

## Estudos para a prova DESENVOLVIMENTO WEB

O php foi criado em 1994 por RASMUS LERDOF com o objetivo de ter dados estatísticos sobre os acessos ao site pessoal.

A equipe do PHP resolveu chamar a nova versão de PHP 7. Sendo assim não existiu uma versão PHP 6 lançada oficialmente.

A versão do PHP 7 foi um marco tão importante, pois além de remover várias funções e comandos obsoletos a linguagem ganhou um novo motor, que se mostrou cerca de nove vezes mais rápido que sua versão anterior.

PHP é uma linguagem interpretada, livre, usada para o desenvolvimento de aplicações que possuem conteúdo dinâmico e que funciona no lado servidor (server side).

Client-side, "lado do cliente", ou seja, aplicações que rodam no computador do usuário.

Basicamente HTML, CSS, JavaScript, etc.

Server-side, "lado do servidor", ou seja, aplicações que rodam no servidor, como PHP, Ruby, Java, JavaScript, Python, Perl, etc.

Resumindo, PHP é uma linguagem que permite o acesso ao banco de dados e torna as páginas HTML dinâmicas. Ele não tem relação com o layout da página HTML e é invisível ao usuário, pois, sua função é gerar código HTML.

Com o php podemos

- gerar conteúdo de página dinâmica
- criar, abrir, ler, escrever, excluir e fechar arquivos no servidor
- coletar dados de formulário
- enviar e receber cookies
- adicionar, excluir, modificar dados em seu banco de dados
- controlar o acesso do usuário
- criptografar dados

Em resumo, a linguagem PHP não está limitada a produzir HTML, é possível gerar imagens, arquivos PDF, JSON, XML entre outros.

Um script PHP pode ser colocado em qualquer lugar do documento. O código PHP necessariamente começa “<?php” e termina com “>”

Obs: NOMES DE VARIÁVEIS DIFERENCIAM MAIÚSCULAS DE MINÚSCULAS

primeira pagina criando um olá mundo:

```
<?php
$ola = "Ola mundo";
echo $ola
?>
```

o comando phpinfo(); mostra as informações do seu PHP

a linguagem php suporta dois tipos de comentarios:

```
//
#
```

a diferença entre echo e print, o echo não tem valor de retorno, enquanto print tem valor de retorno 1 para que possa ser usado em expressões.

O comando echo é ligeiramente mais rápido que o print

## Aula03

O que são variáveis?

São identificadores utilizados para representar valores mutáveis que só existem durante a execução do programa.

Elas são armazenadas na memória RAM e seu conteúdo é destruído após a execução do programa.

De forma resumida, variáveis são "contêineres" para armazenar informações.

No php uma variavel começa com um cifrão \$

OBS:

→ Um nome de variável não pode começar com um número

→ Um nome de variável pode conter apenas caracteres alfanuméricos e sublinhados (underline) (Az, 0-9 e \_)

→ Os nomes das variáveis diferenciam maiúsculas de minúsculas (case sensitive)

O PHP suporta os tipos de dados: string, int, float boolean, Array, Object, NULL e Resource.

Importante, há uma diferença significativa na declaração de uma string com aspas duplas ou aspas simples. Todo conteúdo dentro de aspas duplas é avaliado pelo PHP. Assim, se a string contém uma variável, esta será traduzida para seu valor.

```
1  <?php
2
3  $fruta = "Banana";
4
5  echo "$fruta não é mistura!";
6  echo '$fruta não é mistura!';
7
8  ?>
```

Um Array é um tipo de dados que armazena vários valores em uma única variável.

As funções `mysqli_connect()` e `pg_connect()`, por exemplo, ao conectarem-se ao banco de dados, retornam uma variável de referência do tipo recurso.

Constantes são como variáveis, exceto que, uma vez definidas, elas não podem ser alteradas ou indefinidas.

Sua função é armazenar valores que não podem/devem sofrer modificações durante a execução do programa. Assim como as variáveis elas possuem identificadores porém elas não tem o símbolo do "\$".

As regras de nomenclatura de constantes seguem as mesmas regras das variáveis (com exceção do sinal cifrão) e por convenção utilizam letras maiúsculas.

Para criar uma constante, usamos a função `define()`

método get é o método padrão, quando não for especificado nenhum valor no atributo method os dados serão enviados usando GET

Quando o método GET é usado os dados do formulário são enviados de forma visível no campo de endereço da página.

O método GET dá origem a uma matriz de variáveis (\$\_GET) passadas para por meio dos parâmetros de URL.

Método POST e envia os dados colocando-os no corpo da mensagem. Ele deixa a URL separada dos dados que serão enviados.

As informações enviadas com o POST são invisíveis para outros (todos os nomes/valores são incorporados no corpo da solicitação HTTP) e não têm limites na quantidade de informações a serem enviadas.

## **Aula05:**

### **Operadores :**

O php divide os operadores nos seguintes grupos:

- operadores aritméticos.
- Operadores de atribuição
- Operadores de comparação
- operadores de incremento/decremento
- Operadores lógicos
- Operadores de string
- Operadores de matriz
- Operadores de atribuição condicional .

Operadores aritméticos

Os operadores aritméticos do PHP são usados com valores numéricos para realizar operações aritméticas comuns, como adição, subtração, multiplicação etc.

Os operadores de atribuição PHP são usados para definir o valor de uma variável. O operador básico de atribuição é “=”.

Os operadores de comparação do PHP são usados para comparar dois valores (números ou strings).

Operador "nave espacial" (spaceship) \$x <=> \$y

Operador "nave espacial" (spaceship) O operador nave espacial é utilizado para comparar duas expressões. Ele retorna -1, 0 ou 1 quando \$x for respectivamente menor que, igual a ou maior que \$y

Os operadores de incremento são usados para incrementar o valor de uma variável enquanto os operadores de decremento são usados para decrementar o valor de uma variável.

Os operadores lógicos do PHP são usados para combinar instruções condicionais.

Estruturas condicionais IF, ELSE E ELSEIF

O PHP suporta as seguintes declarações condicionais:

if – executa algum código se uma condição for verdadeira.

if...else – executa algum código se uma condição for verdadeira e outro código se essa condição for falsa

if...elseif...else – executa códigos diferentes para mais de duas condições

switch – seleciona um dos muitos blocos de código a serem executados

**IMPORTANTE!!!!**

Quando usado com strings o comando switch ele executa por padrão uma comparação do tipo “==” (igual, não identico)

## **Aula06**

WHILE - percorre um bloco de código enquanto a condição especificada for verdadeira.

do...while – percorre um bloco de código uma vez e, em seguida, repete o loop enquanto a condição especificada for verdadeira.

For- percorre um bloco de código um numero especificado varias vezes

foreach – percorre um bloco de código para cada elemento em uma matriz

## **Aula07**

Inclusão de arquivos (include e require)

Quando nos desenvolvemos uma aplicação web com PHP algumas funcionalidades ou trechos das páginas podem ser necessarios com grande frequencia, tornando-se ótimos candidatos a separação de código visando a inclusão nos scripts.

Vamos imaginar a seguinte situação, uma função que dá boas vindas ao usuario, ou faz algum calculo importante que deve ser utilizado em diferentes partes de uma aplicação.

Outro exemplo clássico é a divisão de componentes que se repetem como cabeçalho (header), rodapé (footer), menus e outros componentes que podem se repetir em todas as páginas da aplicação em arquivos separados..

Para fazer a inclusão dos arquivos podemos usar os comandos:

- include
- require
- include\_once
- require\_once

o include inclui e avalia o arquivo informado

As instruções include e require são idênticas, exceto em caso de falha:

- require produzirá um erro fatal (E\_COMPILE\_ERROR) e interromperá o script
- include produzirá apenas um aviso (E\_WARNING) e o script continuará

Por tanto, se você deseja que a execução continue e mostre aos usuarios a saída, mesmo que o arquivo de inclusão esteja ausente, use a instruções include.

Caso contrário, no caso de um Framework, CMS ou uma codificação de aplicativo PHP complexa, use a instrução require para incluir um arquivo essencial no fluxo de execução.

O `include_once` pode ser útil em casos em que o mesmo arquivo pode ser incluído e avaliado mais de uma vez durante uma execução de um script, ajudando a evitar problemas como redefinição de funções, reatribuições de valores de variáveis e etc.

O `require_once` faz exatamente a mesma coisa do `include_once` porém como a diferença no tratamento de erros (interrompe o script em caso de ausência do arquivo a ser incluído). É importante frisar que todas as instruções utilizadas para requisição de arquivos necessitam que seja especificado o caminho para a localização do arquivo a ser incorporado.

Em um curso na Udemy eu vi o seguinte sobre `Require` e `include`:  
`Include` e `Require`.

- `Require` ele obriga que o arquivo exista e que ele esteja funcionando corretamente o arquivo de inclusão, se ele não estiver funcionando ele gera um erro fatal e vai parar o seu código.
- `Include` ele tenta funcionar mesmo que o arquivo não exista, além dele ter mais recursos.

`Require_once` e `Include_once`, traz apenas uma vez no PHP7 não tem mais erros fatais!

## **Aula 7- B**

Git e Git Hub não são a mesma coisa!!!!

O GIT é um sistema de controle de versão de arquivos.

É um software livre e muito utilizado no desenvolvimento de software.

Um sistema de controle de versão ajuda a acompanhar as mudanças feitas no código base. E mais, ele também registra quem efetuou a mudança e permite a restauração do código removido ou modificado.

Sempre quando alguém disponibiliza sua parte do projeto no GIT, ele gerencia as alterações feitas e guarda um histórico. Isso é importante pois se houver algum problema você pode desfazer as alterações e voltar para a versão que estava estável.

Ele foi projetado e desenvolvido por Linus Torvalds para o desenvolvimento do Linux e foi adotado por muitos projetos.

Existem sites que provêm hospedagem gratuita de código fonte para repositório GIT, um deles é o GitHub

O GitHub é um repositório remoto, criado como um serviço online de hospedagem de repositórios do Git.

Funciona como um servidor que agrega todas as modificações realizadas por cada uma das pessoas envolvidas em um projeto, unificando as diferentes versões de código e seus históricos, permitindo compartilhamento entre as equipes.

O Git foi lançado em 2005 hoje ele tem 17 anos de existência e encontra na versão 2.36

Este branch é como se fosse uma cópia, um espelho do projeto original. Agora você trabalha apenas nesse branch, até acertar todos os detalhes dele. Após isso, você pode fazer um merge de volta do seu branch até o projeto original.

O git possui algumas diretivas (comandos) muito utilizados como o `ADD`, o `commit`, o `pull` e o `push`

**Repositório** É o local físico onde os arquivos e suas cópias são armazenados. Pode ser local ou remoto, podendo salvar não apenas arquivos de texto, mas também imagens, áudios e outros elementos relacionados ao projeto.

**Branch** Branches são os ramos, cópias do código original que podem ser manipuladas, sem afetar o código-fonte do projeto original (ou as funcionalidades do ambiente de produção).

**Merge** Após a finalização de uma alteração em um Branch, é necessário realizar o Merge, ou seja, fundir a cópia dos arquivos modificados com o ramo principal do projeto. Isso acontece no repositório local para que ajustes possam ser feitos

**Push Request** O Push Request é o envio das modificações após o Merge para o repositório central, para que todas as outras pessoas que atuam no desenvolvimento possam atualizar suas cópias e revisar o código criado, verificando conflitos com seus próprios trabalhos.

**Pull Request** O Pull Request é utilizado quando outra pessoa que atua no desenvolvimento muda o ramo principal no repositório central, puxando as modificações realizadas para a sua máquina, fundindo a nova versão com o seu código local.

**Fork** Comando que copia um repositório remoto para uma máquina local, realizado sempre que vamos começar a trabalhar em um projeto que já existe. Também é usado para pegar um código público para modificação e utilização em um novo projeto.

A programação procedural foi a abordagem mais utilizada pelos programadores PHP durante muito tempo.

Ela consiste basicamente na construção de procedimentos (funções ou páginas isoladas) que resolvem determinado problema.

Na programação procedural é possível ter um reaproveitamento de código, porém devido a sua natureza, o reúso de procedimentos torna-se mais complicado.

## **Aula09-a)**

Em php a definição de uma classe começa com a palavra-chave class, seguida do nome da classe, um par de colchetes que englobam as definições de propriedades e métodos que englobam as definições de propriedades e métodos pertencentes à classe.

O encapsulamento é um recurso interessante da orientação a objetos, pois ele provê um mecanismo de proteção de acesso às características de uma determinada classe.

As três opções de encapsulamento ou visibilidade são public, private ou protected cada uma prove diferentes formas de acesso.

**Public**, estes atributos podem ser acessados livremente na própria classe ou em classes dependentes a partir de programas que instanciam esta classe. É a visibilidade mais permissiva.

**Private**, de forma resumida é o contrário do public. Atributos private podem ser acessados apenas na própria classe, não sendo permitido sua manipulação direta e sim por métodos “get” “set”.

**Protected**, atributos deste tipo podem ser manipulados em sua própria classe e em classes descendentes, porém não podem ser manipulados diretamente em programas que fazem instâncias desta classe, sendo necessário os “gets” e “sets”

Para acessar um método estático pode-se utilizar o nome da classe seguido por dois sinais e dois pontos(::)

O composer é um gerenciador de dependências de programas em PHP

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.

Ele permite que você declare quais bibliotecas são necessárias para a execução de um projeto e ele gerencia (instala, atualiza ou remove) para você.

É possível usar o composer para gerenciar pacotes de diversas fontes, no entanto, é mais comum que ele busque no packagist.

Packagist é o repositório de pacotes padrão do Composer. Ele permite que você encontre pacotes e que o Composer saiba de onde obter o código.

## LARAVEL:

“Laravel é um framework de aplicação web com sintaxe expressiva e elegante. Nós já lançamos as bases - liberando você para criar sem se preocupar com as pequenas coisas.”

O Laravel possui uma série de características interessantes no contexto do desenvolvimento web:

Progressivo

Com testes da unidade integrados

Os arquivos de views fica no diretório resources\views\ eles são arquivos do Blade. Observe a extensão .blade.php

O Blade é um sistema de templates (template engine) que tem dentre suas vantagens o isolamento do código PHP, o melhor desempenho das páginas e a maior facilidade da codificação da camada de visualização

Crie então um novo arquivo no diretório resources\views\ com algum nome, como por exemplo pag1.blade.php

Se tentarmos abrir esse arquivo diretamente no navegador veremos um erro, pois o Laravel precisa de uma rota para encontrar esses arquivos.

As rotas são responsáveis por capturar os endereços da URL e localizar o arquivo ou recurso correspondente. As rotas ficam no arquivo routes\web.php.