

Tribunal de Justiça Militar do Estado de São Paulo

TJM – SP

Técnico em Comunicação e Processamento de Dados
Judiciário (Desenvolvedor)

DZ083-N9

Todos os direitos autorais desta obra são protegidos pela Lei nº 9.610, de 19/12/1998.
Proibida a reprodução, total ou parcialmente, sem autorização prévia expressa por escrito da editora e do autor. Se você conhece algum caso de "pirataria" de nossos materiais, denuncie pelo sac@novaconcursos.com.br.

OBRA

Tribunal de Justiça Militar do Estado de São Paulo - TJM-SP

Técnico em Comunicação e Processamento de Dados Judiciário (Desenvolvedor)

Edital Nº 001/2019

AUTORES

Língua Portuguesa - Profª Zenaide Auxiliadora Pachegas Branco
Raciocínio Lógico - Profº Bruno Chierregatti e João de Sá Brasil
Conhecimentos Específicos - Profº Carlos Quiqueto

PRODUÇÃO EDITORIAL/REVISÃO

Leandro Filho

DIAGRAMAÇÃO

Thais Regis
Renato Vilela

CAPA

Joel Ferreira dos Santos



www.novaconcursos.com.br

sac@novaconcursos.com.br

APRESENTAÇÃO

PARABÉNS! ESTE É O PASSAPORTE PARA SUA APROVAÇÃO.

A Nova Concursos tem um único propósito: mudar a vida das pessoas.

Vamos ajudar você a alcançar o tão desejado cargo público.

Nossos livros são elaborados por professores que atuam na área de Concursos Públicos. Assim a matéria é organizada de forma que otimize o tempo do candidato. Afinal corremos contra o tempo, por isso a preparação é muito importante.

Aproveitando, convidamos você para conhecer nossa linha de produtos "Cursos online", conteúdos preparatórios e por edital, ministrados pelos melhores professores do mercado.

Estar à frente é nosso objetivo, sempre.

Contamos com índice de aprovação de 87%*.

O que nos motiva é a busca da excelência. Aumentar este índice é nossa meta.

Acesse **www.novaconcursos.com.br** e conheça todos os nossos produtos.

Oferecemos uma solução completa com foco na sua aprovação, como: apostilas, livros, cursos online, questões comentadas e treinamentos com simulados online.

Desejamos-lhe muito sucesso nesta nova etapa da sua vida!

Obrigado e bons estudos!

*Índice de aprovação baseado em ferramentas internas de medição.

CURSO ONLINE



PASSO 1

Acesse:
www.novaconcursos.com.br/passaporte



PASSO 2

Digite o código do produto no campo indicado no site.

O código encontra-se no verso da capa da apostila.

*Utilize sempre os 8 primeiros dígitos.

Ex: JN001-19



PASSO 3

Pronto!
Você já pode acessar os conteúdos online.

SUMÁRIO

LÍNGUA PORTUGUESA

Leitura e interpretação de diversos tipos de textos (literários e não literários).....	01
Sinônimos e antônimos.....	10
Sentido próprio e figurado das palavras.....	10
Processos de constituição dos enunciados: coordenação, subordinação; concordâncias verbal e nominal.....	62
Pontuação.....	13
Classes de palavras: substantivo, adjetivo, numeral, artigo, pronome, verbo, advérbio, preposição e conjunção: emprego e sentido que imprimem às relações que estabelecem.....	17
Concordância verbal e nominal.....	55
Regência verbal e nominal.....	62
Colocação pronominal.....	68
Crase.....	68

RACIOCÍNIO LÓGICO

Estrutura lógica de relações arbitrárias entre pessoas, lugares, objetos ou eventos fictícios. Dedução de novas informações das relações fornecidas e avaliação das condições usadas para estabelecer a estrutura daquelas relações.....	01
Compreensão e elaboração da lógica das situações por meio de: raciocínio verbal; raciocínio matemático (que envolva, dentre outros, conjuntos numéricos racionais e reais – operações, propriedades, problemas envolvendo as quatro operações nas formas fracionária e decimal, conjuntos numéricos complexos, números e grandezas proporcionais, razão e proporção, divisão proporcional, regra de três simples e composta, porcentagem); raciocínio sequencial; orientação espacial e temporal; formação de conceitos; discriminação de elementos. Compreensão do processo lógico que, a partir de um conjunto de hipóteses, conduz, de forma válida, a conclusões determinadas.....	33

SUMÁRIO

CONHECIMENTOS ESPECÍFICOS

a) Desenvolvimento para web: PHP, Javascript, jQuery, ASP; C#, Java EE: JSP, Servlets, JPA, EJB, JSF, JDBC, Hibernate; XML, HTML5 e CSS3; web services; integração de aplicações com bancos de dados; linguagens de programação: Java SE; construção de programas.....	01
b) Estrutura da linguagem: JVM e bytecode; anotações; coleções; serialização; reflexão; operadores; estruturas de decisão e de repetição; tipos; enumeradores; arrays; Shell. Servidores: noções de uso e administração de contêineres (Apache, JBoss AS/WildFly).....	01
c) Ambientes de desenvolvimento: Eclipse, NetBeans e Visual Studio.....	01
d) Bancos de dados: Conceitos e fundamentos de sistemas gerenciadores de banco de dados (SGBDs): SQLServer, Postgres e MySQL; Linguagem SQL; Consultas e subconsultas; Comandos de manipulação de dados (DML), controle (DCL) e descrição de dados (DDL). Transação de dados (DTL); expressões regulares; gatilho (trigger); visão (view); interfaces de utilização: principais propriedades e características das bibliotecas mais difundidas; PG/SQL: estrutura da linguagem; stored procedures; tratamento de erros; cursores; SQL dinâmico; Package; Function; Array; projeto e modelagem de banco de dados relacional: modelo entidade-relacionamento; normalização; conceitos da modelagem dimensional.....	122
e) Modelagem de sistema: UML: Conceitos gerais, Diagramas, Casos de Uso, Sequência, Classes, Estados, Atividades, deployment; testes de software: tipos de testes, planos de testes, JUnit; padrões de projeto; desenvolvimento baseado em componentes; desenvolvimento baseado em serviços; princípios de interface com o usuário; segurança no desenvolvimento: práticas de programação segura e revisão de código; controles e testes de segurança para aplicações web; controles e testes de segurança para Web Services.....	159
f) Fundamento de computação: algoritmos iterativos, recursivos; teste de mesa; lógica de programação, tabela verdade; operações lógicas: negação, conjunção, disjunção, operação condicional, operação bicondicional, contradição. Estruturas de dados: vetores e matrizes, listas, pilhas, filas, árvores binárias;.....	183
g) Estruturas de repetição: repetição pré-testada, repetição pós- testada, repetição com variável de controle, iteração de coleção.....	183
h) Estruturas de decisão: condições, operadores relacionais, operadores lógicos, seleção; Compilador, interpretador.....	183
i) Programação Orientada a Objetos (POO): Fundamentos: abstração, classe, objeto, atributo e método, interface, associação e mensagem, herança, polimorfismo, encapsulamento, coesão, pacotes; Algoritmos fundamentais: busca, inserção, atualização e remoção em diversas estruturas (listas, árvores, árvores balanceadas, heaps); Algoritmos de ordenação; Tabelas de dispersão (hashing).....	183
j) Noções de Engenharia de Software: Conceitos, Ciclo de vida, Ciclo de desenvolvimento, Métodos e modelos de desenvolvimento, Modelagem, Processos de software, Requisitos, qualidade do software, métricas e prototipagem, métricas de Processo e Projeto, Arquitetura de aplicações para o ambiente Web.....	223

ÍNDICE

CONHECIMENTOS ESPECÍFICOS – TÉCNICO EM COMUNICAÇÃO E PROCESSAMENTO DE DADOS JUDICIÁRIO (DESENVOLVEDOR)

a) Desenvolvimento para web: PHP, Javascript, jQuery, ASP; C#, Java EE: JSP, Servlets, JPA, EJB, JSF, JDBC, Hibernate; XML, HTML5 e CSS3; web services; integração de aplicações com bancos de dados; linguagens de programação: Java SE; construção de programas.....	01
b) Estrutura da linguagem: JVM e bytecode; anotações; coleções; serialização; reflexão; operadores; estruturas de decisão e de repetição; tipos; enumeradores; arrays; Shell. Servidores: noções de uso e administração de contêineres (Apache, JBoss AS/WildFly).....	01
c) Ambientes de desenvolvimento: Eclipse, NetBeans e Visual Studio.....	01
d) Bancos de dados: Conceitos e fundamentos de sistemas gerenciadores de banco de dados (SGBDs): SQLServer, Postgres e MySQL; Linguagem SQL; Consultas e subconsultas; Comandos de manipulação de dados (DML), controle (DCL) e descrição de dados (DDL). Transação de dados (DTL); expressões regulares; gatilho (trigger); visão (view); interfaces de utilização: principais propriedades e características das bibliotecas mais difundidas; PG/SQL: estrutura da linguagem; stored procedures; tratamento de erros; cursores; SQL dinâmico; Package; Function; Array; projeto e modelagem de banco de dados relacional: modelo entidade-relacionamento; normalização; conceitos da modelagem dimensional.....	122
e) Modelagem de sistema: UML: Conceitos gerais, Diagramas, Casos de Uso, Sequência, Classes, Estados, Atividades, deployment; testes de software: tipos de testes, planos de testes, JUnit; padrões de projeto; desenvolvimento baseado em componentes; desenvolvimento baseado em serviços; princípios de interface com o usuário; segurança no desenvolvimento: práticas de programação segura e revisão de código; controles e testes de segurança para aplicações web; controles e testes de segurança para Web Services.....	159
f) Fundamento de computação: algoritmos interativos, recursivos; teste de mesa; lógica de programação, tabela verdade; operações lógicas: negação, conjunção, disjunção, operação condicional, operação bicondicional, contradição. Estruturas de dados: vetores e matrizes, listas, pilhas, filas, árvores binárias;.....	183
g) Estruturas de repetição: repetição pré-testada, repetição pós- testada, repetição com variável de controle, iteração de coleção.....	183
h) Estruturas de decisão: condições, operadores relacionais, operadores lógicos, seleção; Compilador, interpretador.....	183
i) Programação Orientada a Objetos (POO): Fundamentos: abstração, classe, objeto, atributo e método, interface, associação e mensagem, herança, polimorfismo, encapsulamento, coesão, pacotes; Algoritmos fundamentais: busca, inserção, atualização e remoção em diversas estruturas (listas, árvores, árvores balanceadas, heaps); Algoritmos de ordenação; Tabelas de dispersão (hashing).....	183
j) Noções de Engenharia de Software: Conceitos, Ciclo de vida, Ciclo de desenvolvimento, Métodos e modelos de desenvolvimento, Modelagem, Processos de software, Requisitos, qualidade do software, métricas e prototipagem, métricas de Processo e Projeto, Arquitetura de aplicações para o ambiente Web.....	223

A) DESENVOLVIMENTO PARA WEB: PHP, JAVASCRIPT, JQUERY, ASP; C#, JAVA EE: JSP, SERVLETS, JPA, EJB, JSF, JDBC, HIBERNATE; XML, HTML5 E CSS3; WEB SERVICES; INTEGRAÇÃO DE APLICAÇÕES COM BANCOS DE DADOS; LINGUAGENS DE PROGRAMAÇÃO: JAVA SE; CONSTRUÇÃO DE PROGRAMAS;
B) ESTRUTURA DA LINGUAGEM: JVM E BYTECODE; ANOTAÇÕES; COLEÇÕES; SERIALIZAÇÃO; REFLEXÃO; OPERADORES; ESTRUTURAS DE DECISÃO E DE REPETIÇÃO; TIPOS; ENUMERADORES; ARRAYS; SHELL. SERVIDORES: NOÇÕES DE USO E ADMINISTRAÇÃO DE CONTÊINERES (APACHE, JBOSS AS/WILDFLY);
C) AMBIENTES DE DESENVOLVIMENTO: ECLIPSE, NETBEANS E VISUAL STUDIO;

PHP

Em curtas palavras, PHP é uma linguagem de programação utilizada por programadores e desenvolvedores para construir sites dinâmicos, extensões de integração de aplicações e agilizar no desenvolvimento de um sistema.

Essa linguagem é mundialmente conhecida e uma das mais utilizadas pela facilidade em aprendê-la, manuseá-la, além de ser compatível com quase todos os sistemas operacionais que existem, tornando o custo menor.

Nos dias atuais, o PHP, é uma das linguagens de programação mais usadas no mundo. O termo PHP, que antes significava Personal Home Page e agora é conhecido como PHP Hypertext Preprocessor, foi criado com apenas um aglomerado de códigos CGI, um elemento que torna a ligação física ou lógica entre dois sistemas ou servidores, descritos em uma linguagem C.

A ideia inicial era acompanhar o tráfego do site pessoal do criador. Os anos passaram e o criador desenvolvia scripts, o que aumentava os recursos que o site dele tinha.

O sucesso dessa linguagem foi tão grande que o criador, Rasmus Lerdof, transformou o aglomerado de códigos CGI em uma linguagem de programação.. Com isso, a grande maioria dos sites e aplicações passaram a utilizar o PHP como linguagem principal.

Back-end e Front-end – dois setores distintos, porém importantes

Existem duas etapas de trabalho quando se fala em criação de um site, projeto online, jogos, softwares ou aplicativos, são o front-end e back-end. Sabe quando você acessa um site ou um aplicativo na internet, o que você está olhando é a interface da página, na qual foi feita com os trabalhos de front-end.

A forma de criação que é utilizado pelo PHP, é a de back-end, ou seja, quando você quer acessar algum site na internet, todos os dados processados para você conseguir entrar na página desejada, são processadas pelo back-end para enviar as informações e dados até o front-end, que programou toda a interface da página para o usuário.

E o PHP é conhecido por utilizar o back-end porque todo o uso desse setor é voltado para os servidores. Nós utilizamos neste mesmo site a linguagem PHP, além de outros programas. Além de sites, alguns programadores também realizam back-end para aplicativos de celulares, mas isso depende da habilidade do programador e da necessidade do projeto, pois na maioria dos casos, essa linguagem não é adequada. Ou seja, front-end e back-end são setores que trabalham de maneiras diferentes.

O que dá para fazer utilizando o PHP?

Quando você quer criar algum site ou algum sistema web, como um e-commerce ou um CMS, irá se deparar com scripts. Eles são roteiros que trazem informações que serão analisadas, processadas e transformadas em ações, por um determinado tipo de linguagem. Neste caso, a linguagem é o PHP.

Com ele, você consegue criar aplicações para efetuarem alguma tarefa que o usuário determinar. Essas aplicações são compiladas dentro de um servidor, chamado de server-side, ou, script do lado do servidor, um termo tradicional e muito utilizado pelos programadores quando se fala em PHP.

Para você conseguir fazer esses scripts funcionarem, vai precisar dos seguintes itens: um navegador, o interpretador do PHP e um servidor web. Você pode aprender a instalar esses itens no guia de instalação e configuração do php.

Ainda falando em scripts, com o PHP você pode criar um script e utilizá-lo sem a necessidade de ter um navegador. Você precisa somente do interpretador. Essa função é muito útil para programadores que desejam executar o script para criação de APIs. Essa função é chamada de execução de script via linha de comando.



#FicaDica

Você também pode criar aplicações para desktop. Não é muito comum, mas se você é um programador que gosta de desafios, conseguirá utilizar os serviços mais avançados. Para fazer isso, você precisa utilizar o PHP-GTK, uma extensão do PHP que vai te ajudar a criar essas aplicações.

Vantagens em utilizar PHP

Como foi citado acima, o PHP roda na grande maioria dos sistemas operacionais, isso faz com que a manutenção de um site que foi programado em PHP seja muito mais fácil e barata. Além disso, os gastos com servidores e implementações são mais baixos que outras linguagens.

O PHP funciona bem em quase todos os bancos de dados, como MSSQL, Oracle, DB2, MySQL. Ou seja, é um custo a menos, pois não tem a necessidade de migrar toda a base de dados do seu site para poder migrar para o PHP.



#FicaDica

O interpretador PHP também pode ser utilizado pela grande maioria dos servidores web nos mais diversos sistemas operacionais, como Linux, Windows, Mac e etc..

Além disso, o PHP possui recursos para poder processar textos e outras ferramentas, contando com uma vasta lista de extensões, como XML, PDO, GEOF. Você pode conferir a lista inteira de extensões no site oficial do PHP.

Sua primeira página PHP ¶

Crie um novo arquivo chamado ola.php e coloque-o no diretório root do seu servidor web (DOCUMENT_ROOT) com o seguinte conteúdo:

Exemplo #1 Nosso primeiro script PHP: ola.php

```
<html>
<head>
  <title>Teste PHP</title>
</head>
<body>
<?php echo "<p>Olá Mundo</p>"; ?>
</body>
</html>
```

Use o seu navegador para acessar o arquivo com a URL de seu servidor web, terminando com a referência ao arquivo /ola.php. Quando o desenvolvimento for local esta URL será algo como http://localhost/ola.php ou http://127.0.0.1/ola.php mas isso depende da configuração do seu servidor web. Se tudo foi configurado corretamente, este arquivo será interpretado pelo PHP e a seguinte mensagem será enviada ao seu navegador:

```
<html>
<head>
  <title>Teste PHP</title>
</head>
<body>
<p>Olá Mundo</p>
</body>
</html>
```

Este programa é realmente simples e você não precisa do PHP para criar uma página assim (fixa). Tudo o que ela faz é mostrar: Hello World utilizando a instrução echo.

Note que o arquivo não precisa ser executável ou especial. O servidor web descobre que este arquivo precisa ser interpretado pelo PHP por causa da extensão ".php", que o servidor é configurado para repassar ao PHP. Pense nisso como um arquivo HTML normal que por acaso possui um conjunto de tags especiais disponíveis para você fazer muitas coisas interessantes.

Se você tentar rodar este exemplo e ele não mostrar nenhuma mensagem de saída, ou aparecer uma caixa de diálogo pedindo para você salvar o arquivo, ou você ver todo o conteúdo do arquivo como texto, há uma grande chance do seu servidor não ter o PHP habilitado ou não estar configurado corretamente. Peça ao seu administrador para habilitar o PHP para você usando o capítulo de Instalação do manual. Se você está desenvolvendo localmente, leia também o capítulo de instalação indicado acima para ter certeza de que configurou tudo corretamente. Confirme que está acessando o arquivo via HTTP através do servidor web. Se acessar o arquivo através do sistema de arquivos, então ele não será interpretado pelo PHP. Caso o problema persista, não hesite em nos chamar nos vários » canais de suporte do PHP.

O objetivo do exemplo é mostrar o formato especial das tags do PHP. Neste exemplo nós usamos <?php para indicar que a partir daquele ponto entramos no modo PHP. Então nós colocamos a instrução do PHP e saímos do modo PHP adicionando a tag de fechamento, ?>. Você pode entrar e sair do modo PHP num arquivo HTML desta maneira em qualquer lugar que você queira. Para mais detalhes, leia a seção do manual que fala da sintaxe básica do PHP.

Nota: Uma Nota sobre Fins de linha

Fins de linha têm pouco significado no HTML, entretanto ainda é uma boa idéia fazer o seu HTML ter uma boa aparência e organização através da divisão em linhas. Um fim de linha imediatamente após a tag de fechamento ?> será removido pelo PHP. Isso pode ser extremamente útil quando você está juntando vários blocos de PHP ou arquivos incluídos contendo apenas PHP que não deveriam imprimir nada no processo. Ao mesmo tempo isso pode ser um pouco confuso.



#FicaDica

Você pode colocar um espaço depois da tag de fechamento ?> para forçar a impressão de um espaço e fim de linha ou você pode explicitamente imprimir o fim de linha com um último echo/print de dentro de seu bloco PHP.

Nota: Uma Nota sobre os Editores de Texto

Há muitos editores de textos e Integrated Development Enviroments (IDEs) que você pode usar para criar, editar e gerenciar arquivos PHP. Uma lista parcial destas ferramentas pode ser vista na » Lista de Editores para PHP. Se você gostaria de recomendar um editor, por favor visite a página acima e peça ao administrador da página para adicionar o editor à lista. Ter um editor que destaca a sintaxe das tags pode ser muito útil.

Nota: Uma Nota sobre os Processadores de Texto

Processadores de texto como o StarOffice Writer, Microsoft Word e Abiword não são boas escolhas para editar arquivos PHP. Se você deseja usar um desses para este script de teste, você precisa verificar se você está salvando o arquivo como texto plano ou o PHP não será capaz de ler e executar o script.

Nota: Uma Nota sobre o Bloco de Notas do Windows

Se você está escrevendo seus scripts PHP usando o Bloco de Notas do Windows, você precisará ter certeza de que seus arquivos estão sendo salvos com a extensão .php. (O Bloco de Notas do Windows adiciona automaticamente a extensão .txt aos arquivos a não ser que você siga um dos passos a seguir para prevenir isto). Quando a caixa de diálogo Salvar estiver aberta e você for digitar o nome do seu arquivo, coloque o nome do arquivo entre aspas (i.e. "ola.php"). Uma alternativa é você clicar na lista drop-down 'Documentos de Texto' na caixa de diálogo 'Salvar' e alterar a opção para "Todos os tipos de arquivos". Você agora pode digitar o nome do seu arquivo sem usar as aspas.

Agora que você criou com sucesso um script simples em PHP, é hora de criar o mais famoso dos scripts PHP! Uma chamada à função phpinfo() e você verá muitas informações úteis sobre seu sistema e configurações como as Variáveis Predefinidas disponíveis, módulos carregados pelo PHP, e as opções de configuração. Tire algum tempo para ver e revisar estas informações importantes.

Exemplo #2 Como obter informações sobre o PHP

```
<?php phpinfo(); ?>
Tags PHP ¶
```

Quando o PHP interpreta um arquivo ele procura pelas tags de abertura e fechamento, <?php e ?>, que dizem ao PHP para iniciar ou parar a interpretação do código entre elas. A interpretação dessa maneira, permite o PHP ser incluído em vários tipos de documentos, pois tudo que está fora dessas tags é ignorado pelo interpretador do PHP.

O PHP também permite a tag curta <? (cujo uso é desencorajado pois essa opção está disponível somente quando habilitada na diretiva short_open_tag no arquivo de configuração php.ini, ou quando o PHP tiver sido compilado com a opção --enable-short-tags).

Se um arquivo for código PHP puro, é preferível omitir a tag de fechamento no final do arquivo. Prevenindo a existência de espaços ou linhas em branco após a tag, que podem causar efeitos indesejáveis, por que o PHP iniciará o buffer de saída quando não existir intenção do programador de enviar alguma saída neste ponto do script.

```
<?php
echo "Hello world";

// ... mais código

echo «última instrução»;
// o script termina aqui, sem tag de fechamento PHP
```

Escapando o HTML

Tudo o que estiver fora das tags PHP é ignorado pelo interpretador, o que permite arquivos PHP de conteúdo misto. Permite que o PHP seja incluído dentro de documentos HTML, para, por exemplo, para a criação de templates.

```
<p>Isto vai ser ignorado pelo PHP em enviado ao navegador.</p>
<?php echo 'Enquanto isto vai ser interpretado.'; ?>
<p>Isto também vai ser ignorado pelo PHP em enviado ao navegador.</p>
```

Isso funcionará porque quando o interpretador do PHP encontra ?>, a tag de fechamento, ele simplesmente começa a repassar qualquer coisa que encontre (exceto um fim de linha imediato, ver a seção sobre separação de instruções), até que ele encontre outra tag de abertura a não ser que esteja no meio de uma instrução condicional, onde então o interpretador vai determinar o resultado da condicional e assim decidir qual caminho tomar. Veja no próximo exemplo.

Utilizando estruturas avançadas

Exemplo #1 Escape avançado usando condições

```
<?php if ($expression == true): ?>
    Isto irá aparecer se a expressão for verdadeira.
<?php else: ?>
    Senão isso que aparecerá.
<?php endif; ?>
```

Nesse exemplo o PHP irá escapar os blocos em que a condição não seja satisfeita, mesmo que o trecho de código esteja fora das tags de abertura/fechamento do PHP, pois o interpretador do PHP, irá pular os conteúdos de blocos que não possuem uma condição que não foi satisfeita.

Para impressão de grandes blocos de texto, sair do modo de interpretação do PHP é geralmente mais eficiente que enviar todo o texto através das funções echo ou print.

No PHP 5 existem cinco diferentes pares de tags de abertura e fechamento disponíveis, dependendo de como o interpretador estiver configurado. Dois deles, <?php ?> e <script language="php"> </script> estão sempre disponíveis. Também a tag curta de echo <?= ?>, que está sempre disponível desde o PHP 5.4.0.

As outras duas opções são as tags curtas e tags estilo ASP. Assim, embora algumas pessoas achem as tags curtas e ASP convenientes, são menos portáteis, e geralmente não recomendadas.

Nota:

Também note que se estiver incluindo o PHP dentro de XML ou XHTML, será necessário o uso das tags <?php ?> para aderência aos padrões.

O PHP 7 removeu o suporte a tags ASP e <script language="php">. Assim, é recomendado utilizar apenas <?php ?> e <?= ?> ao se escrever códigos PHP com maior compatibilidade.

Exemplo #2 Abrindo e Fechando as Tags do PHP

1. `<?php echo 'se você quer servir documentos XHTML ou XML, escreva assim'; ?>`

2. Você pode utilizar também a tag curta de echo para `<?= 'imprimir isso' ?>`.

Ela sempre está disponível do PHP 5.4.0 em diante, e é equivalente a

`<?php echo 'imprimir isso' ?>`.

3. `<? echo 'esse código entre tags curtas somente funcionará '.`

`'se short_open_tag estiver habitado'; ?>`

4. `<script language="php">`

`echo 'alguns editores (como o FrontPage) não suportam processar instruções com tags assim';`
`</script>`

Esta sintaxe foi removida no PHP 7.0.0.

5. `<% echo 'Você também pode utilizar tags no estilo ASP'; %>`

`<%= $variable; %>` é um atalho para `<% echo $variable; %>`

Essas duas sintaxes também foram removidas no PHP 7.0.0.

Tags curtas (exemplo três) estão disponíveis apenas quando são habilitada pela configuração `short_open_tag` no arquivo `php.ini`, ou se o PHP foi configurado com a opção `--enable-short-tags`.

Tags ASP (exemplo cinco) estão disponíveis somente quando habilitadas através da diretiva de configuração `asp_tags` no arquivo `php.ini`, e foram removidas no PHP 7.0.0.

Separação de instruções

Como no C ou Perl, o PHP requer que as instruções sejam terminadas com um ponto-e-vírgula ao final de cada comando. A tag de fechamento de um bloco de código PHP automaticamente implica em um ponto-e-vírgula; você não precisa ter um ponto-e-vírgula terminando a última linha de um bloco PHP. A tag de fechamento do bloco irá incluir uma nova linha logo após, se estiver presente.

```
<?php
echo 'Isto é um teste';
?>
```

```
<?php echo 'Isto é um teste' ?>
```

```
<?php echo 'Nós omitimos a última tag de fechamento';
```

Comentários ¶

O PHP suporta comentários no estilo 'C', 'C++' e do Unix shell (estilo Perl). Por exemplo:

```
<?php
echo 'Isto é um teste'; // Estilo de comentário de
uma linha em c++
/* Este é um comentário de múltiplas linhas
ainda outra linha de comentário */
echo 'Isto é ainda outro teste';
echo 'Um teste final'; # Este é um comentário de
uma linha no estilo shell
?>
```

Os comentários de estilo "uma linha" apenas comentam até o final da linha ou do bloco PHP de código corrente, o que chegar primeiro. Isto significa que o código HTML após `// ... ?>` ou `# ... ?>` SERÁ impresso: `?>` interrompe o modo PHP e retorna para o modo HTML, e `//` ou `#` não podem influenciar isto. Se a diretiva de configuração `asp_tags` estiver habilitada, se comportará da mesma maneira que `// %>` e `# %>`. Entretanto, a tag `</script>` não interrompe o modo PHP em um comentário de uma linha.

```
<h1>Isto é um <?php # echo 'simples';?> exemplo.</h1>
```

```
<p>O cabeçalho acima irá dizer 'Isto é um exemplo'.</p>
```

Comentários no estilo 'C' terminam ao primeiro `*/` encontrado. Tenha certeza de não aninhar comentários no estilo 'C'. É fácil fazer este equívoco se estiver tentando comentar grandes blocos de código.

```
<?php
/*
echo 'Isto é um teste'; /* Este comentário irá causar
um problema */
*/
?>
```

Tipos

O PHP suporta oito tipos primitivos.

Quatro tipos escalares:

boolean

integer

float (número de ponto flutuante, ou também double)

string

Três tipos compostos:

array

object

callable

E finalmente dois tipos especiais:

resource

NULL

pseudo-tipos:

mixed

number

callback (também chamado callable)

array|object

void

E a pseudo-variável `$...`