a expressão é avaliada antes de entrar no bloco de execução, já no segundo exemplo, a expressão é avaliada depois de ter executado pelo menos uma vez o bloco.

Subseção C – for

Para criar lógicas mais complexas de repetição, o PHP dispõe da estrutura for. Ela funciona basicamente com um conjunto de três expressões. A primeira é avaliada apenas uma vez e somente na primeira iteração (normalmente utilizada para inicializar a variável de controle), a segunda é avaliada no início de cada iteração. Caso o resultado dessa expressão seja falso, o laço for é então encerrado, e a terceira é avaliada ao fim de cada iteração (normalmente utilizada para manipular a variável de controle para que o loop possa se encerrar).

```
<?php
    for($i=0;$i<10;$i++){
        echo $i; //resultado:
0123456789
    }
?>
```

Soares (2014) ressalta que as expressões podem ser vazias, possibilitando o controle dentro do próprio loop. Teremos, então, um loop infinito que precisará de uma instrução break, que veremos mais adiante, para poder encerrar o loop.

Subseção D – foreach

Essa estrutura permite que as condições para o loop não precisem ser passadas diretamente, já que ela percorre um array, para retornar cada valor. Antes de exemplificar o funcionamento da estrutura, vamos entender o que é um array.

Também conhecido por lista ou vetor, é uma variável que pode conter um conjunto de elementos de qualquer tipo (inteiro, string, booleano). No PHP, o array pode ser unidimensional ou multidimensional. Nesse segundo, cada posição do array aponta para outro array.

O grande diferencial do array do PHP é que os elementos dele podem ser acessados por meio de índices ou também por chaves associativas.

```
<?php
    $arr = array();
    $arr[1] = 1;
    $arr["nome"] = "Felipe";
?>
```

No exemplo acima, um array foi instanciado na variável \$arr. Foi atribuída à posição 1 do vetor o valor 1, e na chave “nome” foi atribuído o valor “Felipe”. Para a recuperação desses valores podemos utilizar essas informações, mas quando não sabemos exatamente o índice a chamar, usamos, então, o foreach:

```
<?php
    $arr = array();
    $arr[1] = 1;
    $arr["nome"] = "Felipe";
    foreach($arr as $key => $value){
        echo "$key = $value <br>";
    }
//resultado:
//1 = 1
//nome = Felipe
?>
```

Subseção E – switch

Muitas vezes utilizar somente o if, elseif e else deixará o código extenso. Para isso, a estrutura switch simplifica quando somente uma variável precisa ser comparada com diversos valores.

```
<?php
    $i = 0;
```



```

switch($i) {
    case 0:
        echo "i tem valor 0";
        break;
    case 1:
        echo "i tem valor 1";
        break;
    case 2:
        echo "i tem valor 2";
        break;
    default:
        echo "I não tem valor
0, 1 ou 2";
        break;
}
?>

```

O switch avalia a expressão ou variável que foi passada para ele, então o bloco de código que segue o valor correto para a expressão ou variável é executado. Nessa estrutura a instrução break pode ser omitida. Dessa forma, o switch começa a executar pelo caso correto e executará os outros em sequência, até encontrar um break, ou o fim da estrutura.

Subseção F – break e continue

As instruções break e continue, assim que executadas, informam as estruturas de repetição que o loop deve ser encerrado. Com a diferença que o break encerra todo o loop da estrutura de repetição, enquanto o continue encerra a iteração, porém, segue no loop.

```

<?php
for($i=0;$i<10;$i++){
if($i%2!=0)
continue;
echo $i; //resultado: 02468
}
for($i=0;$i<10;$i++){
if($i%2!=0)
break;
echo $i; //resultado: 0
}
?>

```

No primeiro for, sempre que um número ímpar for encontrado na variável \$i, a iteração se encerra e reinicia o loop, enquanto no segundo for, quando o primeiro valor ímpar foi encontrado, o for foi encerrado.

Retomando a aula

Chegamos ao final da primeira aula. Vamos recordar?

1 – O que é PHP

O PHP foi definido explicando o funcionamento do seu interpretador, que reconhece o tipo de arquivo solicitado pelo

lado do cliente, constrói o documento dinâmico e permite o envio para o cliente pelo servidor.

2 – Sintaxe Básica

Assim como todas as linguagens de programação, o PHP possui uma estrutura básica. Primeiro, não podemos esquecer de delimitar onde inicia e termina o código PHP para, em seguida, poder utilizar os conceitos de variáveis constantes e comentários vistos.

3 – Operadores

Foram apresentados os operadores presentes em diversas linguagens de programação, desde as operações básicas matemáticas, até os operadores lógicos que possibilitam a criação de expressões lógicas.

4 – Estruturas de Controle

Por fim, foram exemplificadas as estruturas de controle que permitem que o programador tenha controle sobre como o fluxo de instruções será executado.

Vale a pena



Vale a pena ler,

SOARES, W. *Programação Web com PHP 5*, Érica, 2014.
 BEIGHLEY, L; MORRISON, M. *Use a Cabeça! PHP & MySQL*. Alta Books, 2010.
 SILVA, J. M. C. *PHP na prática 200 Exercícios resolvidos*. Elsevier, 2014.
 BENTO, E. J. *Desenvolvimento web com PHP e MySQL*. Editora Casa do Código, 2014.

Vale a pena acessar,



PHP In: PHP Documentation, 2019. Disponível em: <
<https://www.php.net/docs.php>>. Acesso em: 20 out. 2019.