

Iniciando PHP

Objetivo Geral: Introdução ao PHP

Conteúdo:

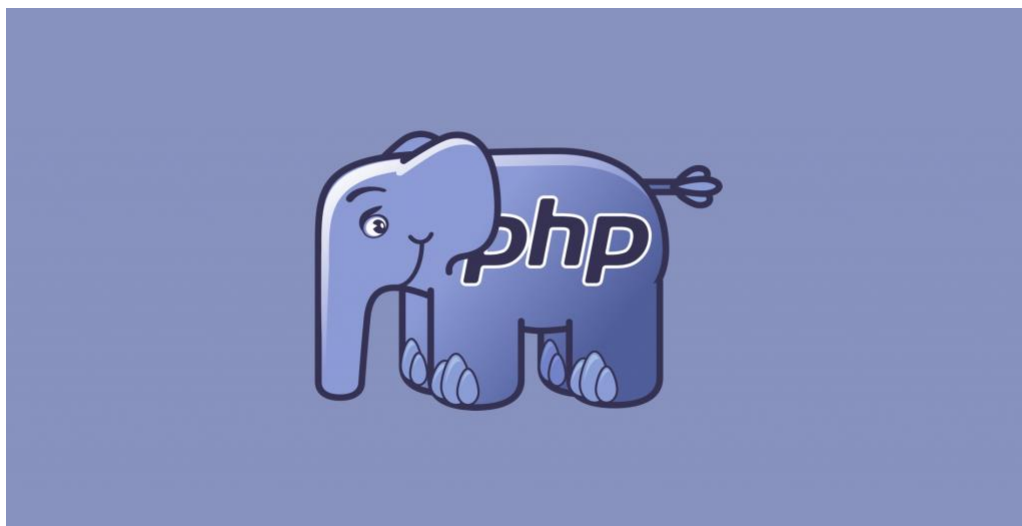
1. O que é PHP
2. Instalando e Configurando PHP
3. Estudo de Codigos em PHP

Metodologia e Estratégia:

1. Aula expositiva dialogada com apoio de tutorial;
2. Exercícios de aplicação.

O que é PHP

PHP (PHP:Hypertext Preprocessor) é uma linguagem de programação interpretada e open source, focada no desenvolvimento de sistemas para web, que pode ser utilizada junto com HTML para a criação de páginas web dinâmicas.



O PHP de fato é uma das linguagens de programação mais populares da atualidade. Ela permite que você crie aplicações web robustas, de uma maneira muito simplificada e direto ao ponto.

Sem contar que a linguagem traz diversos recursos que facilitam e aceleram o processo de desenvolvimento de sites e sistemas para web.

E além do mais, ela ainda tem um ótimo ecossistema, um excelente comunidade, e um grande mercado de trabalho.

Instalando o PHP

Muitas vezes procuramos ferramentas que vão facilitar a configuração do ambiente de desenvolvimento, mas temos que tomar o devido cuidado para que não fiquemos

presos nestas ferramentas de modo que não saibamos a trabalhar em outro ambiente. Ainda mais para quem é iniciante ou está aprendendo, é importante saber trabalhar de modo “desapegado”, tanto para ter flexibilidade, quanto para entender como as coisas funcionam.

Fazer o download do PHP e configura-lo no Windows.

Acesse <http://php.net/downloads.php> e faça o download do PHP para o Windows. Você pode fazer o download da versão que quiser desde que seja maior que a 5.4 (dê preferência para Non Thread Safety). Sugiro fazer o download da versão 8.1 ou superior, pois a versão 5 tem seu suporte finalizado em 18 de dezembro de 2018.

PHP 8.1 (8.1.12)

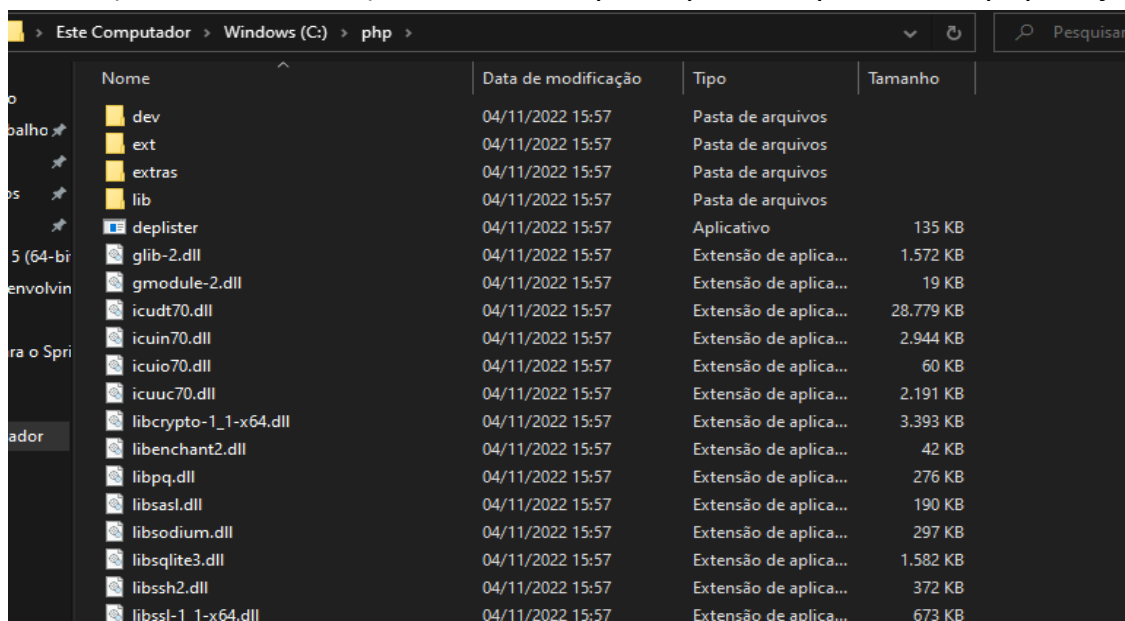
[Download source code](#) [25.37MB]

[Download tests package \(phpt\)](#) [15.15MB]

VS16 x64 Non Thread Safe (2022-Oct-25 19:03:48)

- [Zip](#) [29.25MB]
sha256: dbf95bc61fd342c4b5a086c5a34129d710126b22f74485576367030b3bb97df1
- [Debug Pack](#) [23.84MB]
sha256: 0c853165aacf82f931d3d0ca61480157d73158010682776049e0b0e41cb1ca74
- [Development package \(SDK to develop PHP extensions\)](#) [1.21MB]
sha256: 82aac4a116aa0cd017c6ca187a012bda260b3bddc41b765c2528d537031b643c

Após o download, descompacte o arquivo .zip na partição principal do seu computador (normalmente C:) e renomeie a pasta para simplesmente php. Veja:



Configurando o PHP no Computador

Para o PHP funcionar corretamente, precisamos ativar suas configurações padrões. Dentro da pasta descompactada no C:, temos vários arquivos, o executável do PHP (php.exe) e dois arquivos de configurações: php.ini-development e php.ini-production. Estes dois arquivos apresentam configurações para rodar o PHP em modo de produção (mais adequado quando se hospeda uma aplicação em um servidor na internet) ou em modo desenvolvimento que é adequado para se desenvolver aplicações PHP, pois com isso ativamos o controle de erros e debug da aplicação. Portanto, renomeie o arquivo php.ini-development para somente php.ini. O executável do PHP sempre carregará as configurações do arquivo php.ini.

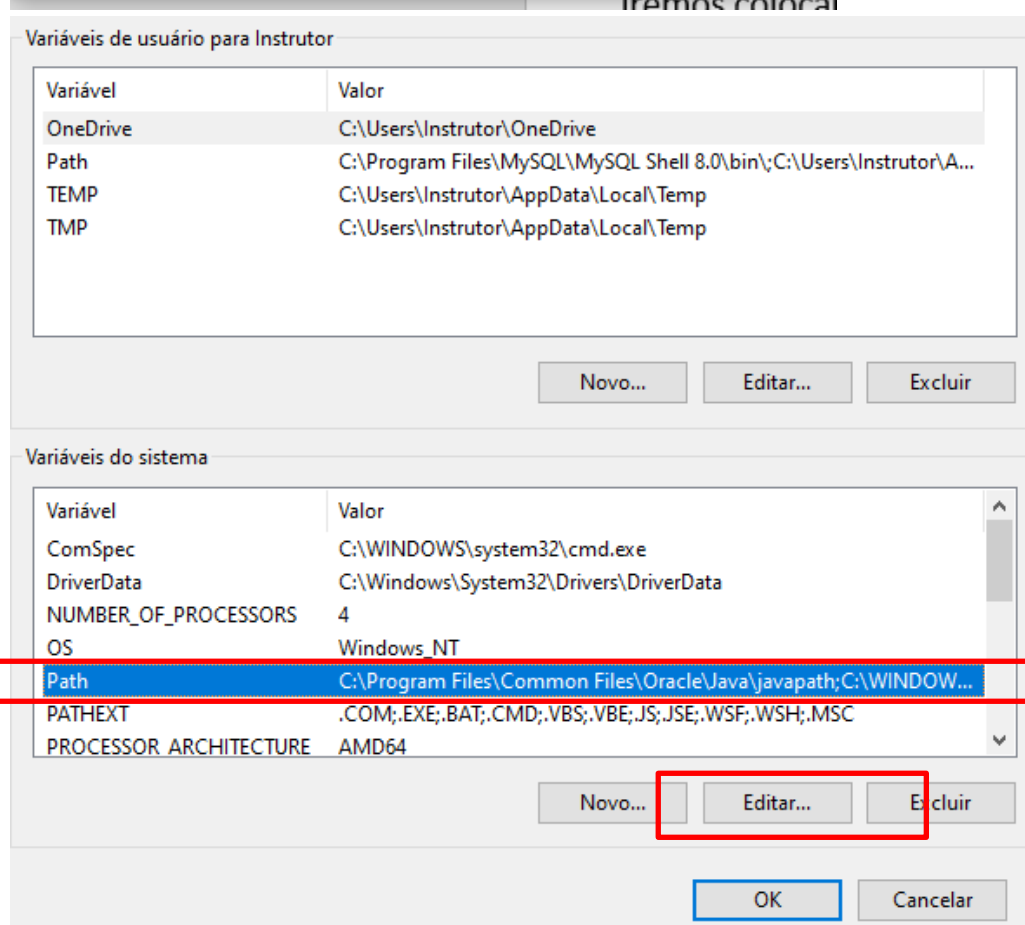
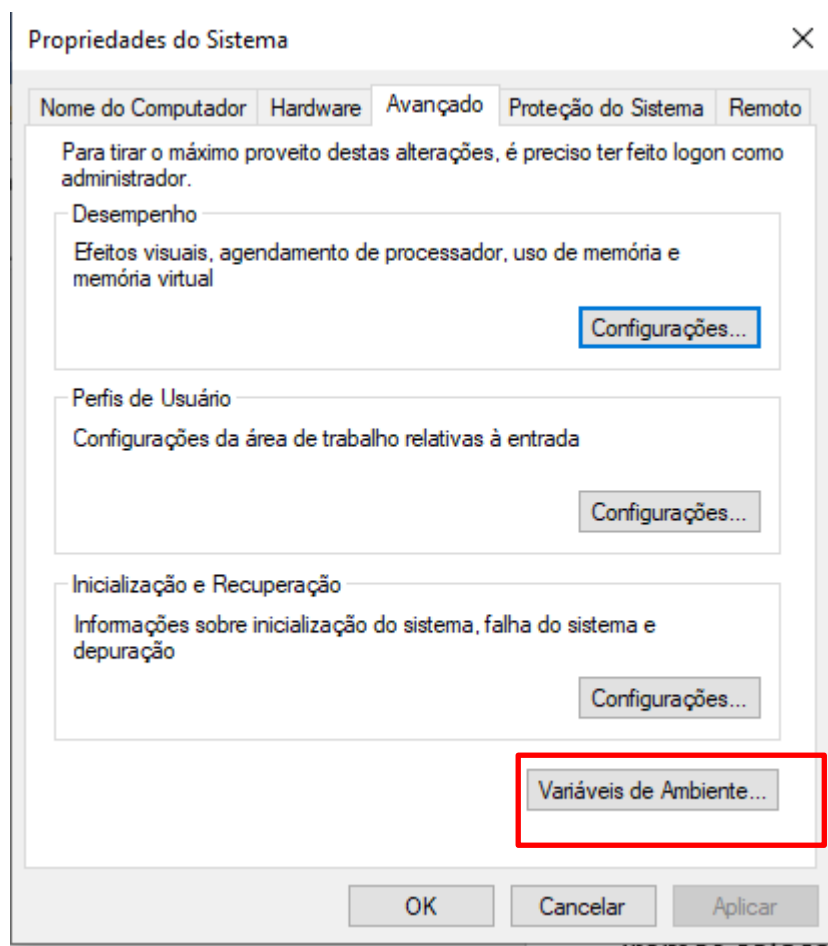
phar.phar	04/11/2022 15:57	Arquivo em Lotes ...	1 KB
pharcommand.phar	04/11/2022 15:57	Arquivo PHAR	65 KB
php	04/11/2022 15:57	Aplicativo	139 KB
php.ini-development	04/11/2022 15:57	Arquivo INI-DEVE...	73 KB
php.ini-production	04/11/2022 15:57	Arquivo INI-PROD...	74 KB
php8.dll	04/11/2022 15:57	Extensão de aplica...	8.913 KB
php8embed.lib	04/11/2022 15:57	Arquivo LIB	902 KB

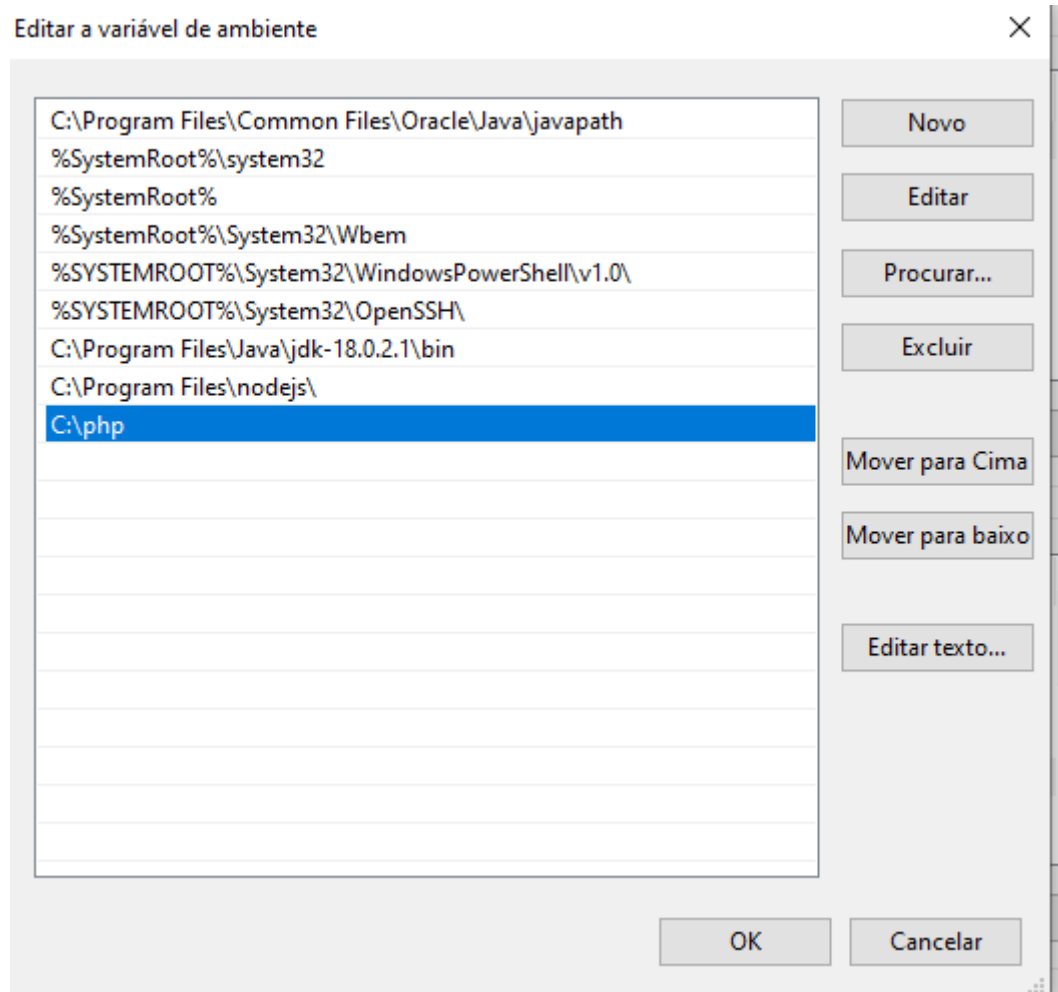
php.ini

pharcommand.phar	04/11/2022 15:57	Arquivo PHAR	65 KB
php.exe	04/11/2022 15:57	Aplicativo	139 KB
php.ini	04/11/2022 15:57	Parâmetros de co...	73 KB
php.ini-production	04/11/2022 15:57	Arquivo INI-PROD...	74 KB
php8.dll: Arquivo INI-PRODUCTION	04/11/2022 15:57	Extensão de aplica...	8.913 KB

Temos que configurar o PHP para funcionar em qualquer momento no terminal.

Abra o Painel de Controle, vá em Sistema, selecione a guia Avançado, depois clique em Variáveis de ambiente no rodapé da janela. Na seção Variáveis do sistema, selecione Path, agora clique em Editar, em Valor da variável, vá até o final do campo de texto, agora iremos colocar o caminho onde o nosso PHP está, acrescente antes um ; (ponto e vírgula) para finalizar os caminhos anteriores e coloque c:\php, então, ficará assim:;c:\php. Confirme tudo e o PHP está configurado.





Instalando o Composer

O Composer é o gerenciador de dependências mais utilizado do PHP, que com o passar dos anos foi ganhando cada vez mais espaço e se tornando dia após dia mais indispensável ao desenvolvedor.

Inspirado no npm do Node e no bundler do Ruby, o Composer foi desenvolvido por Nils Adermann e Jordi Boggiano e foi lançado no ano de 2012 sobre a licença MIT.

O Composer fornece recursos de carregamento automático para bibliotecas para facilitar o uso de código de terceiros, tornando o desenvolvimento do projeto mais simples e permitindo com que o desenvolvedor foque apenas em seu código, não no uso de código de terceiros.



A Dependency Manager for PHP

Latest: **2.4.4** ([changelog](#))

[Getting Started](#)

[Download](#)

<https://getcomposer.org/>

Após a Instalação do Composer

Abrir o Arquivo php.ini

```
;extension=bz2
extension=curl
;extension=ffi
;extension=ftp
extension=fileinfo
;extension=gd
;extension=gettext
;extension=gmp
;extension=intl
;extension=imap
;extension=ldap
extension=mbstring
;extension=exif      ; Must be after mbstring as it depends on it
;extension=mysqli
;extension=oci8_12c  ; Use with Oracle Database 12c Instant Client
;extension=oci8_19  ; Use with Oracle Database 19 Instant Client
;extension=odbc
extension=openssl
;extension=pdo_firebird
;extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
extension=pdo_pgsql
;extension=pdo_sqlite
extension=pgsql
;extension=shmop
```

Modificando a extension do PHP

Curso Programação de Internet

Disponível em:

<https://drive.google.com/drive/folders/15yoeHEwDOyfHb2OEfOwWEh2wBuKx1yad?usp=sharing>

Elaborado por Diogo Takamori Barbosa

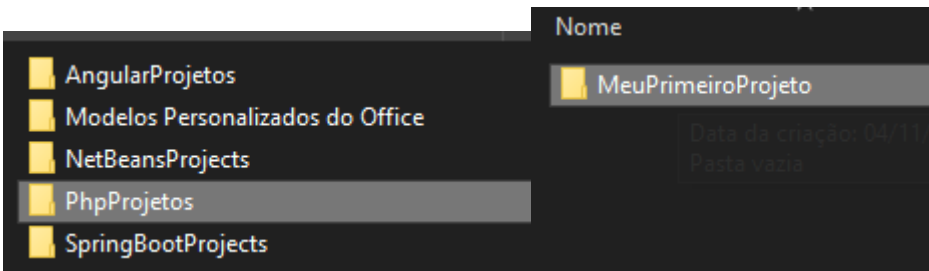
Abra o terminal de sua preferência e digite:

php -v

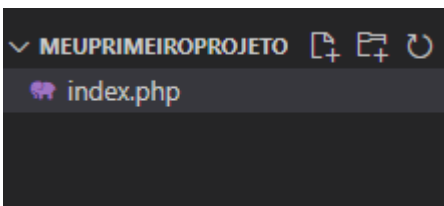
```
C:\Users\Instrutor>php -v
PHP 8.1.12 (cli) (built: Oct 25 2022 18:20:48) (NTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.1.12, Copyright (c) Zend Technologies
```

Criando Meu Primeiro Projeto em PHP

Crie uma pasta Para os Projetos em PHP no seu DRIVE



e abrir a pasta no VS Code.



criar o arquivo index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
<h1>Olá Alunos Vamos Aprender PHP????</h1>
  <?php
    echo "Olá, Mundo!!!";
  ?>
</body>
</html>
```

Abra o Terminal no VS Code

E digite:

Curso Programação de Internet

Disponível em:

<https://drive.google.com/drive/folders/15yoeHEwDOyfHb2OefOwWEh2wBuKx1yad?usp=sharing>

Elaborado por Diogo Takamori Barbosa

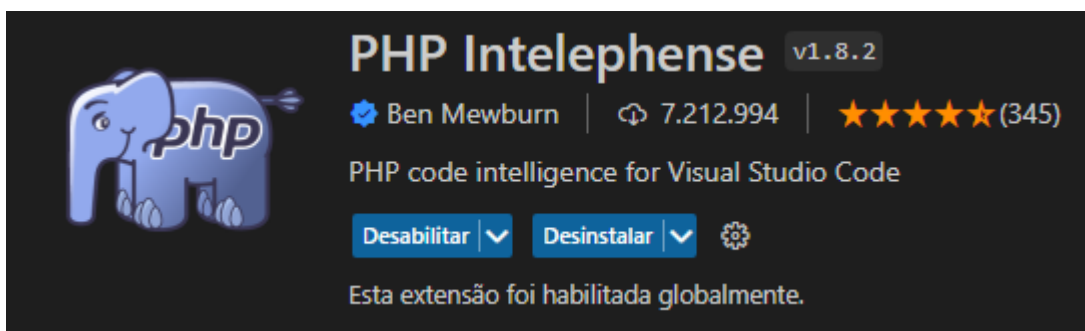
`php -S localhost:8080`

OBS: o "S" tem que ser maiúsculo

Até a versão do PHP 5.3 tínhamos que ter um servidor web (Apache, Nginx, etc) a parte instalado para rodar aplicações no browser, mas, a partir da versão 5.4 foi introduzido um servidor embutido no PHP. Assim, podemos iniciar um servidor a qualquer momento e em qualquer pasta, e ainda ter múltiplos servidores rodando simultaneamente em portas diferentes. Isto para o desenvolvimento é uma "mão na roda".

Instalando Extensao para PHP no VS Code

PHP Intelephense



Para utilizarmos a PHP Intelephense devemos desabilitar a `@builtin php`

Estudo Funções em PHP

PHP é uma linguagem de programação bem versátil e muito familiar a linguagem HTML. O código PHP está escrito entre as tags `<?php ?>`. O arquivo index foi salvo com a extensão `.php` para mostrarmos ao nosso interpretador que há um código PHP a ser interpretado. Além disso, no exemplo usamos a função `echo` para escrever na tela uma mensagem.

Como comentar o código no PHP

Para comentarmos o nosso código PHP usamos duas barras ou `#` para comentários de uma linha, e para comentários de múltiplas linhas usamos `/* */`, o mesmo usado no CSS.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Programação</title>
</head>
```

Curso Programação de Internet

Disponível em:

<https://drive.google.com/drive/folders/15yoeHEwDOyfHb2OefOwWEh2wBuKx1yad?usp=sharing>


```
<body>
<h1>Olá Alunos!!! Vamos Aprender PHP????</h1>
<?php
    echo "Oi, Eu serei visto na sua tela<br>";
    // Eu não! Sou apenas um comentário.

    echo "Oi, Eu também serei visto por você<br>";
    # Já eu não serei!

    echo "E eu aqui novamente na sua tela, rs<br>";
    /* Eu não aparecerei na sua tela novamente
    pois sou um comentário */
?>
</body>
</html>
```

Variáveis no PHP

Para criarmos uma variável basta utilizar o sinal de cifrão. Uma variável pode armazenar textos e números. Além disso, a linguagem PHP é case sensitive, então A é diferente de a. Observe um exemplo de uso de variáveis

```
<?php
$name = "Diogo";
$age = 38;

echo $name; // Diogo
echo "<br>";
echo $age; // 38
?>
```

No exemplo criamos uma variável \$name e declaramos a ela uma string, sendo assim precisamos colocá-la entre aspas. Já a outra variável \$age é declarada como inteiro, então não é necessário o uso de aspas. Ao usarmos o comando echo nas variáveis, o resultado impresso é o conteúdo dessa variável.

Para a nomeação de variáveis, as dicas a seguir são necessárias:

- Não inicie o nome de uma variável com números;
- Não utilize espaços em brancos;
- Não utilize caracteres especiais, somente underline;
- Crie variáveis com nomes que ajudarão a identificar melhor a mesma;
- Evite utilizar letras maiúsculas.

Tipos de Variáveis - Falaremos agora sobre alguns dos tipos de variáveis que existem no PHP:

- Booleanos: Este é o tipo mais simples, pois só pode expressar apenas dois valores: TRUE (1) ou FALSE (0, null ou uma string vazia);
- Integer: é um número inteiro, podendo ser negativo ou positivo;
- Float: também chamado de double ou números reais representados com um ponto para separar os dígitos do valor inteiro dos dígitos do valor das casas decimais.
- Strings: é uma palavra ou frase entre aspas simples ou duplas, assim como também pode ser binário, como o conteúdo de um arquivo MP3 ou JPG.

Obs1: Quando queremos que o PHP interprete o valor de nossa variável dentro de uma string é necessário o uso de aspas duplas.

Obs2: Podemos usar um ponto para concatenar strings, assim como o sinal + para o JavaScript.

Constantes no PHP

O valor de uma constante jamais poderá ser alterado enquanto estiver sendo executada e para defini-la utilizamos a função define() ou const.

```
<?php
define("PHP", "Linguagem Open - Source");

const HTML = "Linguagem de marcação";

echo PHP; // Linguagem Open - Source

echo HTML; // Linguagem de marcação
?>
```

Utilizando a função define() ou a palavra reservada const definimos que as constantes com os nomes de PHP e HTML respectivamente, com o valor: Linguagem Open – Source e Linguagem de marcação.

Array no PHP

Às vezes existe a necessidade de armazenar diversos dados em um único local com fácil acesso, vimos que as variáveis têm o papel de guardar dados de forma temporária, mas nelas, não podemos armazenar múltiplos dados, apenas uma informação por vez.

E para resolver este problema, existem os arrays, que permitem que você armazene múltiplos dados de uma só vez em um mesmo lugar, de uma forma organizada, podemos dizer que os arrays são “super variáveis”.

Como criar um array

Para criar um array, primeiramente você vai criar uma variável, e vai atribuir para ela um par de chaves([]).

Dentro desse conjunto de chaves, você pode armazenar os valores que quiser, todos separados por vírgula. Veja o exemplo.

```
<?php
$meuArray = ["Senai","de","limeira",60,"anos"];
// Para exibir um array por completo, usamos o comando vardump
var_dump($meuArray);
?>
```

Um array no PHP é uma estrutura de dados heterogênea, isso significa que ele pode guardar vários dados de tipos diferentes no mesmo lugar, como no exemplo acima, onde eu tenho várias strings e um texto.

Também como você pode observar, que para exibir o array eu estou utilizando o comando `var_dump`, que serve para mostrar informações de uma variável ou array em detalhes, é importante ressaltar, que quando você quiser exibir um array por inteiro na tela, o comando `echo` não vai funcionar.

```
<?php
$php = array("Zend" => "CERTIFICAÇÃO", 6 => 20);
echo $php["Zend"]; // CERTIFICAÇÃO
echo "<br>";
echo $php[6]; // 0

// Zend é nossa chave e CERTIFICAÇÃO nosso valor
// 6 é nossa chave e 60 é nosso valor
?>
```

Note que nossa primeira chave se chama Zend, e a outra chama-se 6, mas quanto a nomeação de chaves de array pode ser tanto string ou um integer. Para o valor pode ser qualquer coisa.

Acessando dados únicos

Você pode estar se perguntando: “E como faço para pegar um dado único dentro do meu Array?”.

E para fazer isso, primeiro você precisa entender que os arrays trabalham com um sistema de índices, todo valor que você coloca dentro de um array é “endereço” dentro dele, e esse endereçamento começa a partir do número 0.

Ou seja, isso quer dizer que, o índice do primeiro valor do array é 0, do segundo é 1, do terceiro é 2, e assim vai.

Array com chaves

Além de você poder acessar dados de um array por índices, você também pode definir um nome personalizado para cada dado

Array multidimensionais(Matriz)

Como já foi dito anteriormente, os arrays são estruturas heterogênea que permitem que você salve múltiplos dados de tipos diferentes em um mesmo lugar.

Seguindo esta ideia, os arrays no PHP, permitem que você também guarde arrays dentro de arrays, e isso forma a estrutura de uma matriz, veja o exemplo abaixo.

```
<?php
$minhaMatriz = [
    [10,30],
    ["Diogo","Barbosa"],
    ["Senai","de","limeira"]
];

// Quero acessar o valor 30 do primeiro array!
// como posso fazer isso?
// Simples!
echo $minhaMatriz[0][1];
// O primeiro 0 significa que eu quero acessar o primeiro array da matriz
// E o segundo número[1] significa que eu quero acessar o segundo elemento do primeiro array

echo "<br>"; //Quebra de linha

echo $minhaMatriz[2][2];
// Neste outro caso, eu estou acessando o último elemento do último array
?>
```

Conversão de tipos

Os tipos de variáveis no PHP são dinâmicos. Para forçarmos os tipos de nossas variáveis utilizamos uma técnica conhecida como type casting, ou simplesmente troca de tipos.

```
<?php
$var = 100;
$type_casting = (bool) $var; // torna - se booleano
$type_casting = (int) $var; // torna - se inteiro
$type_casting = (float) $var; // torna - se float
$type_casting = (string) $var; // torna - se string
$type_casting = (array) $var; // torna - se array
echo $type_casting = (bool)$var; // 1
?>
```

Explode no PHP

Explode é uma função do PHP, que serve para transformar um texto em um array, é bastante útil quando precisamos fazer algum tipo de manipulação mais avançada com string, veja como usá-lo neste exemplo:

```
<?php
$frase = "O SENAI é a melhor Escola";
$arrayDeTexto = explode(" ", $frase);
/* No Explode, primeiro você define por qual carácter você quer separar a string
pode ser por vírgula(,) ponto(.), ou seja por qualquer caracter.
E também pode ser pro espaço( que é representado por um espaço em branco)

Neste caso eu estou separando a string por espaço
Então a cada espaço que encontrar o PHP vai dividir as palavras
Veja o resultado
*/
var_dump($arrayDeTexto);
?>
```

Implode no PHP

O Implode é uma função do PHP que faz exatamente o contrário do explode, com o implode você consegue transformar um array em uma string, veja o exemplo:

```
<?php
$arrayDeTexto = ["SENAI","de","Limeira","é","a","melhor","Escola!!"];
$frase = implode(" ", $arrayDeTexto);
// Neste caso, eu quero adicionar um caracter de espaço(" ") a cada palavra do array!
echo $frase;
?>
```

Operadores Aritméticos no PHP

Podemos utilizar operadores matemáticos para efetuar cálculos com os valores de variáveis,

```
<?php
$a = 3;
$b = 3;
$c = $a * $b; // resultado é 9
$d = $a + $b; // resultado é 6
$e = $c - $d; // resultado é 3
?>
```

Criamos as variáveis \$a e \$b e a partir delas conseguimos fazer vários cálculos matemáticos.

Os operadores matemáticos disponíveis em PHP são:

Adição: +

Subtração: -

Multiplicação: *

Divisão: /

Módulo: %

Operadores de Atribuição no PHP

Utilizamos os operadores de atribuição para definir variáveis e seus valores, além de usá-los juntamente com os operadores matemáticos.

```
<?php
$a = 1; // A variável $a é igual a 1
$a += 2; // Somamos 2 ao valor da $a;
echo $a;
$a -= 2; // Subtraímos 2 ao valor da variável $a;
$a *= 2; // Multiplicamos o valor da variável $a por 2;
$a /= 2; // Dividimos o valor da variável $a por 2.
echo ++$a; // Incrementamos 1 e retornamos o valor
echo $a++; // Retornamos o valor e incrementamos 1
echo --$a; // Decrementamos 1 e retornamos o valor
echo $a--; // Retornamos o valor e decrementamos 1
?>
```

Operadores Relacionais

Esses são usados para comparar valores ou expressões, retornando um valor booleano (true ou false):

- Igual: ==
- Idêntico: ===
- Diferente: != ou <>
- Menor que: <
- Maior que: >
- Menor ou igual: <=
- Maior ou igual: >=

É importante lembrar que == não checa o tipo da variável, apenas seu valor. Já o === checa tanto o valor da variável quanto o seu tipo.

Operadores Lógicos

Existem também os operadores lógicos para a criação de testes condicionais:

- \$a and \$b: enquanto A e B forem verdadeiros;
- \$a or \$b: enquanto A ou B forem verdadeiros;
- \$a xor \$b: enquanto A ou B forem verdadeiros, mas não os dois;
- !\$a: verdadeiro se A for falso;
- \$a && \$b: enquanto A e B forem verdadeiros;
- \$a || \$b: enquanto A ou B forem verdadeiros.

Estrutura de Decisão if/else

A condição é avaliada para que, caso algo seja verdadeiro, faça isto, senão, faça aquilo

```
<?php
$idade = 17;

if($idade < 18) {
    echo 'Você não pode entrar aqui!';
} else {
    echo 'Seja bem - vindo';
}
?>
```

Na linha 1 criamos a variável \$idade que guarda um inteiro.

Na linha 4 utilizamos IF para verificar se \$idade é menor que 18, e caso seja será impresso: "Você não pode entra aqui!"

Na linha 6 criamos um ELSE, que é o contrário da primeira condição.

Podemos também utilizar os operadores lógicos junto dos operadores relacionais,

```
<?php
$idade = 21;
$identidade = true;

if($idade > 18 && $identidade == true) {
    echo 'Seja bem-vindo!';
}
?>
```

Switch

```
<<?php
$nome = 'Fulano';

switch($nome) {
    case 'Fulano':
        echo 'E ai Fulano!';
        break;

    case 'Sicrano':
        echo 'E ai Sicrano!';
        break;

    case 'Beltrano':
        echo 'E ai Beltrano!';
        break;

    default:
        echo 'Qual é o seu nome?';
        break;
}

// Resultado é: E ai Fulano!
?>
```

No exemplo foi criada \$nome declarando a ela uma string. É verificado a variável passada em switch entre parênteses: caso o valor contido na variável seja o que estiver em "case", será impresso o que conter no echo já pré-definido, e o break para a verificação. Caso não seja na primeira condição, será verificado todas as demais condições até chegar na última, que perguntará o seu nome. Na sintaxe básica do Switch atente-se sempre ao ":" no case e os ";" depois do echo e break.

Operador Ternário no PHP

No PHP existe uma forma mais curta de criar condições através do Operador Ternário. Exemplo:

```
<?php
$number1 = 1;
$number2 = 2;

if($number2 > $number1) {
    $a = 'Número 2 é maior que número 1';
} else {
    $b = 'Número 2 não é maior que número 1';
}

$ternario = ($number2 > $number1) ?
'Número 2 é maior que número 1' : 'Número 2 não é maior que número 1';

echo $ternario; // Número 2 é maior que número 1
?>
```

Criamos duas variáveis e a partir delas fizemos verificação com if/else e da forma ternária também. Na \$ternario passamos a condição que tem o mesmo valor que o if entre parênteses e o ponto de interrogação faz a pergunta: \$number2 é maior \$number1?

Caso seja, será impresso o que vier depois desse ponto. Caso \$number2 não seja maior que \$number1, será impresso o que estiver depois dos “:” que tem o mesmo valor que else.

Laços de Repetição

Os loops no PHP são estruturas de controle muito importantes que permitem efetuar um laço de repetição enquanto uma determinada condição for verdadeira.

Existem quatro tipos de loop no PHP são eles: while, do while, for e foreach.

While

O while permite que executemos um bloco de código enquanto a expressão passada como parâmetro for verdadeira,