Análise e Desenvolvimento de Sistemas

Ferramentas de Desenvolvimento *Web*

Aula 5

Prof. Silvio Bortoletto

CONVERSA INICIAL

Olá! Seja bem-vindo à quinta aula de **Ferramentas e Desenvolvimento Web**. Nela, você estudará o conceito de PHP, sua sintaxe básica e variáveis, bem como suas funções, banco de dados e orientação a objetos.

Pronto para começar?

CONTEXTUALIZANDO

O PHP (acrônimo recursivo para " *Hypertext Preprocessor*", originalmente *Personal Home Page*) é uma linguagem dinâmica e flexível, que suporta uma variedade de técnicas de programação. O PHP possui um conjunto completo de funcionalidades de programação orientada a objetos, incluindo suporte às classes, interfaces, heranças, construtores, clonagem, exceções e muito mais. Vamos aprender algumas delas na aula de hoje?

PESQUISE

O que é PHP?

PHP *(Hypertext Preprocessor* ou pré-processador de hipertexto) é uma linguagem de programação utilizada em escala mundial no desenvolvimento web. Concorrente direta da ASP (desenvolvida pela Microsoft), o PHP é uma linguagem *open source* que possui grande facilidade na mesclagem com HTML e JavaScript, tornando mais fácil e rápida a geração de páginas e conteúdos dinâmicos. Assim como JavaScript, o PHP é uma linguagem *case sensitive*, ou seja, faz diferenciação entre letras maiúsculas e minúsculas na declaração de métodos, funções e variáveis.

Além da geração de páginas e conteúdos dinâmicos, o PHP também é utilizado na ligação com bancos de dados, pela facilidade e comodidade.

Para seus scripts rodarem em ambiente de desenvolvimento local (na sua máquina, com arquivos no seu disco) é necessário a instalação de um servidor.

Alguns softwares, como o XAMPP e o WAMP Server são utilizados no desenvolvimento em PHP, pois reúnem em um mesmo lugar um servidor (Apache), banco de dados (MySQL) e suporte a linguagens de programação (além do PHP, Python e Perl).

Sintaxe Básica e Variáveis

Assim como JavaScript e/ou jQuery, o código PHP poderá ficar embutido dentro da página HTML ou em um arquivo externo (.php). O compilador e o navegador irão interpretar o código devido a presença dos delimitadores **<? ?>**, assim como acontece em HTML, os *browsers* identificam a tag **<html>** e reconhecem que tal bloco de código é HTML.

As funções **echo** e **print** realizam impressões na tela, assim como em jQuery a função **console.log()** imprime algo no console do navegador.

```
<body>
    <?php
    echo "<p>Hello World</p>";
    ?>

    <?php
    print "<p>Hello World</p>";
    ?>
</body>
```

Assim como toda linguagem de programação, o PHP também possui suas variáveis. A declaração de variáveis segue o seguinte padrão:

$nome da variável = “dado”;

Observe:

```
<?php
    $nome = "Aprendendo PHP";
    echo $nome;
?>
```

Uma variável poderá conter:

- Valores inteiros: ***Integer*** ou ***Long***;

```
$teste = 1234;
//inteiro positivo em base decimal

$teste = -234;
//inteiro negativo em base decimal

$teste = 0234;
//inteiro base octal, simbolizado pelo 0 = equivalente a 156

$teste = 0x34;
//inteiro base hexadecimal
//simbolizado pelo 0x = equivalente a 52 decimal
```

- Pontos flutuantes: ***Float*** ou ***Double****;*

```
$teste = 1.234;

$teste = 23e4;
//equivalente a 230.000
```

- ***Strings***;

```
$teste = "Você está indo bem!";

$teste = 'Variáveis são legais!'

//tanto faz a utilização de "" ou ''
```

- ***Arrays***;

```php
$meses = array(1 => "Janeiro",
               2 => "Fevereiro",
               3 => "Março",
               4 => "Abril",
               5 => "Maio",
               6 => "Junho",
               7 => "Julho",
               8 => "Agosto",
               9 => "Setembro",
               10 => "Outubro",
               11 => "Novembro",
               12 => "Dezembro");
```

Quer aprofundar seus estudos em PHP? Então não deixe de assitir à explicação do professor Sílvio Bortoletto no material online.

Funções

Qualquer código PHP válido pode estar contido em funções. Como a checagem de tipos é dinâmica, o tipo de retorno não deve ser declarado, sendo necessário estar atento para que a função retorne o tipo desejado. Observe:

```php
function nome_da_funcao([argumento1, argumento2]){
    comandos;
    [return <valor de retorno>];
}
```

Existem duas maneiras de fazer com que a função tenha parâmetros passados por referência:

- Indicando isso na declaração da função, fazendo com que a passagem de parâmetros seja sempre assim;
- Na própria chamada da função.

Nos dois casos se utiliza o modificador "**&**".

Em PHP, é possível ter valores-padrão para argumentos de funções, que serão assumidos no caso de nada ser passado no lugar do argumento. Quando algum parâmetro é declarado dessa forma, a passagem dele na chamada da função torna-se opcional.

Fique por dentro dos principais aspectos das funções assistindo à explicação do professor Sílvio Bortoletto sobre elas no material online.

Formulários

A linguagem PHP também é utilizada para o tratamento dos dados enviados através de um formulário em uma página HTML.

Por meio da definição da instrução HTML *method*="..." na tag <**form**>, definimos como os dados contidos no formulário serão enviados para o servidor, para que um script em PHP os receba, interprete e execute alguma função. Isso pode acontecer por meio de dois métodos: GET e POST.

No método GET todas as informações transmitidas pelo *form* para o servidor são ativadas ao final da URL ativa. A maioria dos documentos HTML são recuperados a partir da requisição de uma única URL ao servidor.

No método POST todas as informações transmitidas pelo *form* para o servidor são imediatamente enviadas após a URL, ou seja, quando o servidor recebe uma ativação de um formulário que está utilizando o método POST, ele reconhece que precisa continuar “ouvindo” para obter a informação.

Além da instrução ***method***="...", na tag <**form**>, precisamos definir o atributo *action*="...", que irá definir qual script irá receber os dados do formulário.

Observe como isso acontece no método GET:

```
<form name="" action="" method="" enctype="">
```

```
<form action="script.php" method="get">
    Campo 1: <input type="text" name="campo1">
    Campo 2: <input type="text" name="campo2">
</form>
```

HTML

Agora observe como esse mesmo procedimento ocorre no método POST:

```html
<form action="script.php" method="post">
    Campo 1: <input type="text" name="campo1">
    Campo 2: <input type="text" name="campo2">
</form>
```

HTML

Banco de Dados

Com a instalação do XAMPP ou do *Wamp Server*, possuímos um servidor local, suporte ao PHP e um servidor de banco de dados, o MySQL. Utilizando o MySQL, o trabalho com a linguagem SQL através de PHP se torna fácil. O comando responsável por efetuar a conexão com o servidor MySQL de banco de dados é:

```
mysql_connect(string host [:porta], string login, string senha);
```

Sendo que:

string host indica o endereço do servidor MySQL.

[:porta] indica uma porta opcional na qual o servidor está operando.

string login indica o nome de usuário no qual será realizada a conexão.

string senha indica a senha do usuário que está se conectando ao servidor.

Por questão de comodidade, podemos armazenar essa *string* de conexão em uma variável, assim, sempre que precisarmos dela novamente, podemos apenas chamar o nome da variável e, em uma próxima chamada, apenas **$nome** irá realizar o mesmo comando!

```
$nome = mysql_connect(string host [:porta], string login, string senha);
```

Assim, se a conexão for bem-sucedida, ela ficará armazenada dentro da variável, e poderemos solicitá-la no momento que desejarmos.

```php
<?php
    $conexao = mysql_connect("localhost", "root", "")
    or die("Não foi possível conectar");
?>
```

Após a conexão com o servidor de banco, precisamos selecionar um banco de dados no qual iremos realizar as inserções e/ou mudanças nos registros. O comando irá nos retornar 0 ou 1, se tudo correr bem o retorno será 1, indicando o sucesso na operação. Após isso, qualquer *query* executada será válida para a base de dados selecionada.

```
mysql_select_db(string nome_do_banco, string conexao);
```

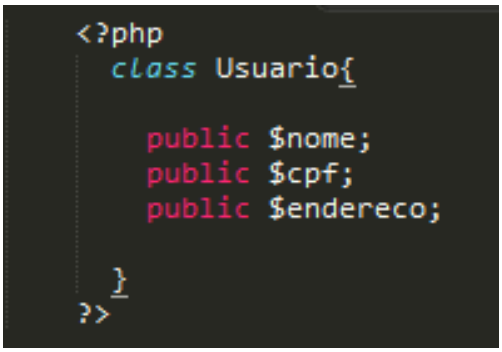
O nome da base de dados (database) que será selecionado deverá sempre ser o primeiro parâmetro fornecido, seguido do identificador da conexão.

O professor Sílvio Bortoletto fala mais sobre os formulários e o banco de dados no material online. Não perca!

Orientação a Objetos

A linguagem PHP não foi criada com o objetivo de ser uma linguagem orientada a objetos, assim como o Java. Mas, a partir da versão 3 passou a suportar a função, já que vários desenvolvedores utilizavam a programação estruturada para simular essa funcionalidade. O primeiro conceito básico da orientação a objetos é a classe, onde são feitas abstrações do software de objetos similares (um template em que os objetos serão criados). Dentro da classe, são definidos os atributos e seus valores.

A visibilidade é outro conceito fundamental em orientação a objetos. A visibilidade de um método ou atributo pode ser definida fixando as palavras-chave ***public***, ***private*** ou ***protected*.**



Agora é com o professor Sílvio Bortoletto! É ele quem fala mais sobre a orientação dos objetos no material online.

NA PRÁTICA

WampServer é um software desenvolvido pela PHP Team. Ele é usado para instalar rapidamente no computador os softwares PHP 5, MySQL e Apache, disponibilizando suporte ao uso de scripts PHP localmente no Windows. Existem vários add-ons para download no site oficial, acesse-os clicando no link a seguir.

http://www.wampserver.com/en/

SÍNTESE

O PHP é uma linguagem de extrema importância no desenvolvimento web, não apenas para websites, mas principalmente para sistemas e aplicações web. É um excelente diferencial no mercado, hoje cada vez mais competitivo.

Referências

W3SCHOOLS. Disponível em: < www.w3schools.com>. Acesso em: 1 abr. 2016.

CODESCHOOLS. Disponível em: < www.codeschool.com >. Acesso em: 1 abr. 2016.

ALVES, William Pereira. **Crie, Anime e Publique Seu Site Utilizando Fireworks CS5, Flash CS5 e Dreamweaver CS5 em Português para Windows**. 1. ed. São Paulo: Érica, 2010.

BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.

PREECE, ROGERS, SHARP. **Designer de Interação: além da interação Homem-Computador**. Ed. Bookman, 2005. 348 p.

NIELSEN, Jakob. **Projetando Websites**. São Paulo. Campus. 2000.

ROBBINS, Jennifer Niederst. **HTML & XHTML: guia de bolso**. Starlin Alta Com. Ltda. 2008.

YNEMINE, Silvana Tauhata. **Dreamweaver CS4**. Florianópolis: Visual Books, 2009.