

Funções e orientação a objetos em PHP

Apresentação

A linguagem PHP oferece ao usuário a possibilidade de utilização de uma infinidade de funções para realizar as mais distintas tarefas.

É possível utilizar os recursos de orientação a objeto no PHP, de modo a obter melhor otimização do código e uma estrutura com muito menos linhas de programação com o uso de classes. Também é possível a utilização de sessões para armazenamento de dados temporários no servidor para uso posterior.

Nesta Unidade de Aprendizagem, você vai estudar mais detalhes sobre as funções no PHP, seus recursos de orientação a objeto e a aplicação desses recursos.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Definir o uso de funções no PHP.
- Reconhecer os recursos de orientação a objeto.
- Aplicar os recursos do PHP.

Infográfico

Uma das mudanças mais significativas entre as versões da linguagem PHP foi a inserção de conceitos de orientação a objetos. Até a versão 4, o PHP tratava os dados como tipos primitivos. A partir da versão 5, passou a manipular também objetos, possibilitando melhor desempenho e funcionalidades adicionais.

Veja, neste Infográfico, os principais conceitos relacionados a orientação a objetos.

ORIENTAÇÃO A OBJETOS

Trata-se de um paradigma de programação no qual as aplicações são desenvolvidas com base em uma coleção de objetos, que são componentes reusáveis.

Veja, a seguir, os **principais conceitos relacionados a orientação a objetos**.

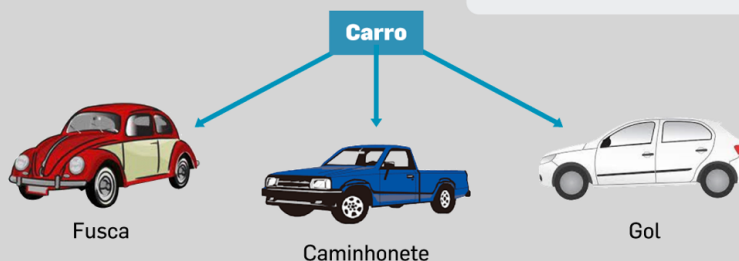


Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Principais conceitos

Carro é uma **classe**, que, por sua vez, é a abstração do objeto. É um modelo a partir do qual os objetos podem ser criados.

Fusca, caminhonete e gol são **objetos**, que, por sua vez, são instâncias da classe carro.



A **classe é uma generalização do objeto**, mostrando quais são suas características (ou atributos) e suas funcionalidades (ou métodos).

Carro
marca
modelo
cor
ano
setMarca(m)
getMarca()
setModelo(x)
getModelo()
setCor(c)
getCor()
setAno(a)
getAno()
getCarro()

Atributos: variáveis definidas para a classe, ou seja, dados que a classe tem.

Métodos: podem alterar os atributos e realizar funções inerentes à classe.

Os métodos *set* atribuem um valor ao atributo, e os métodos *get* recuperam o valor do atributo.

Neste exemplo, definem-se a classe carro com quatro atributos – marca, modelo, cor e ano – e os métodos para modificar esses atributos e recuperar seus valores, além de um último método, chamado *getCarro()*, que retorna todos os atributos do objeto.

Portanto, na programação orientada a objetos, as classes desempenham um papel fundamental. Elas são as estruturas que determinam a descrição dos objetos, que, por sua vez, a utilizam como base. Dessa forma, é nas classes que serão definidos os atributos e métodos que os objetos podem conter.

Conteúdo do Livro

Com a linguagem PHP, o usuário pode utilizar uma infinidade de funções para realizar distintas tarefas.

Acompanhe, no capítulo **Linguagem PHP**, da obra *Desenvolvimento de software II*, base teórica desta Unidade de Aprendizagem, mais detalhes sobre como utilizar os recursos oferecidos, bem como suas aplicações.

Boa leitura.

EVANDRO MANARA MILETTO
SILVIA DE CASTRO BERTAGNOLLI



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO SUL

DESENVOLVIMENTO DE SOFTWARE II

INTRODUÇÃO AO DESENVOLVIMENTO WEB
COM HTML, CSS, JAVASCRIPT E PHP

EIXO INFORMAÇÃO E COMUNICAÇÃO

>> série tekne





D451 Desenvolvimento de software II [recurso eletrônico] : introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP / Organizadores, Evandro Manara Miletto, Sílvia de Castro Bertagnolli. – Dados eletrônicos. – Porto Alegre : Bookman, 2014.

Editado também como livro impresso em 2014.
ISBN 978-85-8260-196-9

1. Informática – Desenvolvimento de software. 2. HTML.
3. CSS. 4. JavaScript. 5. PHP. I. Miletto, Evandro Manara.
II. Bertagnolli, Sílvia de Castro.

CDU 004.41



» IMPORTANTE

Observe o uso, dentro da classe, da variável especial `$this`. Esta variável pode ser utilizada para acessar internamente atributos (`$this->nome`) ou métodos da classe.

apresentados outros dois métodos (GET e set) para manipulação do atributo `nome` da classe (GETs e sets para os demais atributos foram omitidos neste exemplo).

Após a especificação da classe, há a instanciação de um objeto chamado de `$produtoatual` por meio da utilização da palavra-chave `new` em conjunto com o nome da classe e de todos os dados requeridos pelo seu construtor. Por fim, apresentamos a chamada ao método de atribuição do nome do produto (`$produtoatual->setNome("Notebook - edição especial")`) instanciado no objeto atual, passando como argumento uma nova string. O acesso a atributos ou métodos sempre se dá com o uso dos sinais `->` depois da variável que identifica o objeto e antes do elemento/método que se deseja acessar.

Evite a definição dos atributos de uma classe como `public`, a menos que seja realmente necessário. Como boa prática, priorize o acesso aos atributos por meio de métodos disponibilizados pela própria classe, definindo esses métodos como `public`, e os atributos como `private`.



» Agora é a sua vez!

1. A partir da classe `Produto`, apresentada no exemplo anterior, crie um script PHP (`produto.php`) para conter especificamente as definições dessa classe e o armazene dentro de um diretório de classes (`./class/`). O diretório deve conter todos os scripts de definições de classes de seu sistema.
2. No arquivo recém-criado, insira as definições da classe apresentadas em nosso exemplo e amplie sua especificação incluindo os métodos GETs e sets para acesso aos demais atributos. Usaremos esse arquivo criado no decorrer dos exemplos de nosso livro.

» Herança

A implementação de herança em PHP se assemelha muito ao padrão adotado por outras linguagens, como em Java, devendo-se utilizar a palavra-chave `extends` na declaração de uma classe para explicitar que ela herdará atributos e métodos de outra classe informada. O exemplo a seguir, ao ser apresentado em conjunto com o exemplo anterior, formaliza uma classe chamada de `ProdutoVip`, que,

além de ter todos os atributos e métodos da classe `Produto`, também apresenta um novo atributo chamado de `desconto`, bem como os métodos para alterá-lo.



>> EXEMPLO

```
<?php
class ProdutoVip extends Produto {
    private $desconto; // novo atributo, presente apenas em ProdutoVip
    // os demais atributos de Produto são herdados por ProdutoVip
    function __construct ($varnome, $varpreco, $vardescricao, $varquantidade,
        $varcodigo, $vardesconto) { // método construtor da classe
        parent::__construct ($varnome, $varpreco, $vardescricao, $varquantidade,
            $varcodigo); // chamada ao construtor da superclasse Produto
        $this->desconto = $vardesconto; // inicialização do novo atributo da subclasse
        ProdutoVip, não inicializada pelo construtor da superclasse Produto
    }
    public function getDesconto() { // método para obter valor do atributo
        return $this->desconto;
    }
    public function setDesconto($vardesconto) { // método para definir valor do atributo
        $this->nome = $vardesconto;
    }
    // os demais métodos de Produto são herdados por ProdutoVip
}

$produtovipatual = new ProdutoVip("Notebook", 2999, "Intel Core i5 6GB 1TB LED 14
    Touchscreen Windows 8", 200, 28, 17.5); // instanciando um objeto da subclasse,
    chamando seu próprio construtor
echo 'O valor de desconto do produto vip '.$produtovipatual->getNome().' é
    '.$produtovipatual->getDesconto().'%. ' . '%.';
?>
```

Observe que, além do novo atributo `desconto`, a subclasse `ProdutoVip` apresenta o seu próprio construtor, o qual recebe os mesmos parâmetros da superclasse `Produto` e também o valor para o novo atributo `desconto`. Sempre que instanciarmos um objeto de uma classe herdeira, uma chamada será feita ao seu construtor, e não ao construtor da superclasse.

No entanto, note que, dentro do construtor de `ProdutoVip`, é utilizada a palavra-chave `parent` seguida de `::`, o que determina que o atributo ou método que vier a seguir está se referindo à superclasse da qual se está herdando. No caso do construtor de `ProdutoVip`, com o trecho `parent::__construct` estamos determinando que será chamado o construtor da superclasse `Produto` e este inicializa as variáveis presentes na superclasse. Já o novo atributo apresentado pela subclasse `ProdutoVip` passa a ser inicializado logo em seguida.



Por fim, a última linha de instruções do código solicita a impressão de uma frase em que serão apresentados atributos do objeto. Observe que, mesmo se tratando de um objeto da classe `ProdutoVip`, é possível acessar o método `getNome` que se encontra definido na classe `Produto` (esse método foi herdado por `ProdutoVip`). O método `getDesconto`, definido na classe `ProdutoVip`, também é solicitado nessa linha, cujo resultado a ser impresso no navegador seria: "O valor de desconto do produto vip Notebook é 17,5%".

»» PARA SABER MAIS

Informações mais aprofundadas sobre conceitos de orientação a objetos fogem do escopo desta obra. Para a obtenção de um maior detalhamento, sugere-se a consulta do livro *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*, 3. ed. (Veja referência ao final do capítulo.)

»» Reuso de código: Include e Require

No processo de desenvolvimento de um sistema, muitas vezes pode ser interessante realizar o reaproveitamento de trechos de códigos já desenvolvidos. O reuso de código proporciona benefícios ao reduzir o tempo necessário para a realização de procedimentos já formalizados e testados, bem como ao favorecer a manutenção desses procedimentos em caso de necessidade de futuras alterações.

Assim, o `Include` e o `Require` são recursos muito semelhantes no PHP. Ambos são utilizados para a inclusão de um determinado arquivo no script em que está sendo chamado. Para usar o `include`, devemos especificar o arquivo que será incluído no script em que a função é chamada. O arquivo será incluído tomando como base o caminho relativo apresentado (conceito abordado no Capítulo 4) ou, nos casos de não haver especificação, o caminho definido por `include_path` nas configurações do PHP no servidor.

Veja a seguir um exemplo para a inclusão de um arquivo que contém as definições da classe `Produto`. Ele permite visualizar como podemos fazer uso no script atual dos recursos definidos pelo arquivo incluído, sem a necessidade de reescrita do trecho de código com as definições da classe.

Página propositalmente deixada em branco



>> capítulo 8

Integração de PHP e MySQL

Neste capítulo, apresentaremos como se dá a utilização do PHP em conjunto com o Sistema Gerenciador de Banco de Dados (SGBD), mais especificamente com o MySQL. Serão abordados os procedimentos necessários para permitir o gerenciamento e a consulta das informações armazenadas no exemplo de sistema de compras online trabalhado em todos os capítulos.

Objetivos de aprendizagem

- >> Realizar os procedimentos necessários para a estruturação das bases de dados utilizadas em sistemas Web.
- >> Gerenciar as informações armazenadas nas bases de dados.
- >> Realizar e tratar os resultados de consultas realizadas sobre bases de dados MySQL.



>> IMPORTANTE

O MySQL também possibilita a aquisição de licenças específicas para uso comercial do SGBD.

>> Introdução

Conforme visto no capítulo anterior, o PHP apresenta compatibilidade com um conjunto bastante significativo de Sistemas Gerenciadores de Bancos de Dados (SGBD). Dentre os sistemas relacionais suportados, destaca-se o **MySQL**, distribuído sob licença GPL (*GNU General Public License*), e de ampla utilização e compatibilidade com distintas linguagem de programação e plataformas de operação.



>> CURIOSIDADE

O MySQL foi disponibilizado, em sua primeira versão, em 1996, apesar de já ter ocorrido um lançamento interno do sistema em maio de 1995. É considerado o segundo SGBD mais utilizado no mundo, de acordo com dados de outubro de 2013, da pesquisa da DB-Engines.

www.

>> NO SITE

Para obter detalhes sobre os procedimentos específicos para instalação do MySQL, nas mais distintas plataformas, acesse o ambiente virtual de aprendizagem Tekne: www.bookman.com.br/tekne.

Para atuar em conjunto com o SGBD MySQL, a linguagem PHP apresenta um amplo conjunto de funções. Apresentaremos, no decorrer deste capítulo, as fundamentais para habilitá-lo a implementar um sistema Web que possibilite o gerenciamento e a consulta de informações armazenadas em um banco de dados.

A fim de que seja possível realizar testes e implementar os exemplos que serão apresentados, é fundamental que você tenha o MySQL instalado e configurado em um servidor ou em seu computador pessoal.

Inicialmente, abordaremos os aspectos fundamentais para que se estabeleça a conexão com um SGBD MySQL. Na sequência, apresentaremos os procedimentos para o gerenciamento e seleção de bases de dados. A interação com uma base de dados é apresentada no tópico seguinte, em que será explorada a realização de consultas SQL. Por fim, chegará a vez das funções e dos procedimentos para o tratamento das consultas realizadas e das possibilidades de recursos para utilização do MySQL com orientação a objetos.

www.

>> NO SITE

Para mais informações sobre a pesquisa da DB-Engines, acesse o ambiente virtual de aprendizagem Tekne.



>> EXEMPLO

```
<?php
$con = mysql_connect('localhost', 'administrador', 'minhasenha');
if (!$con)
    exit('Não foi possível conectar: ' . mysql_error());
// insira aqui o restante do código para utilização da conexão...
mysql_close($con); // encerramento da conexão
?>
```



>> IMPORTANTE

O PHP disponibiliza também a função `mysql_pconnect()` para o estabelecimento de conexão com o MySQL, diferenciando-se da `mysql_connect()` por estabelecer uma conexão persistente com o banco, não sendo encerrada ao final da execução do script.

Observe, no exemplo, que é feito um teste após a tentativa de estabelecimento de conexão, para verificar se o procedimento foi realizado com sucesso. Caso tenha ocorrido algum problema, `$con` receberá o valor `false` e será utilizada a função `mysql_error()` para apresentar informações complementares sobre a falha no estabelecimento, encerrando a execução do script com a utilização da construção de linguagem `exit()`.

Após a inclusão do código adicional que fará uso da conexão recém-estabelecida, deve-se inserir uma chamada à função `mysql_close()`, a qual recebe como parâmetro o identificador da conexão para proceder com seu encerramento. Caso não seja passado um identificador como parâmetro, a última conexão estabelecida será encerrada. Se não for feito uso dessa função, a conexão será encerrada tão logo a execução do script terminar.



>> CURIOSIDADE

O procedimento de fazer uso da última conexão estabelecida em caso de supressão do identificador é geralmente adotado nas demais funções relacionadas ao MySQL que não apresentam esse parâmetro como obrigatório.

>> Seleção de bases de dados

Uma vez estabelecida uma conexão com o MySQL e partindo-se de uma base de dados já existente, deve-se passar à seleção dessa base de dados, para que seja possível o gerenciamento das informações nela armazenadas. É a função `mysql_select_db()` que possibilita essa seleção. Para realizar a seleção de uma base de dados, é necessário especificar o nome da base que se deseja selecionar e, opcionalmente, o identificador de conexão adotado. Em caso de sucesso na seleção, aparece o retorno do valor `true`, e, para casos de insucesso, o retorno `false`.

A sintaxe da função `mysql_select_db()` é:

```
bool mysql_select_db (string $database_name [,  
resource $ link_identifier ])
```

A criação de uma base de dados pode ser feita com a função específica do PHP `mysql_create_db()`. No entanto, o uso dessa função tem se tornado obsoleto, sendo preferencial o uso da consulta, em SQL, `CREATE DATABASE`, em conjunto com a função `mysql_query()`. Veremos os aspectos relacionados ao uso desta função de execução de comandos SQL no decorrer deste capítulo.

Veja o exemplo que apresenta um trecho de código do procedimento de seleção para interação de uma base de dados já existente denominada “sitedecompras”.

>> EXEMPLO

```
<?php  
include 'conectamysql.php'; // arquivo que estabelece a conexão  
$db_selecionado = mysql_select_db('sitedecompras', $con);  
if (!$db_selecionado)  
    exit('Não foi possível selecionar a base de dados: '. mysql_error());  
// insira aqui seu código para uso da base de dados...  
mysql_close($con); // encerramento da conexão  
?>
```

No código apresentado no exemplo anterior, foi estabelecida uma conexão com o banco de dados do servidor e, em seguida, realizada a seleção da base de dados “sitedecompras”. Caso tenha ocorrido algum problema na seleção da base, é apresentada uma mensagem ao usuário especificando o problema.

Observe que o procedimento de estabelecimento de conexão apresentado no exemplo da seção “Conexão com o banco de dados” foi disponibilizado dentro do arquivo `conectamysql.php`, o qual foi incluído no script na primeira linha. Esse procedimento é recomendado para evitar que informações referentes ao usuário e à senha sejam disponibilizadas em todos os scripts que interajam com o banco de dados, gerando problemas de segurança e dificuldade na manutenção do sistema em caso de alterações nas configurações do servidor MySQL.

» Realização de consultas SQL

A partir do estabelecimento de conexão com o banco de dados e da seleção da base de dados adotada, é possível passar à interação com esta última. Esse processo se dará principalmente por meio da realização de consultas SQL, possibilitada pelo uso da função `mysql_query()`.

A sintaxe para `mysql_query()` é:

```
resource mysql_query (string $query [, resource  
$link_identifier ])
```

Ao fazer uso dessa função, deve-se passar como primeiro parâmetro a consulta SQL que se deseja realizar (`$query`) e, opcionalmente, o identificador de uma conexão já estabelecida com o banco de dados. O valor retornado pela execução com sucesso da função dependerá do tipo de comando SQL utilizado: `SELECT`, `SHOW`, `DESCRIBE` ou `EXPLAIN` retornam um `resource` que poderá ser tratado no decorrer do script; já as demais consultas SQL (como `UPDATE` ou `DELETE`) retornam `true`. Para qualquer SQL realizado, o valor `false` será retornado sempre que ocorrerem falhas.

Observe a seguir o exemplo sobre o uso dessa função no processo de criação da base de dados “`sitedecompras`”, caso ela ainda não houvesse sido criada.

Dica do Professor

A linguagem PHP oferece ao desenvolvedor a possibilidade de utilizar uma infinidade de funções para realizar as diversas tarefas e também a possibilidade de criação de funções. Objetos também são usados em PHP desde sua versão 4, na qual foram incluídos os conceitos de orientação a objetos.

Veja, nesta Dica do Professor, como a linguagem PHP se comporta com o uso de funções e orientação a objetos.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) O PHP oferece o uso de uma infinidade de funções ao desenvolvedor e possibilita a criação de funções.

O que é uma função no PHP?

- A) Um comando pré-programado que não pode ser alterado nem contém argumentos.
- B) O mesmo que uma classe.
- C) Um bloco de instruções que realizam determinado procedimento e que podem ser chamadas sempre que for necessário.
- D) Um *array* de dados diversos que pode ser acessado com ou sem um banco de dados.
- E) Um conjunto de variáveis para ordenar informações vindas do banco de dados.

- 2) O PHP permite que o desenvolvedor opte pelo desenvolvimento de aplicações *web* utilizando programação orientada a objetos.

Qual das alternativas cita uma das vantagens da programação orientada a objetos?

- A) Fazer a conexão com o banco de dados de forma bidirecional e assíncrona de uma ou mais expressões simultâneas.
- B) Fazer a utilização de operadores para retornar os valores de uma ou mais expressões simultâneas.
- C) Fazer o controle dos atributos da página, como formulários e textos.
- D) Reduzir o número de linhas de código, evitando redundâncias e repetições.
- E) Permitir os recursos de interatividade da página com o usuário/cliente.

- 3) A orientação a objetos é um paradigma de programação baseado na abstração do mundo real e envolve outros conceitos, como os objetos e as classes.

O que é uma classe?

- A) Ação ou serviço específico que um ou mais objetos podem utilizar.
 - B) Uma variável superglobal utilizada em sessões de PHP.
 - C) Um identificador que só utiliza um valor, não podendo ser alterado durante o *script*.
 - D) Um elemento que especifica um campo de entrada de dados de um formulário.
 - E) Uma estrutura que define os tipos de dados (atributos e métodos) de seus objetos.
- 4) **Desenvolvedores PHP usam sessão principalmente em carrinhos de compras de *sites* de vendas.**

Afinal, o que é uma sessão?

- A) Uma variável de informações constantes.
 - B) Recurso utilizado para reúso de código e otimização.
 - C) Recurso utilizado para manter informações salvas entre as páginas de um *site*.
 - D) Um conjunto de chaves com a relação de atributos e métodos de uma classe.
 - E) A chave de uso para orientação a objeto.
- 5) **Sessões em PHP são variáveis que permanecem setadas até o usuário fechar o navegador *web* ou a sessão ser encerrada.**

Qual comando se utiliza para encerrar uma sessão em PHP?

- A) "mysqli_num_rows".
- B) "require_once".
- C) "session_destroy()".
- D) "SELECT * FROM banco".
- E) "session_start()".

Na prática

Uma sessão é um recurso por meio do qual se pode manter dados durante a navegação por várias páginas de uma aplicação *web*. A linguagem PHP dispõe de alguns parâmetros e funções específicos para controle das sessões.

Acompanhe, Na Prática, a história de João, desenvolvedor PHP, que utilizou sessões em sua aplicação *web* de carrinho de compras.

CRIANDO UMA APLICAÇÃO DE CARRINHO DE COMPRAS EM PHP USANDO SESSÃO

João é desenvolvedor iniciante em PHP e está envolvido em um novo **projeto de desenvolvimento de uma loja virtual**.

Nesse sistema, o usuário escolherá os produtos que deseja comprar e os colocará dentro do carrinho de compras.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Ele já desenvolveu parte da aplicação, mas, neste momento, **está diante de um problema, que encontrou durante os testes da aplicação: os produtos do carrinho estão se perdendo no meio do caminho**.

Acompanhe, na prática, como João resolveu esse problema.



Pesquisa

Então, João fez uma busca rápida na internet e viu que, para manter as informações necessárias para a navegação eficiente de uma aplicação, **é necessário estabelecer sessões**.

E foi isso que ele fez em seu projeto, que está sendo desenvolvido em PHP.

Trecho script php desenvolvido

Veja, a seguir, o trecho de um *script* PHP desenvolvido por João:

```
<?php
session_start();
if (isset($_POST["codproduto"]) and isset($_POST["quantidadeproduto"]))
{
    $codproduto = $_POST["codproduto"];
    $quantidadeproduto = $_POST["quantidadeproduto"];
    $_SESSION["produtopedido"][$codproduto] = intval($quantidadeproduto);
}
?>
```

Nesse *script*, João usou o comando `session_start()` para inicializar a sessão e usou o código do produto como um índice de um *array* de produtos do pedido. A quantidade de itens informada foi armazenada para cada produto.

Já a função `intval()` foi aplicada para forçar a conversão do valor recebido em *quantidadeproduto* para o tipo inteiro.

Conclusão

Portanto, João conseguiu solucionar o problema com o uso de **sessões em PHP**, e o carrinho de compras passou a armazenar corretamente as quantidades do produto informadas pelo usuário. Isso só foi possível porque **a linguagem PHP trabalha com esse tipo de recurso, guardando os dados para que possam ser usados posteriormente nas aplicações web**.

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Sessões em PHP

Veja, no *site* a seguir, como iniciar uma sessão em PHP, recuperar e alterar seus valores e também como encerrá-la.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Funções em PHP

Acesse o *site* a seguir e aprenda mais sobre funções em PHP. O *site* ainda oferece a possibilidade de colocar em prática o que você aprendeu.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Desenvolvimento de software II – Introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP

No Capítulo 7, Linguagem PHP, aprenda um pouco mais sobre a linguagem PHP, sua sintaxe, suas estruturas de controle, uso de funções, orientação a objetos, entre outros recursos.

Conteúdo interativo disponível na plataforma de ensino!