



PHP: Hypertext Preprocessor"

Personal Home Page

```
index.php x
1 <?php
2 $link = new mysqli('localhost', 'root', '', '');
3 $link->set_charset('utf8');
4
```

Cronograma

- 19/09, Aula 01: Setup, Hello-World, Algoritmos
- 21/09, Aula 02: MVC, Laravel
- 25/09, Aula 03: Banco de dados
- 26/09, Aula 04: Banco de dados
- 28/09: Aula 05: Backend Crud - Área Administrativa
- 02/10: Aula 06: Backend Crud - Área Administrativa
- 03/10: Aula 07: Backend Crud - API Rest
- 05/10: Aula XX: Integração PHP + HMTL



Surgido em 1995

Última versão 8.2.10 (31 / 08 / 2023)

Estilo de tipagem: dinâmica e fraca

Página oficial www.php.net

Licença:

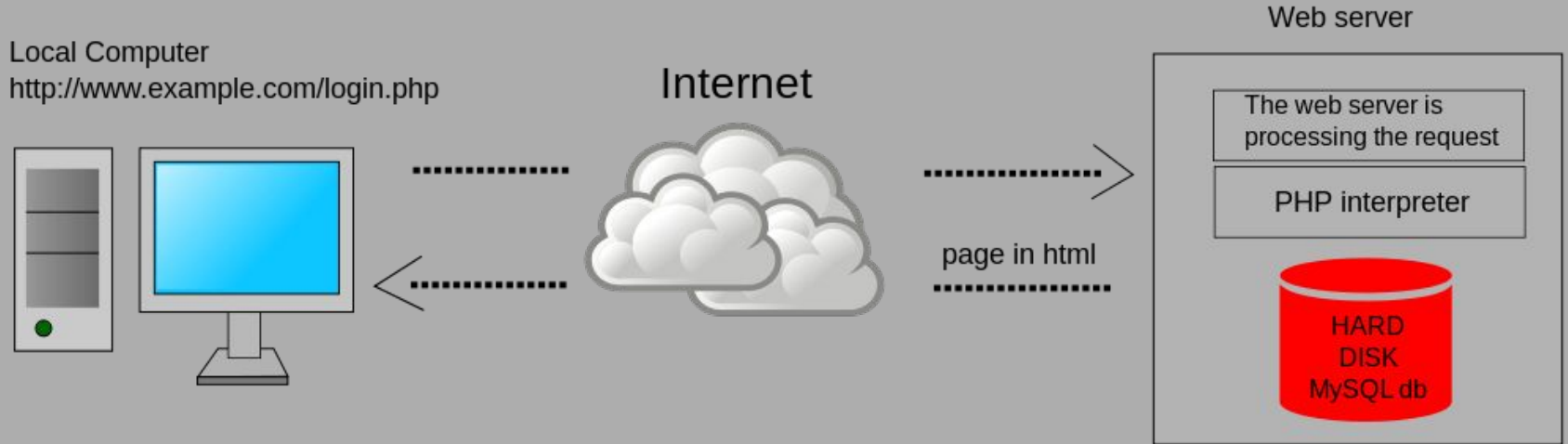
PHP é um software gratuito e de código aberto disponível sob a PHP License, que afirma:

Produtos derivados deste software não devem ser chamado de PHP, nem pode conter "PHP" em seu nome, sem prévia permissão por escrito da group@php.net.

Você pode indicar que o software funciona em conjunto com o PHP,

dizendo "Foo para PHP", em vez de chamá-lo "PHP Foo" ou "phpfoo".

Como funciona



Setup - Instalação Manual

Apache

There are several builds of Apache2 for Windows. The Apache builds of ApacheLounge are recommended, but other options include XAMPP, WampServer and BitNami, which provide automatic installer tools. PHP can be used on Apache through mod_php or mod_fastcgi. mod_php requires a TS build of Apache built with same version of Visual C and same CPU (x86 or x64).

Choose Build

Windows builds can be downloaded from » <http://windows.php.net/download/>. All builds are optimized (PGO), and QA and GA releases are thoroughly tested.

There are 4 types of PHP builds:

- Thread-Safe(TS) - for single process web servers, like Apache with mod_php
- Non-Thread-Safe(NTS) - for IIS and other FastCGI web servers (Apache with mod_fastcgi) and recommended for command-line scripts
- x86 - for 32-bits systems.
- x64 - for 64-bits systems.

IIS

IIS is built in to Windows. On Windows Server, the IIS role can be added via the Server Manager. The CGI Role Feature needs to be included. On Windows Desktop, IIS has to be added via the Control Panel's Add/Remove Programs. The Microsoft documentation has » [detailed instructions](#). For desktop web apps and web-development, IIS/Express or PHP Desktop can also be used.

Exemplo #1 Command line to configure IIS and PHP

```
@echo off

REM download .ZIP file of PHP build from http://windows.php.net/downloads/

REM path to directory you decompressed PHP .ZIP file into (no trailing \)
set php_path=c:\php

REM Clear current PHP handlers
%windir%\system32\inetsrv\appcmd clear config /section:system.webServer/fastCGI
REM The following command will generate an error message if PHP is not installed. This can be ignored.
%windir%\system32\inetsrv\appcmd set config /section:system.webServer/handlers /-["name='PHP_via_FastCGI'"]

REM Set up the PHP handler
%windir%\system32\inetsrv\appcmd set config /section:system.webServer/fastCGI /+["fullPath='%php_path%\php-cgi.exe'"]
%windir%\system32\inetsrv\appcmd set config /section:system.webServer/handlers /+
["name='PHP_via_FastCGI',path='*.php',verb='*',modules='FastCgiModule',scriptProcessor='%php_path%\php-cgi.exe',resourceType='Unspecified'"]
%windir%\system32\inetsrv\appcmd set config /section:system.webServer/handlers /accessPolicy:Read,Script

REM Configure FastCGI Variables
%windir%\system32\inetsrv\appcmd set config -section:system.webServer/fastCgi /["fullPath='%php_path%\php-cgi.exe'"].instanceMaxRequests:10000
%windir%\system32\inetsrv\appcmd.exe set config -section:system.webServer/fastCgi /+
["fullPath='%php_path%\php-cgi.exe'"].environmentVariables.[name='PHP_FCGI_MAX_REQUESTS',value='10000']"
%windir%\system32\inetsrv\appcmd.exe set config -section:system.webServer/fastCgi /+
["fullPath='%php_path%\php-cgi.exe'"].environmentVariables.[name='PHPRC',value='%php_path%\php.ini']"
```

● https://www.php.net/manual/pt_BR/install.windows.manual.php

Setup - Instalação facilitada



WAMP

<https://sourceforge.net/projects/wampserver/files/latest/download>



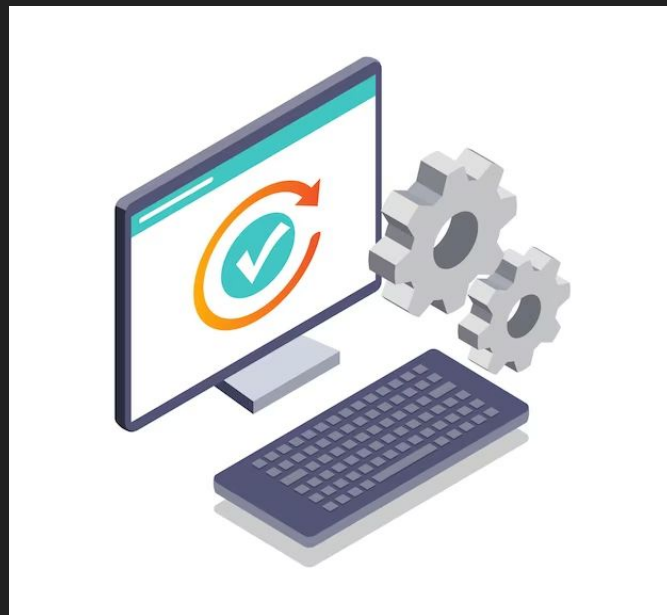
XAMPP

<https://www.apachefriends.org/index.html>

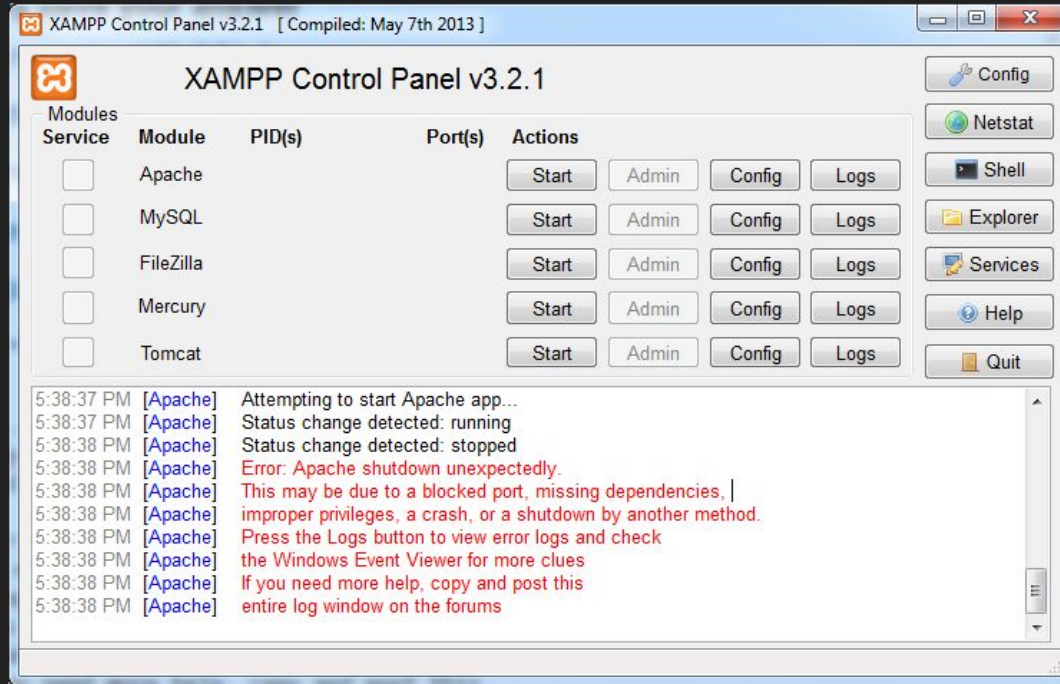
Instalação...



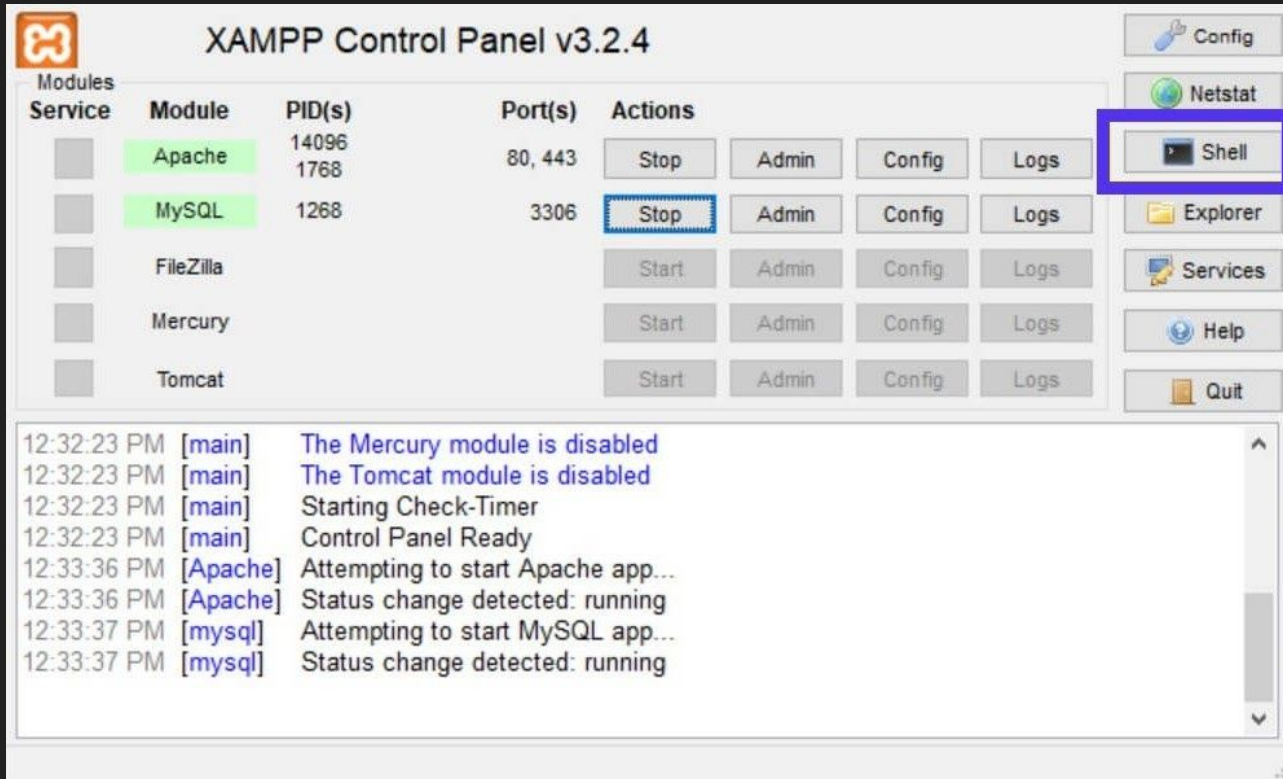
XAMPP



Gerenciando os serviços



Serviços ativos: HTTP + Banco de Dados



The screenshot displays the XAMPP Control Panel v3.2.4 interface. The main table lists the status of various modules. Apache and MySQL are highlighted in green, indicating they are running. The 'Stop' button for MySQL is highlighted with a blue dashed border. A blue rectangle highlights the 'Shell' button in the right-hand sidebar. The log at the bottom shows the sequence of events, including the disabling of Mercury and Tomcat modules and the successful startup of Apache and MySQL.

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	14096 1768	80, 443	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL	1268	3306	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

Right-hand sidebar buttons: Config, Netstat, Shell, Explorer, Services, Help, Quit.

Log entries:

- 12:32:23 PM [main] The Mercury module is disabled
- 12:32:23 PM [main] The Tomcat module is disabled
- 12:32:23 PM [main] Starting Check-Timer
- 12:32:23 PM [main] Control Panel Ready
- 12:33:36 PM [Apache] Attempting to start Apache app...
- 12:33:36 PM [Apache] Status change detected: running
- 12:33:37 PM [mysql] Attempting to start MySQL app...
- 12:33:37 PM [mysql] Status change detected: running

HelloWorld

Incluindo PHP in a File

Arquivos PHP terminam em .php.

Além do próprio PHP, eles podem conter texto, HTML, CSS e JavaScript.

Para que um navegador reconheça o PHP, você precisa colocá-lo entre colchetes:

`<?php ?>`

Consequentemente, você pode executar PHP em uma página:

```
<?php
//coloque o código PHP aqui
?>
```

Algoritmos

Escrevendo comentários

// Indica comentários que abrangem apenas uma linha

Outra forma de produzir comentários de uma única linha

/*...*/ Tudo entre /*e */ não é executado, também funciona em várias linhas

```
/*
Nome do Tema: Vinte e Dezesete
URI do tema: https://wordpress.or
Autor: a equipe do WordPress
URI do autor: https://wordpress.o
Descrição: Twenty Seventeen dá vi
Versão: 1.5
Licença: Licença Pública Geral GN
URI de licença: https://www.gnu.o
Domínio do texto: vinte e dezesse
Tags: uma coluna, duas colunas, b

Este tema, assim como o WordPress
Use-o para criar algo legal, dive
*/
```

Saída de dados

Em PHP, os dados geralmente são gerados usando **echo** ou **print**.

Por exemplo, o título dessa postagem do blog pode ser exibido em uma página como esta:

```
<?php  
  
echo "<h1>Folha de dicas do PHP</h1>";  
  
?>
```

Uma observação importante: como todos os outros comandos PHP, as funções **echo** e **print** não diferenciam maiúsculas de minúsculas. Isso significa que quando você escreve **ECHO**, ou qualquer outra variação, eles continuarão funcionando **EcHo**. **eCH0**

Operadores

Operadores aritméticos

Seus operadores matemáticos padrão.

+ Adição

- Subtração

* Multiplicação

/ Divisão

% Módulo (resto do valor dividido por outro)

** Exponenciação

Operadores de Atribuição

Além do operador de atribuição padrão (=), você também tem as seguintes opções:

$+=$ $a += b$ é o mesmo que $a = a + b$

$-=$ $a -= b$ é o mesmo que $a = a - b$

$*=$ $a *= b$ é o mesmo que $a = a * b$

$/=$ $a /= b$ é o mesmo que $a = a / b$

$\%=$ $a \% b$ é o mesmo que $a = a \% b$

Operadores de comparação

== Igual

=== Idêntico

!= Não igual

<> Não igual

!== Não idênticos

< Menor que

> Maior que

<= Menos que ou igual a

>= Melhor que ou igual a

<=> Menor que, igual ou maior que

Operadores lógicos

and E

or Ou

xor Exclusivo ou

! Não

&& E

|| Ou

Operadores bit a bit

& E

| Ou (inclusivo ou)

^ Xor (exclusivo ou)

~ Não

<< Deslocar para a esquerda

>> Deslocar para a direita

Operadores de string

- . Usado para concatenar (média combinar) argumentos
- .= Usado para anexar o argumento da direita ao argumento do lado esquerdo

Operadores de incremento/decremento

`++$v` Aumenta uma variável em um e depois a retorna

`$v++` Retorna uma variável e depois a incrementa em um

`--$v` Diminui a variável em um, retorna-a depois

`$v--` Retorna a variável e depois a diminui em um

Estruturas de Condição

IF

```
if (condição) {  
    //código a ser executado se a condição for atendida  
}
```

```
IF (condição) {  
    //código a ser executado se a condição for atendida  
}else {  
    //código a ser executado se a condição não for atendida  
}
```


SWITCH

```
switch (n) {  
    case x:  
        código a ser executado se n=x;  
        break;  
    case y:  
        código a ser executado se n=y;  
        break;  
    case z:  
        código a ser executado se n=z;  
        break;  
  
    //adiciona mais casos conforme necessário  
  
    default:  
        código a ser executado se n não for nenhuma das opções acima;  
}  

```

Estruturas de repetição

FOR

```
for (valor inicial; valor final; incremento) {  
    // código para executar vai aqui  
}
```

FOREACH

```
foreach ($InsertYourArrayName as $valor) {  
    // código para executar vai aqui  
}
```

WHILE

```
while (condição que deve ser aplicada) {  
    // código para executar vai aqui  
}
```

DO..WHILE

```
do {
```

```
    // o código a ser executado vai aqui;
```

```
} while (condição que deve ser aplicada);
```

Variáveis

Variáveis e Constantes

```
<?php
```

```
$BlogPostTitle = "Curso de  
PHP da PM-GO e Senai Fatesg";
```

Alguns pontos importantes:

```
?>
```

- As variáveis precisam começar com uma letra ou sublinhado (`_`) e só podem ser compostas por caracteres alfanuméricos
- Variáveis PHP diferenciam maiúsculas de minúsculas, isso significa `$myVar` e `$myvar` não são a mesma coisa
- Se a sua variável consistir em mais de uma palavra, escreva-a `$my_variable` ou `$myVariable`

Escrevendo funções PHP

```
function NomeDaFunção() {
```

```
//coloque o código PHP aqui
```

```
}
```

Tipos de dados

Tipos de dados - Inteiros

Inteiros – Inteiros são números não decimais entre -2.147.483.648 e 2.147.483.647.

Eles devem ter pelo menos um dígito e nenhuma casa decimal.

Pode ser decimal, hexadecimal ou octal.

```
2
3  $x = 24;
4  echo is_int($x); // 1 == true
5
```

Tipos de dados - Floats

Floats - Este é o nome dos números com vírgula decimal ou na forma exponencial.

```
5  
6  $y = 10.365;  
7  echo is_float($y); // 1 == true  
8
```

Tipos de dados - Strings

Strings - Isso significa simplesmente texto.

```
10 <?php
11     $name = "Daniel";
12     echo $name." is a handsome fellow!";
```

Tipos de dados - Boolean

Valores booleanos – Significando declarações verdadeiras/falsas.

```
<?php
```

```
    $bool1 = True;
```

```
    $bool2 = False;
```

```
?>
```

Tipos de dados - Matriz / Array

Matrizes – Matrizes são variáveis que armazenam diversos valores.

```
<?php  
$a = array("a", "b", "c", "d", "e", "f");
```

Tipos de dados - Objetos

Objetos – Os objetos armazenam dados e informações sobre como processá-los.

```
class className
{
    //variable scope must be provided (public, protected or private)
    public $variableName = value;
    function functionName()
    {
        //code;
    }
}

$obj = new className; //$obj variable becomes object data type
$obj->functionName();
$obj->variableName;
```


Tipos de dados - Recursos

São referências a funções e recursos fora do PHP.

- `curl_init()`
- `dba_open()`
- `ftp_connect()`
- `mysql_connect()`

- `pg_connect()`
- `opendir()`
- `fopen()`
- `var_dump()`

<https://www.php.net/manual/en/resource.php>

Funções

Escrevendo funções PHP

```
function NomeDaFunção() {
```

```
//coloque o código PHP aqui
```

```
}
```

Escopo Variável

As variáveis podem estar disponíveis em diferentes escopos, ou seja, a parte de um script que você pode acessá-las. Isso pode ser **global** , **local** e **estático** .

Escopo Variável - Global

Qualquer variável declarada fora de uma função está disponível globalmente.

Isso significa que ele também pode ser acessado fora de uma função.

Escopo Variável - Local

Se você declarar uma variável dentro de uma função, ela terá um escopo local.

A consequência é que ele só pode ser acessado dentro dessa função.

Escopo Variável - Global

Uma maneira de fazer isso é acrescentar uma variável local como global. Dessa forma, passa a fazer parte do escopo global.

```
function minhaFunção() {  
    globais $a, $b;  
    $b = $a - $b;  
}
```

Escopo Variável - Local

Também é possível adicionar `static` na frente de uma variável local.

Dessa forma, ele não será excluído após a execução de sua função e poderá ser reutilizado.

```
function myFunction() {  
    static $count = 0;  
    $count++;  
    echo $count;  
}
```

```
myFunction(); // Output: 1  
myFunction(); // Output: 2  
myFunction(); // Output: 3
```


Variáveis predefinidas

O PHP também vem com uma série de variáveis padrão chamadas superglobais . Isso porque eles são acessíveis de qualquer lugar, independente do escopo.

`$GLOBALS`

`$_SERVER`

`$_GET`

`$_POST`

`$_REQUEST`

Funções de manipulação de variáveis

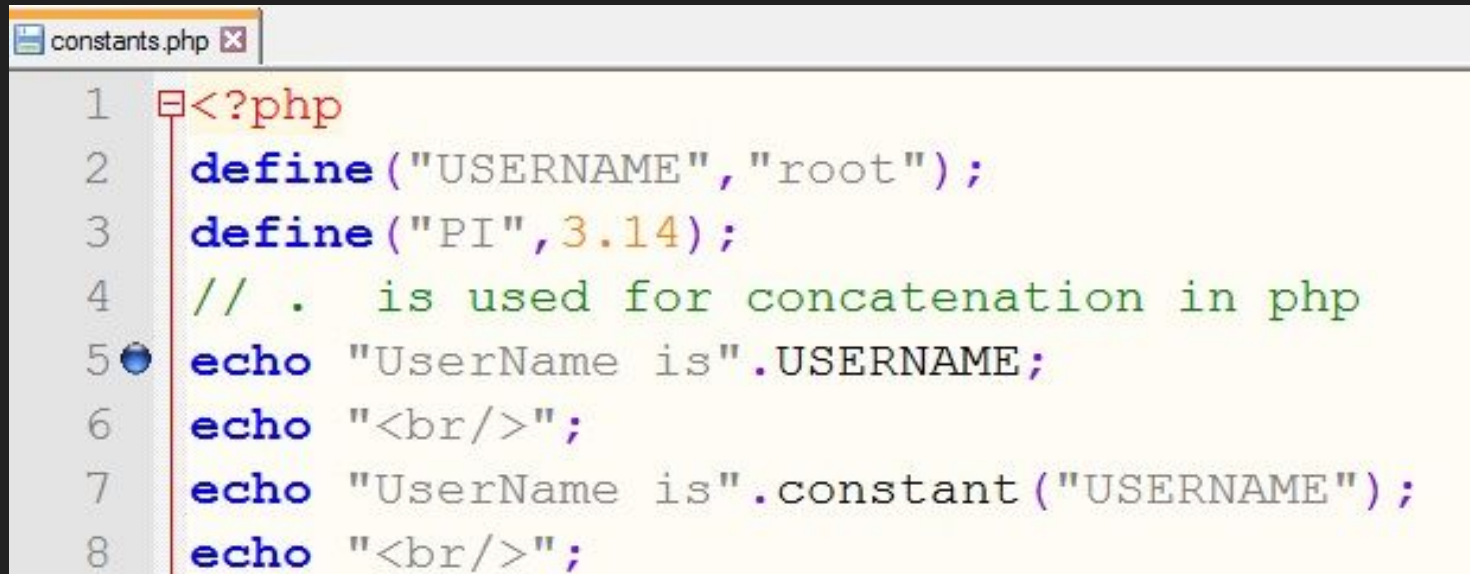
Além disso, há um monte de funções para trabalhar com variáveis:

- `boolval`— Usado para recuperar o valor booleano de uma variável
- `empty`— Verifica se uma variável está vazia ou não
- `floatval`— Obtenha o valor flutuante de uma variável (`doubleval` é outra possibilidade)
- `intval`— Encontre o valor inteiro de uma variável
- `is_bool`— Descubra se uma variável é booleana
- `is_array`— Verifica se uma variável é um array

https://www.php.net/manual/pt_BR/ref.var.php

Constantes

`define(nome, valor, verdadeiro/falso)`



```
1 <?php
2 define("USERNAME", "root");
3 define("PI", 3.14);
4 // . is used for concatenation in php
5 echo "UserName is".USERNAME;
6 echo "<br/>";
7 echo "UserName is".constant("USERNAME");
8 echo "<br/>";
```

Leia
mais

Leia mais

- PHP Roadmap
 - <https://github.com/thecodeholic/php-developer-roadmap>
 - <https://dev.to/he110/php-backend-roadmap-2j1a>
- Manipulação de variáveis
 - https://www.php.net/manual/pt_BR/ref.var.php
- Tipo de Recursos
 - <https://www.php.net/manual/en/resource.php>

Leia mais

- PHP Roadmap
 - <https://github.com/thecodeholic/php-developer-roadmap>
 - <https://dev.to/he110/php-backend-roadmap-2j1a>
- Manipulação de variáveis
 - https://www.php.net/manual/pt_BR/ref.var.php
- Tipo de Recursos
 - <https://www.php.net/manual/en/resource.php>