

PHP - Hypertext Preprocessor



INSTITUTO FEDERAL
GOIANO

PHP - Hypertext Preprocessor
Prof. Marcos Alves Vieira
marcos.vieira@ifgoiano.edu.br

O HTML é limitado

- ▶ Apesar de toda evolução do HTML e dos navegadores atuais, a Web ainda é um ambiente bastante restrito
- ▶ Todo projeto Web sério deve ir além do HTML, CSS e JavaScript
- ▶ Existem diversas linguagens que podem ser usadas no ambiente Web



Ruby



python

Microsoft
C#.net

- ▶ Com o uso de uma linguagem de programação, é possível:
 - ▶ gerar páginas dinamicamente com dados de um banco de dados
 - ▶ enviar *emails* para usuários
 - ▶ processar tarefas complexas
 - ▶ garantir validações de segurança
 - ▶ e muito mais!



INSTITUTO FEDERAL
GOIANO

PHP - Hypertext Preprocessor
Prof. Marcos Alves Vieira
marcos.vieira@ifgoiano.edu.br

Páginas estáticas vs. páginas dinâmicas

- ▶ Página estática
 - ▶ Todo conteúdo é fixo
 - ▶ Não existe interação entre o usuário e a página
- ▶ Página dinâmica
 - ▶ O conteúdo é gerado a cada visita (requisição) à página
 - ▶ Bancos de dados, acesso a fontes externas, etc...
 - ▶ Possibilita interação entre o usuário e a página
 - ▶ Formulários

História do PHP

- ▶ 1995 - PHP/FI (Personal Home Pages/Forms Interpreter)
- ▶ 1997 - PHP/FI 2
 - ▶ Utilizado por 50 mil domínios (1% dos sites na época)
- ▶ 1998 - PHP 3 (PHP: Hypertext Preprocessor)
 - ▶ Zeev Suraski e Andi Gutmans
 - ▶ Utilizado por 10% dos sites
- ▶ 2000 - PHP 4
 - ▶ Reescrita do núcleo do PHP
 - ▶ Mais recursos e melhoria de performance
 - ▶ Zend Engine
- ▶ 2004 - PHP 5
 - ▶ Maior suporte a orientação a objetos
- ▶ Atualmente o PHP encontra-se em sua versão 5.6.5
- ▶ Estima-se que o PHP esteja em cerca de 40% de todos os sites atuais (2013 - [fonte](#))



Rasmus Lerdorf
Criador do PHP

O que o PHP pode fazer?

- ▶ Scripts do lado do servidor (*server-side*)
 - ▶ Interpretador PHP (CGI)
 - ▶ Servidor Web
 - ▶ Navegador Web
- ▶ Scripts em linha de comando
 - ▶ Não necessita de navegador
- ▶ Aplicações desktop
 - ▶ PHP-GTK

O que o PHP pode fazer? *(cont.)*

- ▶ Geração dinâmica de diversos tipos de arquivos, além de HTML
 - ▶ Imagens, PDF, Animações Flash, XML, ...
- ▶ Conexão com banco de dados
- ▶ Comunicação com outros serviços
 - ▶ LDAP, IMAP, SMTP, POP3, HTTP, COM (Windows)
- ▶ E muito mais!

Características do PHP

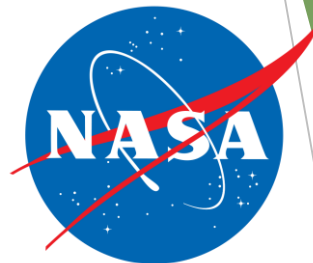
- ▶ Software livre
- ▶ Portabilidade
 - ▶ Multiplataforma
- ▶ Simplicidade
 - ▶ Rápido desenvolvimento
- ▶ Adaptabilidade
 - ▶ Procedural e/ou Orientado a Objetos
- ▶ Performance

O site oficial do PHP é <http://www.php.net>

Lá você encontra *downloads*, código fonte do PHP e um manual completo da linguagem, incluindo comentários de usuários.

Quem utiliza PHP?

facebook



Baidu 百度

YAHOO!

Google



ebay

SONY



flickr

sourceforge

W3C

O que é preciso para o PHP funcionar?

- ▶ Servidor web
 - ▶ Apache HTTP Server
 - ▶ Microsoft IIS
 - ▶ Zend Server



zend[®]Server



- ▶ Interpretador PHP

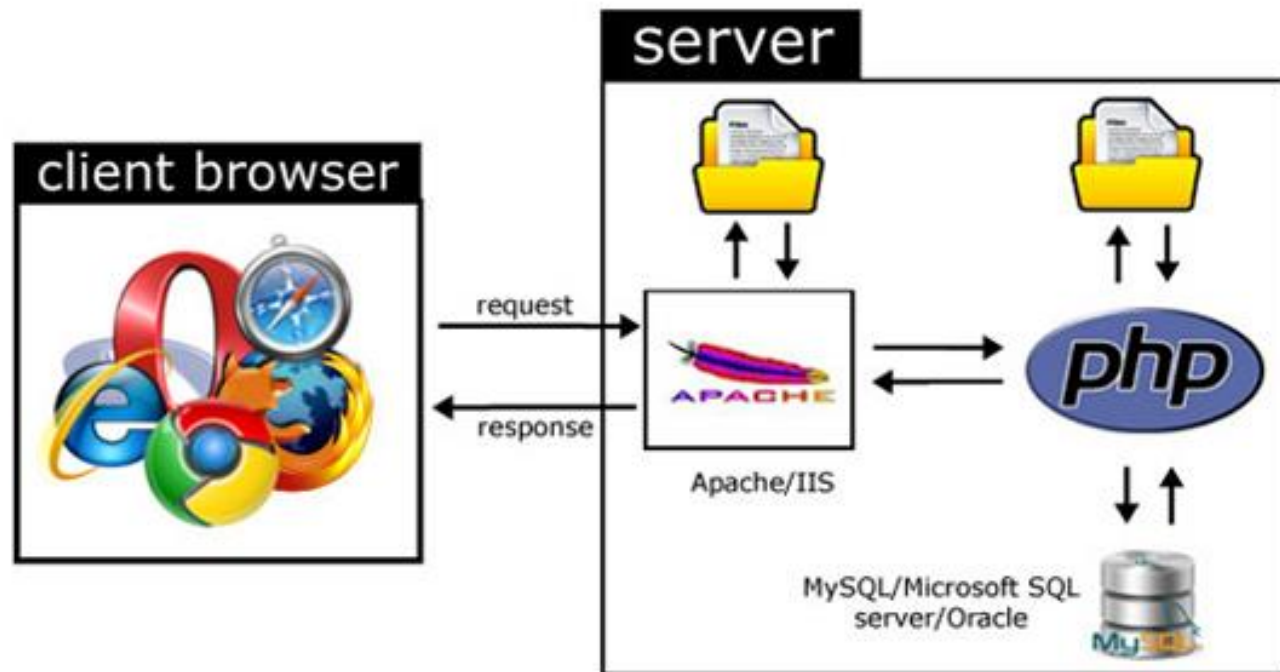


- ▶ Navegador Web



Como o PHP funciona?

Basic Structure



= File System



= Database

webdevtutes.blogspot.com



INSTITUTO FEDERAL
GOIANO

PHP - Hypertext Preprocessor
Prof. Marcos Alves Vieira
marcos.vieira@ifgoiano.edu.br

O que é um servidor web?

- ▶ Programa responsável por aceitar requisições HTTP, trata-las e, posteriormente, devolver a resposta para o requisitante
 - ▶ As requisições são geralmente originadas de navegadores (*browsers*) web
 - ▶ A resposta é devolvida em formato HTML e pode incluir objetos embutidos
 - ▶ Imagens
 - ▶ Vídeos
 - ▶ Áudio
 - ▶ Demais tipos de arquivo (PDF, XML, etc...)
- ▶ Exemplos de servidores web:
 - ▶ Apache, Tomcat, JBoss, IIS, nginx e outros

NGINX



APACHE

Facilite sua vida: simplifique

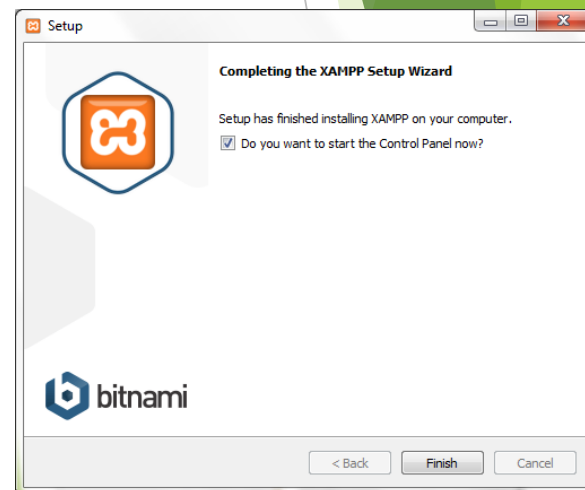
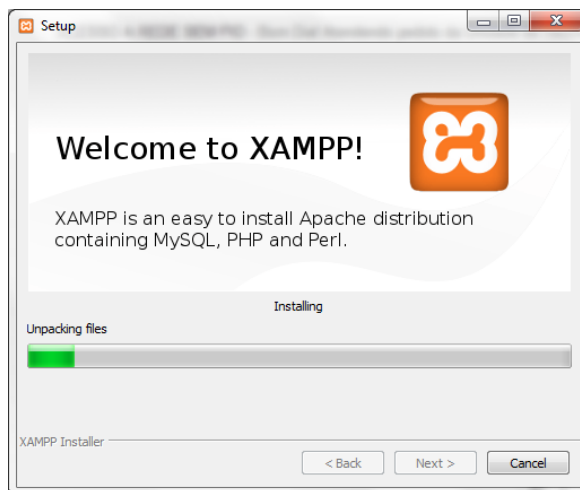
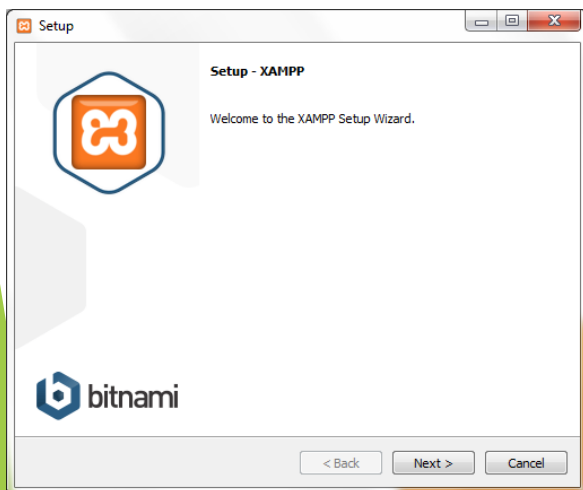
- ▶ XAMPP: Apache + MySQL + PHP + Perl
 - ▶ www.apachefriends.org
 - ▶ Multiplataforma (Windows, Linux e OS X)
 - ▶ **MUITO** fácil de instalar
 - ▶ Inclui phpMyAdmin



XAMPP

Instalação

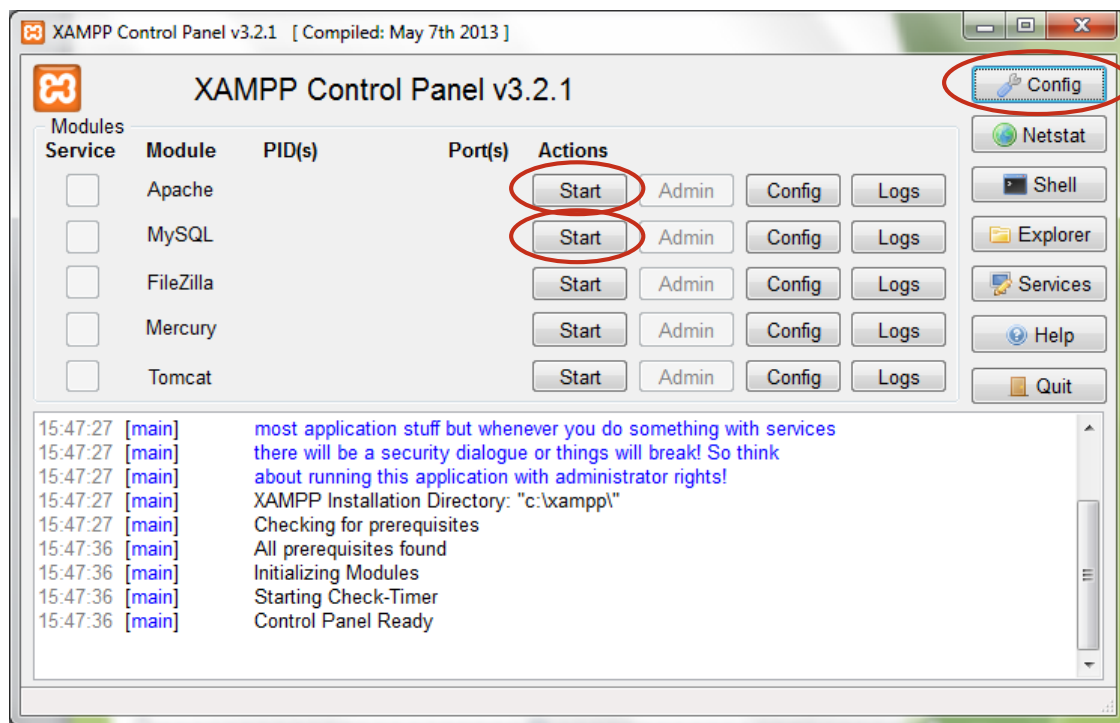
- ▶ Muita atenção na instalação supercomplicada:
 - ▶ Next, Next ... Finish



XAMPP

Configuração

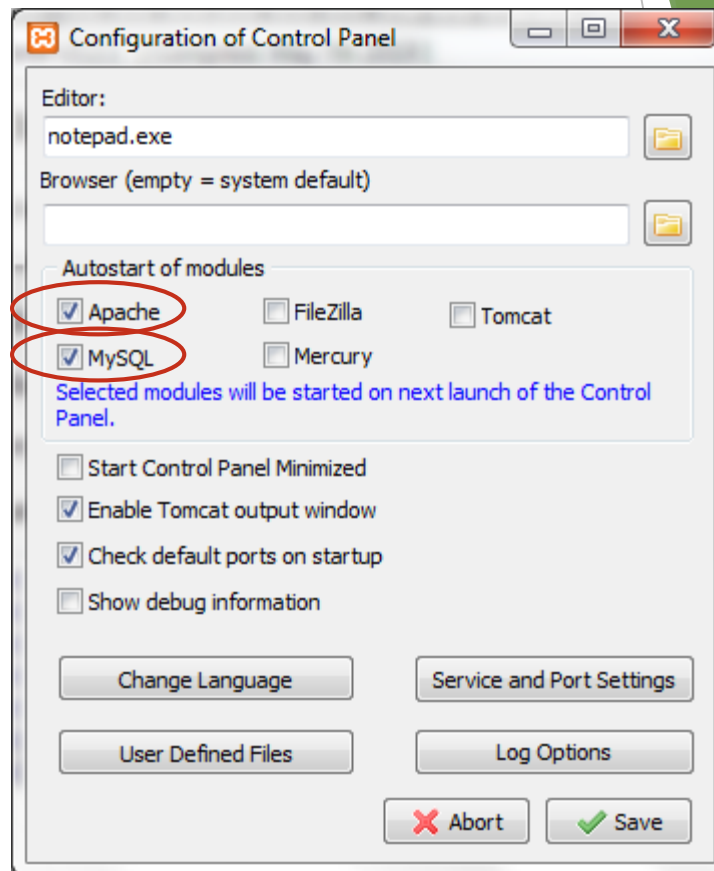
- Iniciar os módulos Apache e MySQL e clicar em "Config"



XAMPP

Configuração (continuação)

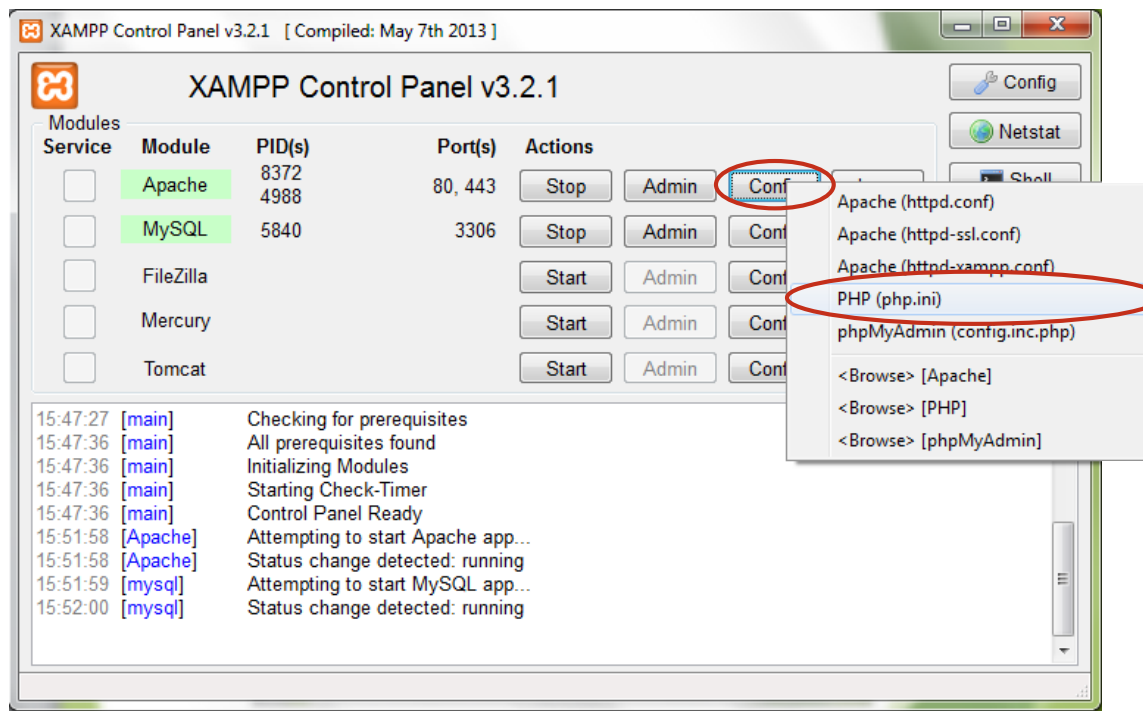
- ▶ Marcar os módulos para início automático:
 - ▶ Apache e MySQL
- ▶ Clicar em "Save"



XAMPP

Configuração (continuação)

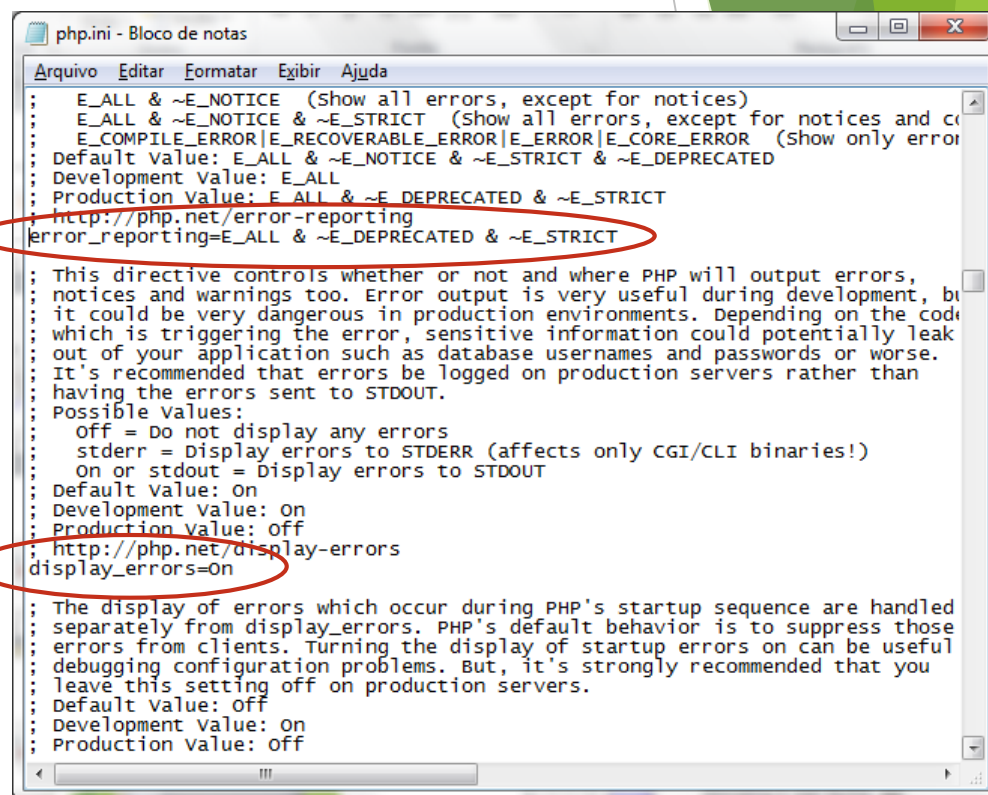
► Editar o arquivo php.ini



XAMPP

Configuração (continuação)

- ▶ Confirmar se as diretivas **error_reporting** e **display_errors** estão ativas (*descomentadas*)
- ▶ Caso seja necessário editar algo
 - ▶ Salvar o arquivo e reiniciar o Apache
 - ▶ Pelo console do XAMPP
 - ▶ Botão "Stop", depois "Start"



```
php.ini - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
; E_ALL & ~E_NOTICE (Show all errors, except for notices)
; E_ALL & ~E_NOTICE & ~E_STRICT (Show all errors, except for notices and c
; E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR (Show only error
; Default value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
; Development value: E_ALL
; Production value: E_ALL & ~E_DEPRECATED & ~E_STRICT
; http://php.net/error-reporting
error_reporting=E_ALL & ~E_DEPRECATED & ~E_STRICT

; This directive controls whether or not and where PHP will output errors,
; notices and warnings too. Error output is very useful during development, b
; it could be very dangerous in production environments. Depending on the cod
; which is triggering the error, sensitive information could potentially leak
; out of your application such as database usernames and passwords or worse.
; It's recommended that errors be logged on production servers rather than
; having the errors sent to STDOUT.
; Possible values:
;   off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default value: On
; Development value: On
; Production value: off
; http://php.net/display-errors
display_errors=on

; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. PHP's default behavior is to suppress those
; errors from clients. Turning the display of startup errors on can be useful
; debugging configuration problems. But, it's strongly recommended that you
; leave this setting off on production servers.
; Default value: off
; Development value: on
; Production value: off
```

XAMPP

Teste

- ▶ Abra o navegador e acesse a URL:
 - ▶ <http://localhost>
- ▶ Pronto! Tudo está OK :-)
- ▶ Os arquivos (PHP, HTML, imagens...) devem ser colocados na pasta
 - ▶ C:\xampp\htdocs
- ▶ Caso queira, pode apagar todo conteúdo inicial desta pasta



Como editar arquivos PHP?

► Editores de texto

- Bloco de notas
- Notepad++
- gedit
- vim



► IDE (*Integrated Development Environment*)

- Eclipse
- NetBeans
- ZendStudio (\$\$\$\$\$\$\$\$\$\$)
- Aptana



NetBeans

zendstudio



aptana

Extensão de arquivos PHP

- Geralmente, os arquivos PHP são nomeados da seguinte forma:

Extensão	Significado
.php	Arquivo PHP contendo um programa
.class.php	Arquivo PHP contendo uma classe
.inc.php	Arquivo PHP a ser incluído, contendo constantes, funções ou configurações

- Outras extensões podem ser encontradas em programas antigos:

Extensão	Significado
.php3	Arquivo contendo um programa PHP na versão 3
.php4	Arquivo contendo um programa PHP na versão 4
.phtml	Arquivo contendo um programa PHP e um arquivo HTML

Delimitadores de código

- O código de um programa PHP deve estar contido entre os delimitadores:

```
<?php  
    //código;  
    //código;  
    //código;  
?>
```

Obs.: os comandos devem ser delimitados com ponto-e-vírgula (;)

Exercício 1

- a) Crie o arquivo `ex1.php` com o seguinte conteúdo:

```
<?php
    echo "Olá, mundo!";
?>
```

- b) Salve na pasta raiz de documentos do seu servidor web (`C:\xampp\htdocs`)

- c) Abra o navegador e acesse o endereço:

```
http://localhost/ex1.php
```

- d) Visualize o código-fonte da página gerada (Ctrl+U). Onde está o código PHP?

- e) Altere o código do arquivo, adicionando *tags* HTML:

```
<?php
    echo "<p><b>Olá</b>, <u>mundo</u>!</p>";
?>
```

- f) Acesse novamente o arquivo pelo navegador e visualize o código-fonte

Comentários

► Comentários de uma linha

- Utilizamos duas barras (//) ou o símbolo #

```
<?php
    //Este código imprime o texto 'Olá, mundo'
    echo "Olá, mundo!";
?>
```

► Comentários de múltiplas linhas

- Devem estar entre /* e */

```
<?php
    /*Este código imprime
    o texto 'Olá, mundo'*/
    echo "Olá, mundo!";
?>
```

Nota

Comentários são ignorados pelo interpretador do PHP

Comandos de saída (*output*)

- ▶ Estes comandos são utilizados para gerar uma saída na tela
 - ▶ Se o PHP for utilizado na linha de comando (*prompt* do sistema), a saída será no próprio console
 - ▶ Se o programa for executado via servidor web, a saída será exibida na página HTML gerada

○ **echo**

Imprime uma ou mais variáveis ou *strings*

```
echo "Olá, mundo!"
```

○ **print**

Imprime uma variável ou *string*

```
print "Olá, mundo!"
```

Pesquisa

Pesquise a(s) diferença(s) entre `echo` e `print`.

Tempo: 5 minutos

Exercício 2

Podemos mesclar trechos de código PHP com HTML

a) Crie o arquivo `ex2.php` com o seguinte conteúdo e salve na pasta raiz de documentos do servidor web:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Aprendendo PHP</title>
5 <meta charset="utf-8">
6 </head>
7 <body>
8 <h1>Mesclando HTML com PHP</h1>
9 <?php
10     echo "<p>Primeira linha</p>";
11     echo "<p>Segunda linha</p>";
12     echo "<p>Terceira linha</p>";
13 ?>
14 </body>
15 </html>
```

b) Acesse o arquivo pelo navegador e visualize o código-fonte



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Aprendendo PHP</title>
5 <meta charset="utf-8">
6 </head>
7 <body>
8 <h1>Mesclando HTML com PHP</h1>
9 <p>Primeira linha</p><p>Segunda linha</p><p>Terceira linha</p></body>
10 </html>
```