

PROGRAMAÇÃO PARA WEB

Linguagem PHP

MSc. Leonardo Rocha

- Unidade de Ensino: 04
- Competência da Unidade: Conhecer e ser capaz de implementar aplicações PHP
- Resumo: Criar e compreender códigos em PHP e entender a importância de sua utilização do Lado do servidor.
- Palavras-chave: PHP, sintaxe, estrutura, funções e métodos
- Título da Teleaula: Linguagem PHP

Teleaula nº: 04

SP1

Uma necessidade muito comum quando se está iniciando a utilização do PHP, é criar código para coisas simples, como: um sistema de login para usuários. Você deve criar um utilizando PHP mas sem se preocupar neste momento com conexão com banco de dados.

SP2

Você deverá criar um código PHP que tem por intenção realizar o login em um sistema, ou seja, é um sistema de autenticação de usuário que faz uma consulta à base de dados.

Conceitos

Introdução ao PHP

O que é PHP?

É um acrônimo recursivo para PHP Hypertext Preprocessor. É uma linguagem de script aberta, é adequada e muito utilizada no desenvolvimento de páginas web e pode ser embutida dentro do HTML. É multiplataforma, ou seja, roda na maioria dos sistemas operacionais como: Linux, Solaris, OpenBSD, Windows, MacOS, RISC OS e outros.

PHP

O código PHP é delimitado pelas instruções de processamento ou as famosas tags que servem para iniciar e encerrar. No PHP essas instruções são:

Início `<?php`

Fim `?>`

Ficando assim: `<?php instrução ?>`

PHP

Focado nos scripts que rodam do lado do servidor. Isso dá uma liberdade para utilização do PHP. Você pode coletar dados de formulários e gerar páginas com conteúdos dinâmicos. As áreas onde pode ser utilizado são:

Script do lado do servidor

Scripts de linha de comando e

Aplicações Desktop

Suporte a banco de dados

Apresenta amplo suporte a variados bancos de dados. Alguns deles são:

Dbase

Mongo

PostgreSQL

SQLite3

MySQL

Sintaxe

Pode-se definir uma variável com o uso da função `define()` ou a palavra-chave *const*

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // imprime "Hello world."
echo Constant; // imprime "Constant" e gera um alerta notice.
?>
```

Fonte: <https://bit.ly/3d1yk9q>

Sintaxe

A definição de uma constante é possível utilizar qualquer expressão escalar (boolean, integer, float e string) a partir da versão 5.6 do PHP. Podem ser ainda do tipo composta:

array

object

```
// A partir do PHP 7
define('ANIMALS', array(
    'dog',
    'cat',
    'bird'
));
echo ANIMALS[1]; // imprime "cat"
```

Declaração de Variáveis

As variáveis podem ter o nome definido e utilizado dinamicamente. Um exemplo de definição de variável é:

```
<?php
```

```
$msg = "Olá mundo";
```

```
echo "$msg";
```

```
?>
```

```
$nome = 'Meu nome 123';
```

```
$ano = 2017;
```

```
$pi = 3.14159265;
```

```
$sim = true;
```

Operadores Aritméticos

$+\$a$ – conversão de a para inteiro ou float

$-\$a$ – negação

$\$a + \b – Adição

$\$a - \b – Subtração

$\$a / \b – Divisão

$\$a * \b – Multiplicação

$\$a \% \b – Modulo

$\$a ** \b – Exponencial

Operadores de comparação

$\$a == \b – igual

$\$a === \b – igual e idêntico (mesmo tipo)

$\$a \neq \b – Diferente

$\$a <> \b – Diferente

$\$a !== \b – Não idêntico

$\$a < \b – Menor que

$\$a > \b – Maior que

$\$a \leq \b – Menor ou igual

$\$a \geq \b Maior ou
igual



Operadores lógicos

$\$a \text{ and } \$b - E$

$\$a \text{ or } \$b - OU$

$\$a \text{ xor } \$b - XOR$

$! \$a - N\tilde{A}O$

$\$a \&\& \$b - E$

$\$a || \$b - OU$

Funções internas (built-in)

É uma função da própria linguagem. As mais conhecidas são:

mysqli_connect() - Função responsável pela conexão com bando de dados;

Phpinfo() - Mostra um volume grande de informações sobre o estado atual do PHP. Muito útil para identificar e resolver problemas.

Funções do usuário

Funções definidas pelo usuário podem ser construídas respeitando a seguinte sintaxe:

```
<?php
```

```
function foo ($val1, $val2, /* ..., */ $valn)
```

```
{
```

```
    echo "Exemplo de construção de função.\n";
```

```
    return $valor_retornado;
```

```
} ?>
```

Funções (Exemplo)

```
<?php
```

```
function exibir_mensagem($nome)
```

```
{
```

```
    echo "Olá\n";
```

```
    echo $nome;
```

```
}
```

```
exibir_mensagem("Leonardo Rocha");
```

```
?>
```

PHP no HTML

```
<!doctype html>
<html lang="en">
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>
    <?php
      if(isset($titulo)) {
        echo $titulo . ' | ';
      }
    ?>
    StackOverflow
  </title>
</head>
<body>

  <?php include('cabecalho.php') ?>

  <?php echo $conteudo ?>

  <?php include('rodape.php') ?>

</body>
</html>
```

Fonte: <https://bit.ly/395bcpB>

Expressões e estrutura de controle

Bloco de construção mais importante do PHP. Quase tudo que se escreve são expressões.

As estruturas de controle são muito semelhantes à existentes em outras linguagens. As principais e mais conhecidas são:

If, else, while, do-while, for, foreach e break.

Estrutura de controle (if)

```
<?php
$t = date("H");
if ($t < "20") {
    echo "Tenha um ótimo dia!";
    if ($t > "20") {
        echo "Tenha uma ótima noite";
    }
} ?>
```

```
<?php
if ($a > $b)
    echo "a é maior que b";
?>
```

Estrutura de controle (while)

```
$i = 1;
```

```
while ($i <= 10):
```

```
    echo $i;
```

```
    $i++;
```

```
?>
```

```
<?php
```

```
$a = 1;
```

```
while($a <= 5) {
```

```
    echo "O número é: $a";
```

```
    $a++;
```

```
}
```

```
?>
```

Estrutura de controle (for e continue)

```
?php
```

```
for ($i = 0; $i < 5; ++$i) {
```

```
    if ($i == 2)
```

```
        continue
```

```
    print "$i\n";
```

```
}
```

```
?>
```

Resolução da SP

SP1

Note que o sistema de login precisará de duas etapas. A primeira parte é o formulário com HTML5 para chamar outras rotinas. Veja na sequência:

```
<!doctype html>
<html>
  <head>
    <title>Sistema de Login - Cadastro Usuário</title>
    <!-- CSS only -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqN000vy
+HMDP7az0uL0xtbFIcaT9wjKhr8RbDVddVHyTfAAsrekWkmP1" crossorigin="anonymous">
    <!-- JavaScript Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kg800JFdroafW"
crossorigin="anonymous"></script>
  </head>
  <body>
    <?php
include("menu.php");
include("session.php");

if (isset($_GET["success"])) {
  echo "<br>
<div class='alert alert-success' role='alert'>
  Usuário cadastrado com sucesso!
</div>";
}
?>
<div class="container">
<label></label>
  <form class="form" method="POST" action="insert-usuario.php" enctype="multipart/form-data">
    <label>Nome:</label>
    <input type="text" class="form-control" name="nome" placeholder="Nome"/>

    <label>Foto de Perfil:</label>
    <input type="file" name="foto" class="form-control">

    <label>E-mail:</label>
    <input type="email" class="form-control" name="email" placeholder="Email"/>

    <label>Senha:</label>
    <input class="form-control" type="password" name="senha" placeholder="Senha"/>
  <br>
    <button type="submit" class="btn btn-primary">Cadastrar</button>
  </form>
</div>
</body>
</html>
```

```
<?php
```

```
include("conexao.php");
```

```
$nome = $_POST["nome"];
```

```
$email = $_POST["email"];
```

```
$senha = $_POST["senha"];
```

```
$foto = $_FILES["foto"];
```

```
$caminho_pasta = "fotoPerfil/";
```

```
$caminho_arquivo = $caminho_pasta.$foto["name"];
```

```
move_uploaded_file($foto['tmp_name'], $caminho_arquivo);
```

```
mysql_query("INSERT INTO usuario (nome, email, senha, foto) VALUES('$nome', '$email', '$senha', '$caminho_arquivo');")
```

```
or die (mysql_error());
```

```
header("location:cadastro-usuario.php?success");
```

```
?>
```

Interação

Dúvidas?

Conceitos

Conhecendo o PHP

Tratamento de arquivos

Uma das características da linguagem PHP é o fato de ela poder acessar arquivos e, por exemplo, escrever nele. Essa possibilidade a torna indispensável em tarefas dessa natureza. É possível realizar esse procedimento apenas informando o nome do arquivo a ser acessado e o modo como ele será aberto.

Tratamento de arquivos

Alguns modos de abertura mais utilizados:

w – abre o arquivo para escrita gravando o conteúdo a partir de seu início. Se ele existir, o conteúdo é apagado, se não, ele é criado.

r – abre o arquivo para leitura e se ele não existir, apresenta um erro.

a – abre o arquivo somente para escrita iniciando a partir do final do mesmo.

Manipulação de arquivos

Funções para isso:

`fopen` – abre o arquivo

`fwrite` – escreve no arquivo

`fread` – lê um arquivo a partir do tamanho em bytes informado.

`fclose` – fecha o arquivo ao final.

Manipulação de arquivos

A criação de um arquivo pode ser feita com a função `fopen` passando o modo `w`. Vale ressaltar que é possível abrir arquivos remotos com essa função. Basta ativar a diretiva `allow_url_fopen` do PHP:

```
<?php $file_handle = fopen('https://your-website-url/tmp.txt', 'r');?>
```

Manipulação de arquivos

fgets – função para leitura de apenas uma linha de um arquivo. **feof** verifica o final do arquivo.

```
<?php
$handle = @fopen("/tmp/arquivodeentrada.txt", "r");
if ($handle) {
    while (($buffer = fgets($handle, 4096)) !== false) {
        echo $buffer;
    }
    if (!feof($handle)) {
        echo "Erro: falha inesperada de fgets()\n";
    }

    fclose($handle);
}
?>
```

Fonte: <https://bit.ly/3f0dv0S>

Manipulação de arquivos

copy – copia arquivos informando origem e destino.

unlink – apaga um arquivo

chmod – modifica permissões do arquivo

chown – modifica o dono do arquivo

file – lê todo o arquivo para um array

filesize – obtém o tamanho do arquivo

```
<?php
$file = 'texto.txt';
$newfile = 'texto.txt.bak';
if (!copy($file, $newfile)) {
    echo "falha ao copiar $file...\n";
} ?>
```

Upload de arquivos

```
<form enctype="multipart/form-data" action="__URL__" method="POST">  
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />  
  Selecionar: <input name="userfile" type="file" />  
  <input type="submit" value="Enviar arquivo" />  
</form>
```

Selecionar: Nenhum arquivo selecionado

Upload de arquivos

```
<?php
$uploaddir = '/var/www/uploads/';
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);

echo '<pre>';
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
    echo "Arquivo válido e enviado com sucesso.\n";
} else {
    echo "Possível ataque de upload de arquivo!\n";
}

echo 'Aqui está mais informações de debug:';
print_r($_FILES);

print "</pre>";

?>
```

Fonte: <https://bit.ly/3f0f1A6>

Upload de arquivos

É possível usar a variável `$_FILES['userfile']['size']` para, por exemplo, descartar qualquer arquivo que seja pequeno demais, evitando assim um ataque hacker.

Ou ainda, bloquear determinado tipo de arquivo com a variável `$_FILES['userfile']['type']`.

SGBD

Sistema Gerenciador de Banco de Dados é responsável por gerenciar dados que serão armazenados, controlar acesso a eles e garantir a consistência e integridade desses dados. Existem dois tipos de bando de dados: os relacionais e os não relacionais.

Relacional e não Relacional

Banco de dados não relacional é o tipo de banco de dados que não utiliza esquema de tabela de linhas e colunas como uma quantidade significativa de bancos de dados utiliza. São conhecidos como banco de dados NoSQL, isto é, não utiliza SQL para tratamento dos dados. O MySQL é um banco de dados relacional e o MongoDB é um banco de dados não relacional.

Exemplo (não relacional)

```
[{
  "_id": ObjectId("5e6261a1df9bcf90c29726d4"),
  "nome": "Henrique Marques Fernandes",
  "idade": 29
},
{
  "_id": ObjectId("5e6261a1df9bcf90c29726d3"),
  "nome": "Terry Crews",
  "idade": 65,
  "pais": "USA"
}]
```

MySQL e seus comandos

Banco de dados relacional que é mantido por organização de mesmo nome e de código aberto. Tem como linguagem o SQL, ou seja, é através dele que se manipula os dados contidos nesse banco de dados.

Select

Update

Delete

Insert into

Create table

Alter table

Exemplo de SQL

```
CREATE TABLE tb_clients{  
  ClientID int,  
  nome varchar(255),  
  endereco varchar(255),  
  cidade varchar(255)  
  ?;
```

```
SELECT nome, endereco, cidade  
FROM tab_clients  
WHERE cidade = 'Curitiba'  
ORDER BY nome;
```

Conexão

Para criar uma conexão com um banco de dados, existem bibliotecas no PHP que fazem esse trabalho. Imagine que você criou um formulário de contato numa página web e precisa guardar os dados desse formulário. Você vai utilizar um banco de dados para isso e o PHP para criar essa conexão. Nós já a conhecemos aqui, é o Mysqli quem fará esse trabalho.

Conexões

Abrir conexão:

```
$conexao = mysqli_connect($server, $username,  
$password)
```

```
return [  
    'class' => 'yii\db\Connection',  
    'dsn' => 'mysql:host=localhost;dbname=catalogo_clientes',  
    'username' => 'usuario',  
    'password' => 'senha',  
    'charset' => 'utf8',
```

Autenticação

Autenticação de usuário é um procedimento extremamente importante.

```
CREATE TABLE tb_clients(  
  'ID' int(11) NOT NULL auto_increment,  
  'nome' varchar(255) NOT NULL,  
  'login' varchar(255) NOT NULL,  
  'senha' varchar(255) NOT NULL,  
  'public' varchar(255) NOT NULL  
);
```

```
INSERT INTO 'tb_clients' VALUES (1,  
  "Leonardo Rocha", "rocha", 'senha', 'S');
```

```
INSERT INTO 'tb_clients' VALUES (1, "Joana  
Prado", "Joana", 'senha', 'N');
```

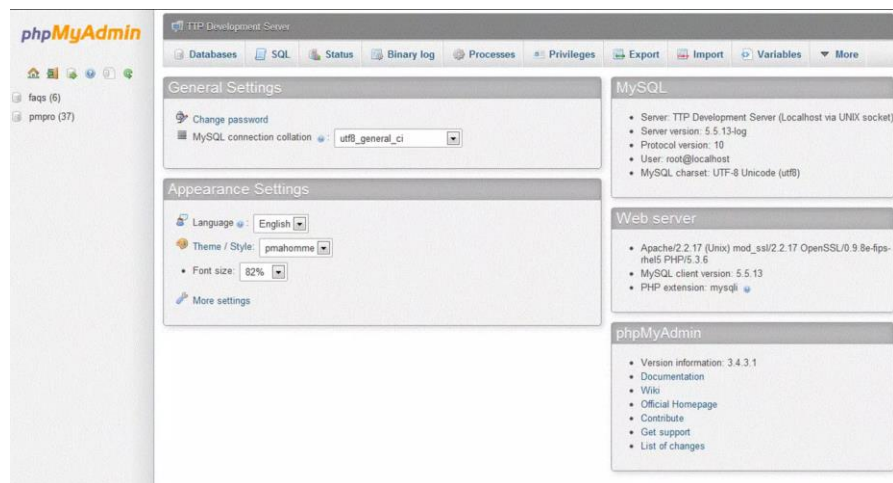
Cookies

Estes são responsáveis por guardar informações que são utilizadas para melhorar a experiência de navegação do usuário. São pequenos arquivos salvos no navegador que serão consultados por um site sempre que o usuário acessá-lo novamente. Sabe dizer algum exemplo de utilização de cookies em sites de e-commerce?

No PHP a função responsável por armazenar informações em Cookies é a **setcookie**.

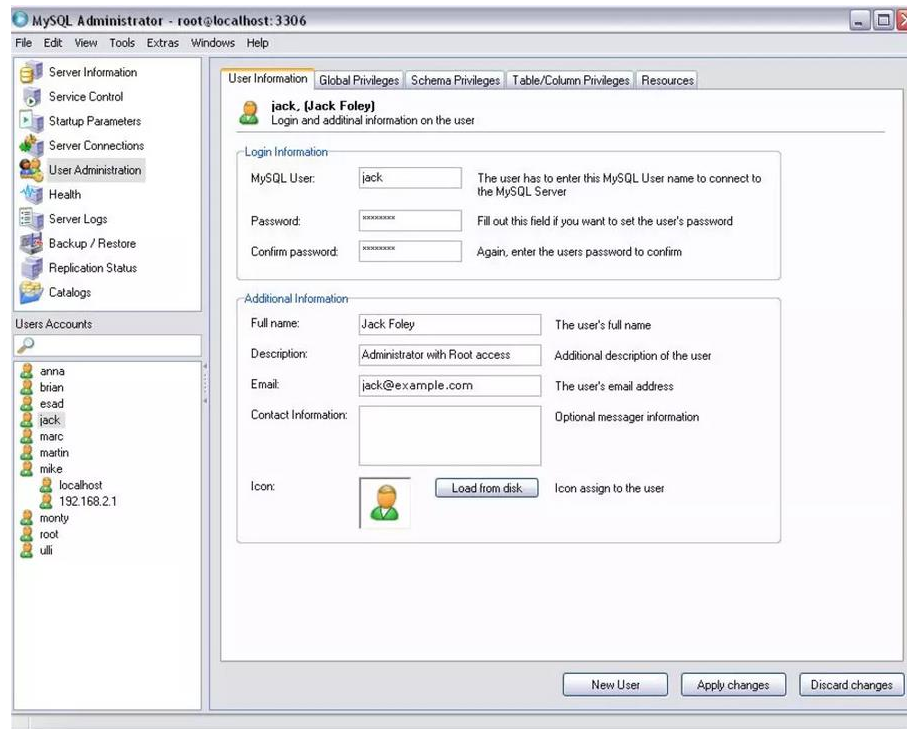
Ferramentas

PHPMyAdmin é uma ferramenta web desenvolvida em PHP que auxilia na administração de bancos de dados MySQL pela internet. Você pode, por exemplo, acessar um banco de dados que está em um servidor em algum lugar, da sua casa. Você pode fazer todas as operações por ele.



Ferramentas

MySQL Administrator é outra ferramenta útil na administração de banco de dados Mysql.



Ferramentas

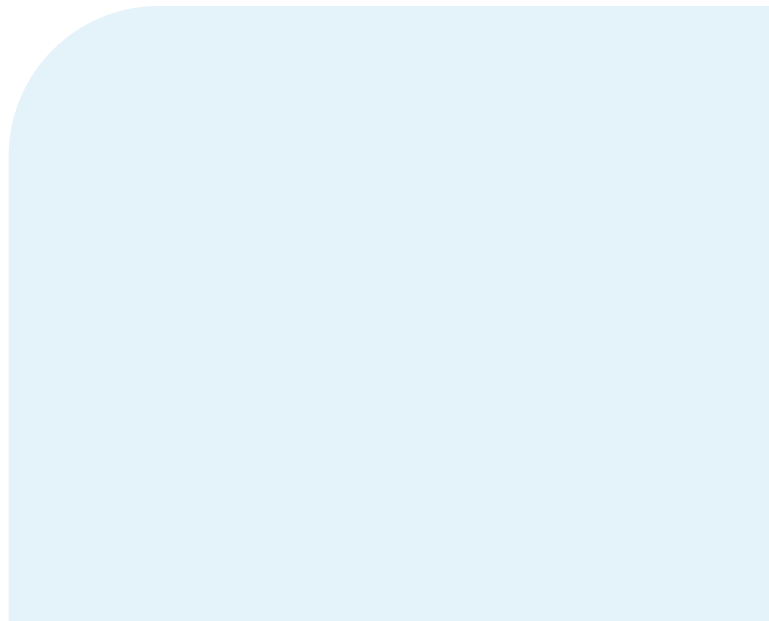
Sigla para *Asynchronous JavaScript and XML (AJAX)* é uma técnica que serve para carregar conteúdos é uma página web com uso de JavaScript e XML. Possibilita realizar requisições assíncronas.

Essa técnica de carregamento de conteúdos pode ser realizada com uso de JavaScript e XML, HTML, TXT, PHP, ASP, JSON ou qualquer linguagem de marcação ou programação que pode ser recuperada de um servidor.

Resolução da SP

SP1

À conexão à base de dados é realizada através de uma chamada de função que é responsável por realizar essa consulta ao banco de dados.



```
<?php
include("conexao.php");

$email = $_POST["email"];
$senha = $_POST["senha"];

$query = mysql_query("SELECT * FROM usuario WHERE email = '$email' AND senha = '$senha'") or die (mysql_error());

$num_rows = mysql_num_rows($query);

if ($num_rows > 0) {
    session_start();
    $_SESSION["email"] = $email;
    $_SESSION["senha"] = $senha;
    header("location:painel.php");
} else {
    header("location:index.php?login=1");
}

?>
```

Conceitos

Recapitulando

Recapitulando

- Estrutura, Sintaxe e delimitação em PHP
 - Operadores, variáveis e funções
 - Integração com HTML5
 - Consultas e conexão a banco de dados
-

