

# Desenvolvimento Web I

## PHP

Tiago Lopes Telecken

telecken@gmail.com



# Funções

- Sintaxe

```
function nome_da_funcao([arg1, arg2,...])  
{  
    Comandos...;  
    [return <valor de retorno>;]  
}
```

- As declarações dos tipos das funções (tipo do retorno) e dos tipos dos argumentos não são obrigatórias
- Para documentação e legibilidade do código os tipos podem ser declarados

# Funcoes

- Retorno
  - › Toda função pode retornar um valor, ou simplesmente executar os comandos.
  - › Uma função só pode retornar 1 valor. Entretanto pode-se retornar um valor composto, como objeto ou array.
- Argumento
  - › tornam-se variáveis pertencentes ao escopo local da função.

# Funções

```
<?
function soma($a,$b)
{
    $resultado = $a + $b;
    return $resultado;
}
echo soma(10,20); // mostra 30
echo " ---- ";
echo soma(4,6); //mostra 10
?>
```

# Funções

## ➤ Argumento padrão

<?

```
function funcao($linguagem="PHP"){  
    return "A aula será sobre $linguagem<br>";  
}  
echo funcao();                // A aula será sobre PHP  
echo funcao("JavaScript");    // A aula será sobre JavaScript  
?>
```

# Escopo das variáveis

- Variáveis definidas dentro das funções tem escopo local (só são reconhecidas dentro da função que a criou)
- Variáveis definidas fora de funções tem escopo global porém não são acessadas dentro das funções

```
<?php  
$php = "Variável global";  
function escreve(){  
    echo $php; //Não escreve nada  
}  
escreve();  
?>
```

# Escopo

//Para acessar variáveis globais nas funções...

```
<?php
```

```
$php = "Variável global1";
```

```
$php2 = "Variável global2";
```

```
function escreve(){  
echo $GLOBALS["php"];
```

```
global $php2;
```

```
echo $php2;
```

```
$php2="alterado2";
```

```
$GLOBALS["php"]="alterado";
```

```
}
```

```
escreve();
```

```
echo $php;      //alterado
```

```
echo $php2;     //alterado2
```

```
?>
```

# Variáveis estáticas

```
<?
function conta_chamadas() {
static $v=0;
$v++;
Echo "Esta função foi chamada $v vezes<br>";
}
conta_chamadas();
conta_chamadas();
conta_chamadas();
?>
```



# Include e Required

- **Include e require** – Permitem que um código php que está em um arquivo externo seja adicionado numa pg php antes dela ser executada
- **Include** – em caso de erro a execução do script continua (Warning)
- **Require** – em caso de erro a execução do script para (Error)

# Exemplo

- **Existe um arquivo php externo chamado "menu.php", com o seguinte código**

```
<a href="/default.php">Principal</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/referencias.php">Referencias</a>
<a href="/exemplos.php">Exemplos</a>
```

- **Outra pg chamada default.php pode ter o seguinte código**

```
<html><body>
    <nav> <?php include("menu.php"); ?></nav>
    <h1>Bem vindo.</h1>
    <p>Texto..texto..texto.</p>
</body> </html>
```

# Exemplo

- A página enviada ao navegador é a seguinte

```
<html><body>
<nav>
<a href="/default.php">Principal</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/referencias.php">Referencias</a>
<a href="/examplos.php">Examplos</a>
</nav>
<h1>Bem vindo.</h1><p>Texto..texto..texto.</p>
</body></html>
```

# Exemplo

- Na página abaixo um conjunto de funções php é colocada no início da página. Estas funções podem ser usadas no restante da página

```
<html>
<body>
<?php require("funcoes.php");?>

<?//chamadas as funções do arquivo funcoes.php?>
</body>
</html>
```

# require\_once, include\_once

```
<?php  
require_once 'cabecalho.php';  
require_once 'conteudo.php';  
require_once 'conteudo.php';  
require_once 'conteudo.php';  
require_once 'rodape.php';  
?>
```

- O arquivo só é incluído 1 vez. Depois ele é ignorado

# header

```
<?php  
header("Location: verifica.php");  
?>
```

- Redireciona o navegador para outra página. Deve ser colocado antes de qualquer saída (antes de qualquer echo por exemplo). O código abaixo gera um erro

```
<HTML>  
<?php  
header("Location: verifica.php");  
?>
```

# SESSION

- Utilizado para passar dados de uma página para outra
- GET - **\$\_GET**
- POST - **\$\_POST**
- SESSION - **\$\_SESSION**
- Os dados são eliminados quando o usuário fecha o navegador.

# Session

```
<?php
session_start(); // session_start inicia a sessão
$log = $_POST['login'];
$sen = $_POST['senha'];

if($log=='fulano' && $sen=='12' ) {
    $_SESSION['login'] = $log; // define variáveis de sessão
    header('location:site.php'); }
else{
    unset ($_SESSION['login']); //apaga as variáveis
    header('location:index.php'); }
?>
```



# Session

```
<?
session_start();
if((isset ($_SESSION['login']))
{
echo" Bem vindo".$_SESSION['login']; //usando uma variável ?>
}
else
{
    echo ('Usuário não autorizado');
}
```

# Session

```
session_start();  
if ($_POST["fecha"]=="sim") {  
    session_destroy();  
}
```

# PHP com BD - Postgres

- No postgres colocar os seguintes comandos

```
# create database loja;
```

```
# psql loja;
```

```
loja=# CREATE TABLE clientes (codigo integer, nome varchar(40));
```

- Rodar os seguintes códigos php

# PHP com BD

**Comando para conectar com o BD.  
A conexão deve ser armazenada em uma  
variável que depois será referenciada**

**Ajustar os parâmetros**

```
<?php  
$conn=pg_connect("host=127.0.0.1 port=5432 dbname='loja' user='postgres'  
password='postgres'");
```

```
pg_query($conn, "INSERT INTO clientes (codigo,nome) values (1,'Joao')");  
pg_query($conn, "INSERT INTO clientes (codigo,nome) values (2,'Jose')");  
pg_query($conn, "INSERT INTO clientes (codigo,nome) values (3,'Joana')");  
pg_query($conn, "INSERT INTO clientes (codigo,nome) values (4,'Maria')");  
pg_query($conn, "INSERT INTO clientes (codigo,nome) values (5,'Ana')");  
pg_query($conn, "INSERT INTO clientes (codigo,nome) values (6,'Mara')");
```

```
pg_close($conn);  
?>
```

**Comando para executar uma query no BD. O primeiro  
Parâmetro é a variável com a conexão. O segundo é  
um sql que será executado. Neste caso um INSERT**

# PHP com BD

```
<?php
$conn=pg_connect("host=localhost port=5432 dbname=loja password=");

$query='SELECT codigo,nome FROM clientes';
$result=pg_query($conn,$query);

if($result)
{
    while ($row=pg_fetch_assoc($result))
    {
        echo $row['codigo'].' - '.$row['nome']."<br>\n";
    }
}

pg_close($conn);
?>
```

**Quando o pg\_query  
Executa um select  
Ele retorna um vetor  
Onde as colunas são  
Os campos solicitados  
E as linhas são os  
Dados retornados.  
Neste caso o vetor é  
Armazenado em  
\$result. É preciso um  
Laço para percorrer os  
Dados. O comando  
pg\_fetch\_assoc coloca  
uma linha de dados em \$row  
\$row é um vetor cujo índice é  
O nome do campo e o valor é  
O dado**

# PHP com BD

No próximo exemplo é mostrado um código onde em uma mesma página:

- o usuário preenche os dados para inserir no BD.
- O php insere os dados no BD
- O php acessa o bd e mostra os dados armazenados na tabela

# Form + BD

```
<? // Se o form nao foi submetido.
if($submit != "Insert") {
?>
<title>Formulário Web</title>
<body>
  <form method="POST" action="form.php">
    Codigo: <input type="text" name="id"> <br />
    Username <input type="text" name="username"> <br />
    Primeiro Nome <input type="text" name="pname"> <br />
    Ultimo Nome <input type="text" name="uname"> <br />
    <hr />
    <input type="submit" name="submit" value="Insert">
  </form>
</body>
<?
} else {
  // o form foi submetido
  $connection = pg_connect("dbname=mydb user=myuser");

  // testa se a conexao falhou
  if (!$connection) {
    print("Connection Failed.");
    exit;
  }
}
```

```
// Insere os valores do formulario
pg_query ($conn, "INSERT INTO users VALUES
('$id','$username','$pname','$uname')");

// seleciona dados a partir do id
$myresult = pg_query($conn, "SELECT * FROM users
WHERE id='$id'");

// processa resultados
$query='SELECT codigo,nome FROM users';
$result=pg_query($conn,$query);

if($result)
{
  while ($row=pg_fetch_assoc($result))
  {
    echo $row['id'].' - '.$row['nome']."<br>\n";
    •echo $row['pname'].' - '.$row['uname']."<br>\n";
  }
}

?>
```

# PHP com BD

Os arquivos conecta.php, insere.php e mostra.php disponíveis no moodle mostram instruções e um código mais completo sobre como:

- conectar com o BD,
  - inserir dados no BD,
  - selecionar e mostrar dados do BD.
- 
- Para que os exemplos rodem, deve-se baixar e rodar o SQL que cria as tabelas do BD 'loja' conforme as instruções do arquivo 'instrucoes.txt'.
  - Utilizaremos nas próximas aulas 2 Bds: O BD loja para rodar os exemplos das aulas e o BD 'locadora' utilizado nos exercícios avaliados



# Desenvolvimento Web I

## PHP

Tiago Lopes Telecken

[telecken@gmail.com](mailto:telecken@gmail.com)

