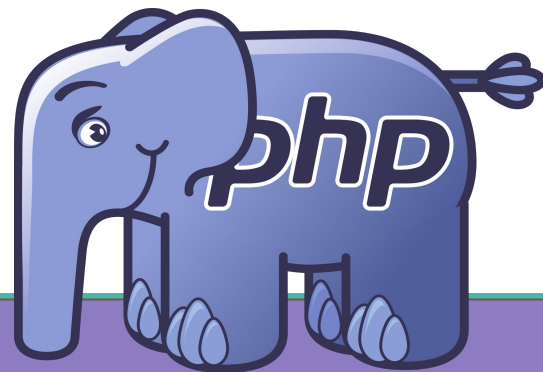




Programador Web



PHP - Fundamentos da Linguagem

Adriel Sales

O que é PHP?



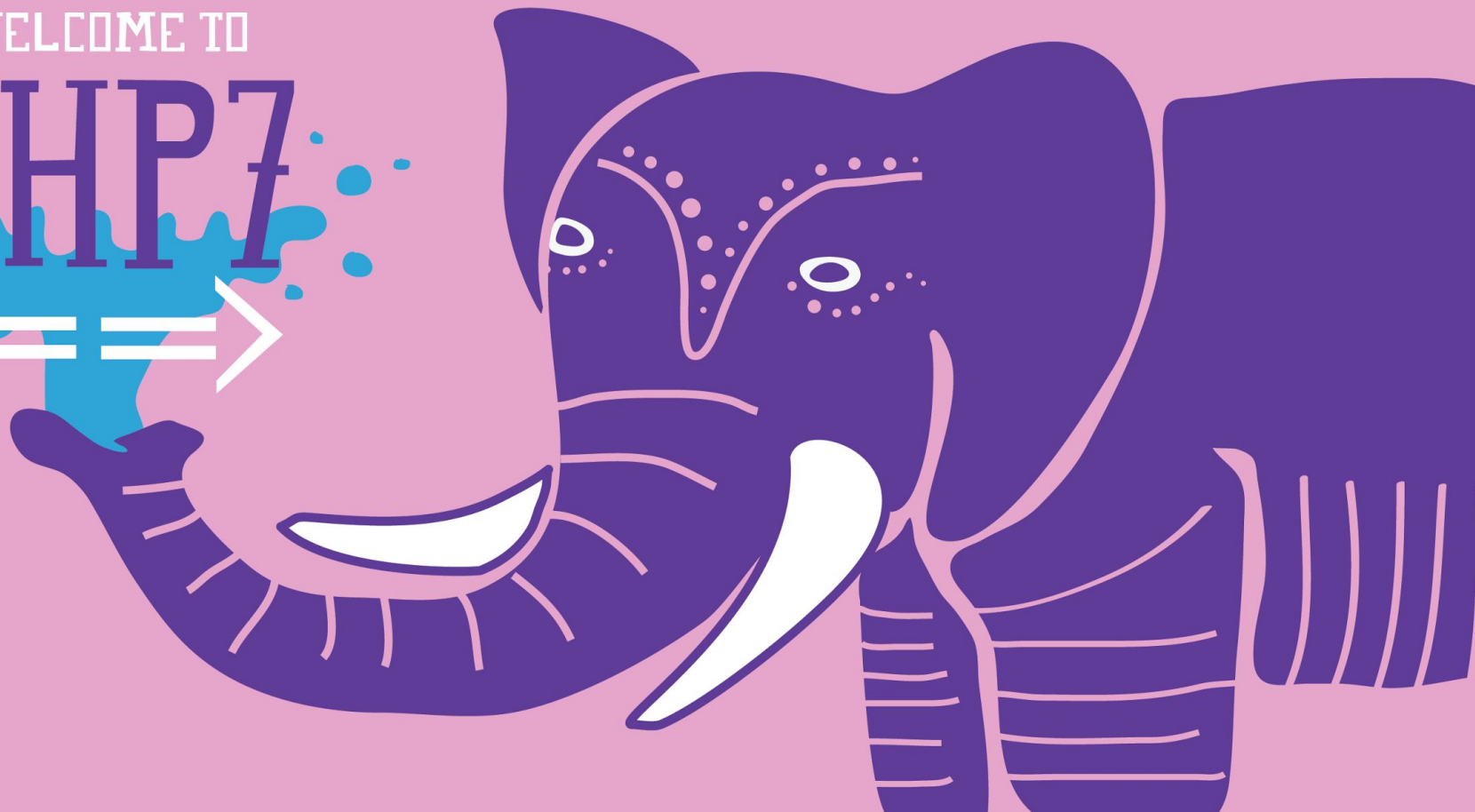
- PHP: ***Hypertext Preprocessor***.
- É uma linguagem de script *open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web.
- Pode ser embutida dentro do HTML.

Um pouco de história



WELCOME TO

PHP7



Por que o PHP?



- O PHP é gratuito;
- Os custos de manutenção de um servidor são muito reduzidos;
- Atualizações consistentes;
- Se integra a quase todos os Databases;
- Muito fácil de aprender;
- Grande quantidade de ambientes de desenvolvimento profissionais disponíveis;
- Um grande banco de classes e funções, prontas para uso.

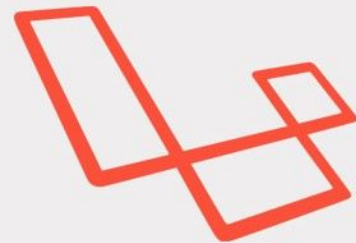
Por que o PHP?



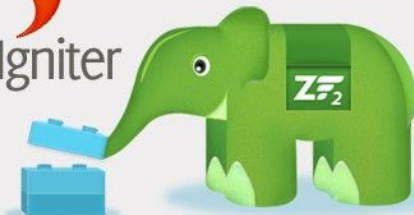
Frameworks



Symfony



Code Igniter



Por que o PHP?

CMS mais usados no mercado.



```
<?php
```

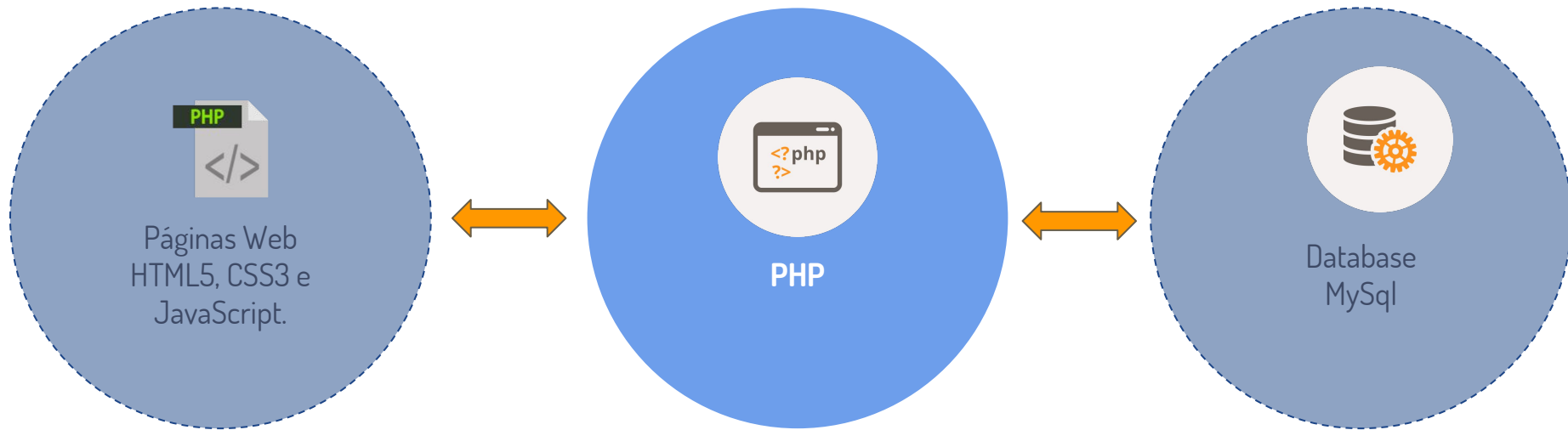
```
echo "Let's go!"
```

```
?>
```



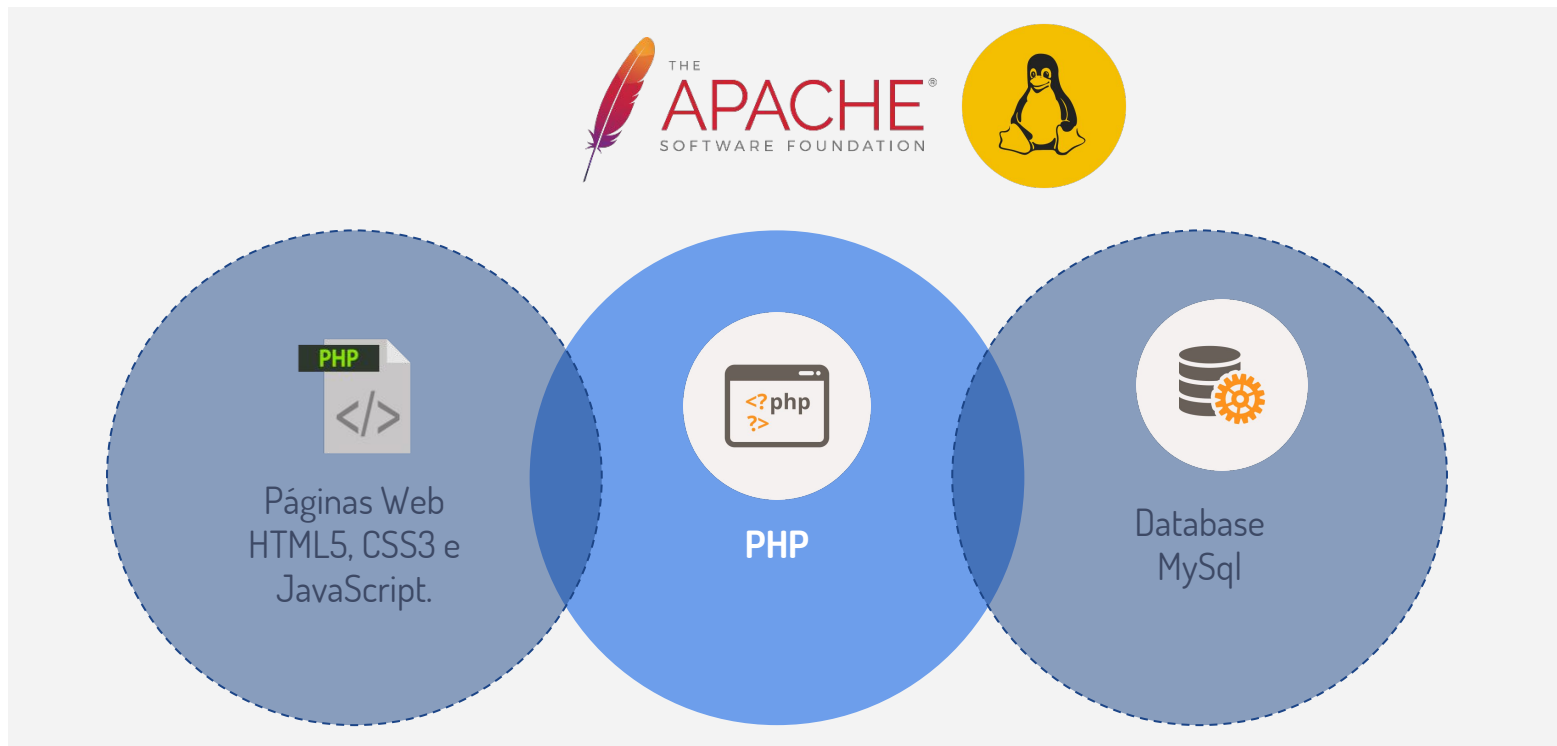


Arquitetura Web utilizando



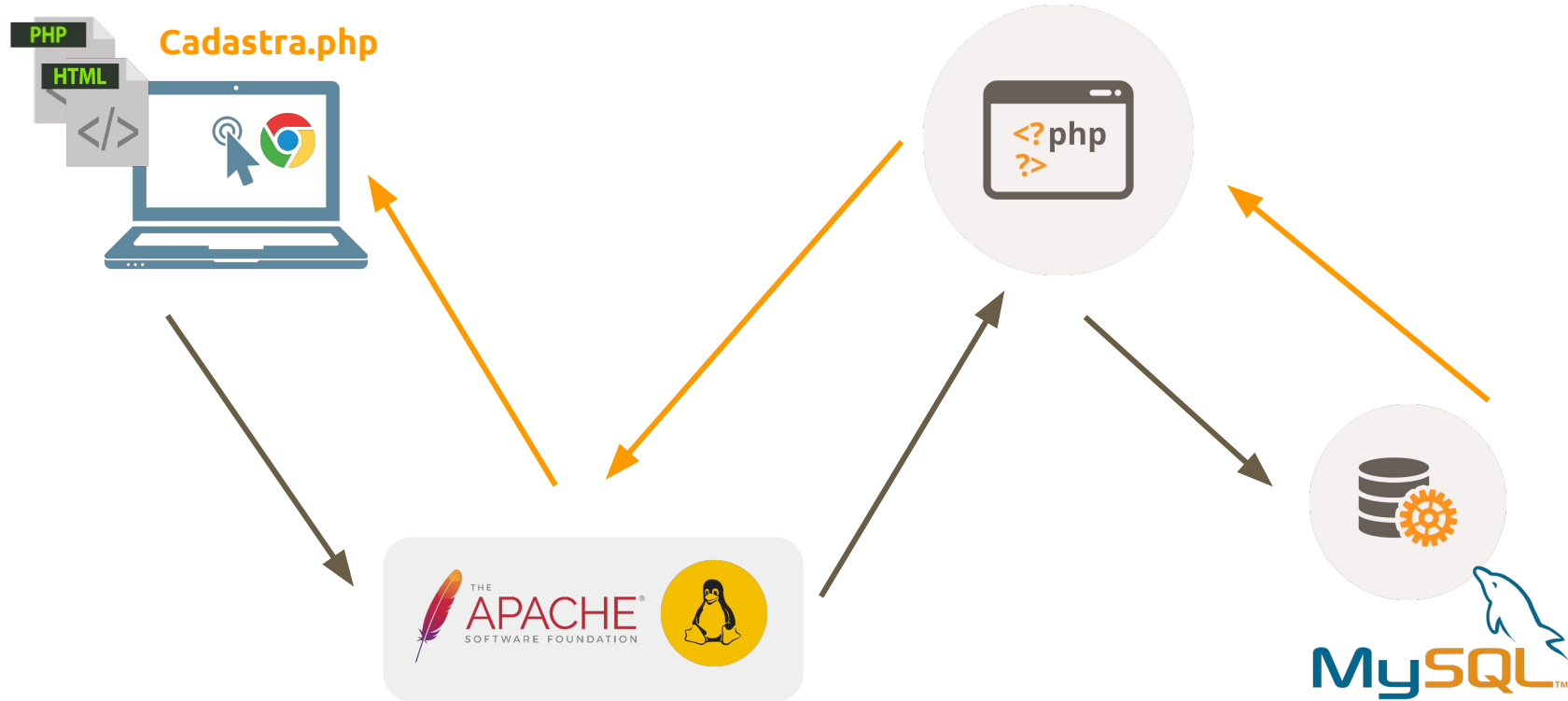


Arquitetura Web utilizando

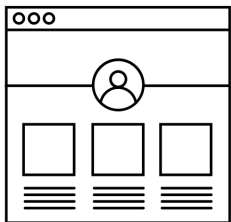




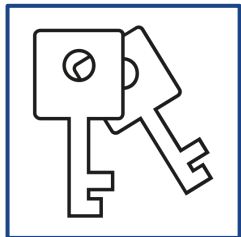
Arquitetura Web utilizando



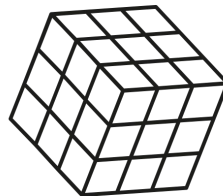
Básico de um sistema



1. Páginas públicas.



2. Páginas privadas.



3. Conexão com um banco de dados.



Por onde começar ?



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. Tudo que está fora dessas tags é ignorado pelo interpretador do PHP.

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

// ... mais código

echo "última instrução";

// o script termina aqui, sem tag de fechamento PHP
```



Tags PHP



O PHP também permite a tag curta `<? , mas essa opção pode não está disponível quando não 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.`

`<?`

`echo "Uma linha de Impressão na tela";`

`?>`



Tags PHP

Se um arquivo for **código PHP puro**, é preferível **omitir a tag de fechamento no final do arquivo**.

Isso irá prevenir a existência de espaços ou linhas em branco após a tag, que podem causar efeitos indesejáveis.



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

// ... mais código

echo "última instrução";

// o script termina aqui, sem tag de fechamento PHP
```



```
1 <?php
2
3 echo "esse arquivo só tem código PHP"
4
5 ?>
```



Escapando o HTML

Tudo o que estiver fora das tags PHP é ignorado pelo interpretador.

Isso permite **arquivos PHP de conteúdo misto**.

Permite que **o PHP seja incluído dentro de documentos HTML**, para, por exemplo, **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>
```

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




Separação de instruções

As instruções PHP devem ser terminadas com um **ponto-e-vírgula ;** ao final de cada comando.

Se o comando for único, você não precisa incluir o ponto-e-vírgula.

```
<?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).

```
<?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
?>
```



O PHP suporta tipos primitivos

Escalares:

- boolean
- integer
- float
- string

Compostos:

- array
- object

Especiais:

- resource
- NULL

```
<?php
$a_bool = TRUE;    // um booleano
$a_str  = "foo";   // uma string
$a_str2 = 'foo';   // uma string
$an_int = 12;      // um inteiro

echo gettype($a_bool); // mostra: boolean
echo gettype($a_str);  // mostra: string

// Se ele é um inteiro, incrementa-o com quatro
if (is_int($an_int)) {
    $an_int += 4;
}
```



Variáveis

As variáveis no PHP são representadas por um **cifrão (\$)** seguido pelo nome da variável. Os nomes de variáveis são case-sensitive.

```
<?php
$var = 'Bob';
$Var = 'Joe';
echo "$var, $Var";           // exibe "Bob, Joe"

$4site = 'not yet';         // inválido; começa com um número
$_4site = 'not yet';        // válido; começa com um sublinhado
$täyte = 'mansikka';        // válido; 'ä' é um caracter ASCII (extendido)

228

?>
```



Variáveis Pré-definidas

As variáveis representam tudo: de **variáveis externas à variáveis nativas de ambiente**, última mensagem de erro ao último cabeçalho recebido.

- Superglobais — Superglobais são variáveis nativas que estão sempre disponíveis em todos escopos
- \$GLOBALS — Referencia todas variáveis disponíveis no escopo global
- \$_SERVER — Informação do servidor e ambiente de execução
- \$_GET — HTTP GET variáveis
- \$_POST — HTTP POST variables
- \$_FILES — HTTP File Upload variáveis
- \$_REQUEST — Variáveis de requisição HTTP
- \$_SESSION — Variáveis de sessão
- \$_ENV — Environment variables
- \$_COOKIE — HTTP Cookies
- \$php_errormsg — A mensagem de erro anterior
- \$HTTP_RAW_POST_DATA — Informação não-tratada do POST
- \$http_response_header — Cabeçalhos de resposta HTTP
- \$argc — O número de argumentos passados para o script
- \$argv — Array de argumentos passados para o script



Constantes

Você pode definir uma constante utilizando a função **define()**, ou a palavra-chave **const** fora da definição de uma classe a partir do PHP 5.3.0.

Uma vez definida, a constante **não poderá ser modificada ou anulada**.

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // imprime "Hello world."
echo Constant; // imprime "Constant" e gera um alerta
notice.
?>
```

```
<?php
// Funciona a partir do PHP 5.3.0
const CONSTANT = 'Hello World';

echo CONSTANT;

// Funciona a partir do PHP 5.6.0
const ANOTHER_CONST = CONSTANT.'; Goodbye World';
echo ANOTHER_CONST;
```



Estruturas de Controle

While

são os mais simples
tipos de laços do PHP

```
<?php
/* example 1 */

$i = 1;
while ($i <= 10) {
    echo $i++; /* the printed value would be
                $i before the increment
                (post-increment) */
}

/* example 2 */

$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
?>
```



Estruturas de Controle

for

Um pouco mais complicado que o while.

```
<?php
/* exemplo 1 */

for ($i = 1; $i <= 10; $i++) {
    echo $i;
}

/* exemplo 2 2 */

for ($i = 1; ; $i++) {
    if ($i > 10) {
        break;
    }
    echo $i;
}
```




Estruturas de Controle

Foreach: fornece uma maneira fácil de iterar sobre arrays.

Funciona somente em arrays e objetos.

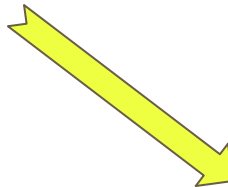
```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
// $arr is now array(2, 4, 6, 8)
```



Incluindo conteúdo

Há quatro funções básicas:

- include “meu-arquivo.php”;
- include_once “meu-arquivo.php”;
- require “meu-arquivo.php”;
- require_once “meu-arquivo.php”;





Incluindo conteúdo

Include

A declaração include inclui e avalia o arquivo informado. Se o arquivo não for encontrado no `include_path`, a declaração include checará no diretório do script que o executa e no diretório de trabalho corrente, antes de falhar.

```
include 'file.txt'; // Works.  
include 'file.php'; // Works.
```



Incluindo conteúdo

include_once

Avalia o arquivo informado durante a execução do script. Este é um comportamento similar a declaração include, com a única diferença que, se o código do arquivo já foi incluído, não o fará novamente, e o include_once retornará TRUE.

```
<?php
include_once "a.php"; // this will include a.php
include_once "A.php"; // this will include a.php again! (PHP 4 only)
?>
```



Incluindo conteúdo

require

É idêntica à include, exceto que em caso de falha também produzirá um erro fatal de nível E_COMPILE_ERROR. Em outras palavras, **ele parará** o script enquanto que o include apenas emitirá um alerta (**E_WARNING**) permitindo que o script continue.

```
<?php  
require('somefile.php');  
?>
```



Incluindo conteúdo

require_once

É idêntica à *require*, exceto que o PHP verificará se o arquivo já foi incluído, e em caso afirmativo, não o incluirá (exigirá) novamente.

```
<?php
define('__ROOT__', dirname(dirname(__FILE__)));
require_once(__ROOT__.'/config.php');
?>
```



Incluindo conteúdo

Um array no PHP é na verdade um mapa ordenado.
Um mapa é um tipo que relaciona valores a chaves.

```
<?php
$array = array("foo", "bar", "hello", "world");
var_dump($array);
?>
```



Arrays

Um array no PHP é na verdade um mapa ordenado.
Um mapa é um tipo que relaciona valores a chaves.

```
array(  
    chave => valor,  
    chave2 => valor2,  
    chave3 => valor3,  
    ...  
)
```

```
<?php  
$array = array(  
    "foo" => "bar",  
    "bar" => "foo",  
);  
  
// a partir do PHP 5.4  
$array = [  
    "foo" => "bar",  
    "bar" => "foo",  
];  
?>
```




Arrays

Um array no PHP é na verdade um mapa ordenado.
Um mapa é um tipo que relaciona valores a chaves.

```
<?php
$array = array("foo", "bar", "hello", "world");
var_dump($array);
?>
```

Fonte? Vá na fonte!

php

Downloads

Documentation

Get Involved

Help

[PHP 7.1.0 Beta 2 Released](#)

Documentation

The PHP Manual is available online in a selection of languages. Please pick a language from the list below.

More information about php.net URL shortcuts by visiting our [URL howto page](#).

Note, that many languages are just under translation, and the untranslated parts are still in English. Also some translated parts might be outdated. The translation teams are open to contributions.

Documentation for PHP 4 has been removed from the manual, but there is archived version still available. For more informations, please read [Documentation for PHP 4](#).

Formats	Destinations
View Online	English , Brazilian Portuguese , Chinese (Simplified) , French , German , Japanese , Korean , Romanian , Russian , Spanish , Turkish
Downloads	For downloadable formats, please visit our documentation downloads page.