

Programa de Formação Web Developers PHP

Agosto/2021



CONSTRUSITE BRASIL
O SEU SUCESSO ONLINE

GALAXPAY 
SEU NEGÓCIO RECORRENTE



01

Conhecendo o PHP

O que é o PHP e quando utilizar?
Boas práticas, sintaxe.

02

Sintaxe + Variáveis

Variáveis, tipagem de dados e constantes. Escrevendo os primeiros códigos em PHP.

03

PSRs

Padrões de projeto

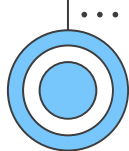
04

Operadores

Incríveis dicas, e utilidades do inspecionar elemento



Introdução



Conhecendo o PHP

O PHP é uma linguagem de programação interpretada.

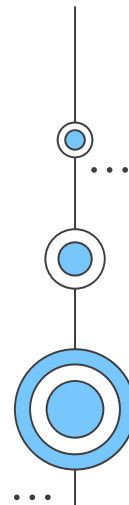
Amplamente utilizado em aplicações web, é uma das tecnologias mais utilizadas na internet.

É uma linguagem fácil de aprender e executar, e possui uma grande comunidade de desenvolvedores para ajudar a desenvolver e difundir a tecnologia.

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

Lá você irá encontrar a documentação em vários idiomas e informações sobre as versões lançadas.

A última versão estável até o momento é o PHP 8.0.



Conhecendo o PHP

- Linguagem madura, 23 anos de existência
- Surgiu em 1993 com a necessidade de um programador chamado **Rasmus Lerdorf** implementar um script para facilitar o controle de seu currículo on-line, controlas view, acessos ...
- Tais scripts foram nomeados de '**Personal Home Page Tools**' e '**Personal Home Page Forms Interpreter**' (PHP / FI)
- Conforme necessidade, Rasmus escreveu uma implementação em C com mais funções, conexões com banco e possibilita aos usuários desenvolverem aplicativos dinâmicos para a Web



Conhecendo o PHP



- Com os avanços, Rasmus liberou código para a comunidade, tornando a linguagem open source, e proporcionando crescimentos incríveis.
- Grande virada de chave, foi quando dois universitários Andi Gutmans e Zeev Surasky , viram potencial na linguagem para trabalhos acadêmicos, e junto com Rasmus , reescrevendo totalmente o código do PHP.
- Criaram uma nova engine (motor ou núcleo da linguagem) chamada 'Zend Engine'

Esta modificação no código proporcionou melhoramentos de performance e adicionou suporte para muitos servidores Web, sessões HTTP, buffer de saída, mais segurança ao manipular dados de entrada e maneiras mais seguras de manipular input de usuários

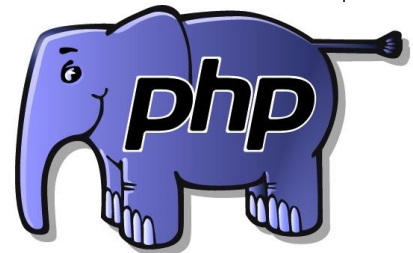
Por que PHP é tão bom?

- O PHP é uma linguagem flexível, que permite aplicação de paradigmas tanto procedural quanto orientado a objeto (O.O)
- Vasta comunidade de desenvolvedores ao redor do mundo.
- Documentação incrível.
- Suporte a extensões, e vários banco de dados.
- O interpretador pode ser instalado em máquinas linux, ou windows, oq ajudou na popularidade.
- PHP 8 e sua grandeza.



Por que PHP é tão bom?

- O PHP é uma linguagem flexível, que permite aplicação de paradigmas tanto procedural quanto orientado a objeto (O.O)
- Vasta comunidade de desenvolvedores ao redor do mundo.
- Documentação incrível.
- Suporte a extensões, e vários banco de dados.
- O interpretador pode ser instalado em máquinas linux, ou windows, popularidade.



Por que PHP é tão bom?

- Mais maduro e completo ecossistema para gerenciamento de dependências (com Composer e Packagist).
- É a linguagem mais utilizada para web, cerca de ~82% de todos os sites a utilizam:

	2016 1 Apr	2016 1 May	2016 1 Jun	2016 1 Jul	2016 1 Aug	2016 1 Sep	2016 1 Oct	2016 1 Nov	2016 1 Dec	2017 1 Jan	2017 1 Feb	2017 1 Mar	2017 1 Apr	2017 3 Apr
PHP	82.1%	82.3%	82.1%	82.1%	82.1%	82.0%	81.9%	82.2%	82.3%	82.4%	82.4%	82.5%	82.6%	82.6%
ASP.NET	15.7%	15.7%	15.8%	15.8%	15.8%	15.7%	15.7%	15.6%	15.4%	15.3%	15.3%	15.2%	15.1%	15.1%
Java	2.9%	2.7%	2.7%	2.7%	2.7%	2.8%	2.9%	2.7%	2.7%	2.7%	2.7%	2.7%	2.6%	2.6%
static files	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%
ColdFusion	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.6%	0.6%	0.6%
Ruby	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%
Perl	0.5%	0.5%	0.5%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%
JavaScript	0.2%	0.2%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%
Python	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%
Erlang	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Miva Script	0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%	<0.1%

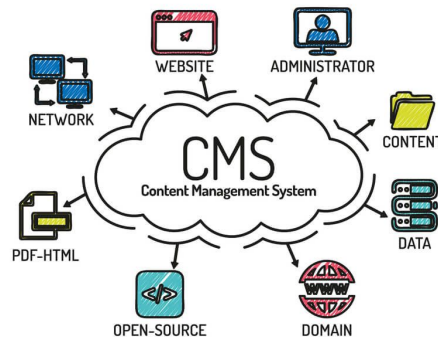
Fonte: <https://w3techs.com>



Por que PHP é tão bom?

- É utilizado pelas principais empresas de tecnologia do mercado e em grandes projetos. Algumas das grandes empresas que usam o PHP: Facebook, Google, Baidu, Wikipedia, Spotify, Uber, Wordpress (CMS mais usado no mundo) dentre outros.
- O PHP possui excelentes e estabelecidos frameworks.

Laravel
Symfony
Zend Framework
CakePHP
Silex
CodeIgniter





PSRs

“PHP e a terra
sem lei ?”



PSRs



- Site Oficial: <https://www.php-fig.org/>
- Recomendações de padrões PHP
- PHP FIG (Framework Interoperability Group)
- Avançar o ecossistema PHP e promover bons padrões



PHP-FIG

Boas práticas: PSRs

The Standards

ID	Name	Description
PSR-0	Autoloading Standard	Mandatory requirements that must be adhered to for autoloader interoperability
PSR-1	Basic Coding Standard	Standard coding elements that are required to ensure a high level of technical interoperability between shared PHP code.
PSR-2	Coding Style Guide	Extends and expands on PSR-1.
PSR-3	Logger Interface	Common interface for logging libraries.
PSR-4	Autoloading Standard	Specification for autoloading classes from file paths.
PSR-6	Caching Interface	Create cache-aware libraries that can be integrated into existing frameworks and systems without the need for custom development.
PSR-7	HTTP Message Interface	Common interfaces for representing HTTP messages.

Sintaxe

Todos os comandos PHP devem ser escritos após a tag de abertura

`<?php` e a tag de fechamento `?>`

Seu código só irá executar caso o arquivo tenha extensão `.php`

Atenção: a máquina precisa ter instalado todas as tecnologias necessárias para servir aplicações em PHP.

Para configurar seu ambiente de desenvolvimento, existem diferentes ferramentas dependendo do sistema operacional.

No Windows, use o WAMP Server:

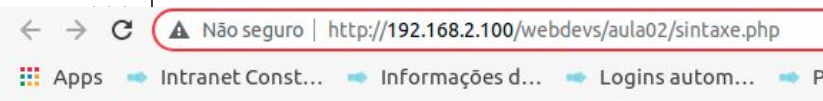
<https://www.wampserver.com/en/>

No Ubuntu, você pode seguir o tutorial oficial:

<https://ubuntu.com/server/docs/programming-php>

Sintaxe

Exemplo clássico:



 **Dica de boas práticas:**

Não use espaços ou caracteres especiais nos nomes dos arquivos!

Ao invés dos espaços, prefira:

- underline (_)
- hífen (-)
- camelCase

Atividades

1. Faça um script php que print o incrível: "Hello world" =D



> Hello, world!_

Variáveis

Todas as variáveis em PHP são definidas como:

`$ + nome da variável`

Exemplo:

- `$password`
- `$_name`
- `$idUser`

Um nome válido de variável deve começar com letra ou underline, seguido pela quantidade de letras, números ou underlines que você quiser.

⚠ Atenção

`$this` é um nome de variável reservado! Você não pode criar variáveis com esse nome.

Tipos de dados

O PHP trabalha com os tipos mais comuns de dados entre as linguagens de programação:

- String: textos e caracteres;
- Boolean: `true` e `false`;
- Array: conjuntos de valores;
- Integer: números inteiros;
- Double: números com casas decimais;
- Object: classes (orientação a objetos).

Porém, o PHP é uma linguagem **fracamente tipada**.

Isso significa que uma variável que foi definida contendo uma string, pode receber um valor booleano mais tarde no código sem gerar erros.

Tipos de dados

Exemplo:

tipos_dados.php

webdevs > aula02 > tipos_dados.php > ...

```
1  <?php
2
3  $variavel = 'algum texto aleatório';
4  echo $variavel;
5
6  echo '<br>';
7  echo '<br>';
8
9  $variavel = 65;
10 echo $variavel;
11
```

← → ↻ ⚠ Não seguro |

Apps ➡ Intranet Const...

algum texto aleatório

65

Constantes

Uma constante é um valor que não pode ser alterado depois que foi inicializado.

Diferente das variáveis, não se utiliza \$ quando definimos/acessamos uma constante.

Porém, assim como as variáveis (e outros rótulos do PHP), o nome deve iniciar com letra ou underline, seguido de letras, números ou underlines.

Para definir uma constante, você deve sempre usar o método `define()` do PHP.

Depois de definida, você pode acessar a constante em qualquer lugar do código escrevendo apenas o nome dela.

Constantes

Exemplo:

constantes.php x

webdevs > aula02 > constantes.php > ...

```
1 <?php
2
3 define('NOME_BANCO', 'meubancodedados');
4
5 echo NOME_BANCO;
6
```

← → ↻ ⚠ Não seguro | 192.168.1.1

Apps Intranet Const... php

meubancodedados

Operadores

Concatenação

...
.: Concatena duas strings,
recomendado nas versões mais
recentes do PHP

...
.=: Concatena a string à direita à
string à esquerda

! **Caso especial:**

+: Também era usado para
concatenar string em versões
antigas do PHP. Hoje só funciona
como operador de soma numérica

Concatenação

Exemplo:

concatenacao.php

webdevs > aula02 > concatenacao.php > ...

```
1 <?php
2
3 $string = 'think ' . 'outside ' . 'the box';
4 echo $string;
5
6 echo '<br>';
7
8 $string .= ' >> aprimoramento';
9 echo $string;
10
```



Não seguro | 192.168.2.100/v



Apps



Intranet Const...



PHP Sanc

think outside the box

think outside the box >> aprimoramento

Atividades

1. Crie variáveis para armazenar seu nome e sua idade, e depois imprima na tela a mensagem: "Meu nome é Fulano e tenho 999 anos".
2. Escreva uma função que receba o nome do usuário logado e mostre uma mensagem de boas-vindas aos usuários do sistema Galax Imob. Você deve usar pelo menos uma variável e uma constante no script.

Operadores matemáticos

...

+: Soma

-: Subtração

/: Divisão

*****: Multiplicação

%: Resto da divisão

++: Incrementa em 1 o valor da variável

--: Decrementa em 1 o valor da variável


+=: Soma a variável à esquerda ao valor da direita

-=: Subtrai o valor à direita da variável à esquerda

***=**: Multiplica a variável pelo valor informado

/=: Divide a variável pelo valor informado

%=: Atribui à variável o resultado de sua divisão pelo valor à direita




Atividades

Um cliente fez a compra dos seguintes itens em uma loja virtual:

- Bolsa feminina: R\$89,90
- Carteira de viagem: R\$49,00
- Óculos de sol: R\$250,00

1. Escreva um script para calcular o valor total da compra.
2. Qual o valor médio que esse cliente gastou nessa compra?
3. Supondo que o cliente tenha usado um cupom de 10% de desconto, qual seria o valor final da compra nesse caso?

Operadores condicionais

...

<: Menor que

>: Maior que

...
<=: Menor ou igual

>=: Maior ou igual

==: Compara se dois valores são iguais

===: Compara se os valores são idênticos (inclusive a tipagem!)

!=: Compara se os valores são diferentes

<>: Equivalente a != (menos utilizado)

== VS. ===

O operador == compara se dois valores são equivalentes, independente do tipo de dado.

Exemplo:

```
'13' == 13
19.0 == 19
0 == false
1 == true
'meu nome' == true
'0' == false
```

Por outro lado, o === compara se os valores são **idênticos**.

Quer dizer, a comparação vai retornar falso se os tipos de dados forem diferentes.

Veja:

```
0 !== false
'169' !== 169
'uma string' !== true
```

Atividades

Considere as idades das seguintes pessoas:

- Jaqueline: 28
- Bruno: 29
- Fernanda: 18
- Rafael: 21
- Luna: 32

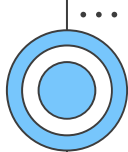
1. Crie um script para calcular a média de idade do grupo.
2. Quais dessas pessoas estão abaixo da média de idade? E quais estão acima?

Operadores lógicos

!: Nega a expressão informada
(inversão lógica)

&&: Retorna verdadeiro se as
condições à esquerda e à direita
forem verdadeiras

||: Retorna verdadeiro se pelo menos
uma das condições for verdadeira



Operadores lógicos

Exemplo:

operadores_logicos.php x

webdevs > aula02 > operadores_logicos.php

```
1 <?php
2
3 var_dump(true || false);
4
5 var_dump(true && false);
6
7 var_dump(!true);
8
```

← → ↻ ⚠ Não seguro | 192.168.2.100/webdevs/aula02/operadores_logicos.php

Apps → Intranet Const... PHP Sandbox, ... Gaddafipsum https://www..

/var/www/html/webdevs/aula02/operadores_logicos.php:3:boolean true

/var/www/html/webdevs/aula02/operadores_logicos.php:5:boolean false

/var/www/html/webdevs/aula02/operadores_logicos.php:7:boolean false

Atividades

1. Crie um script que receba um número, caso ele seja diferente de 10, print a string: "O número XX é diferente de 10;
2. Crie um script que receba dois números informados pelo usuário, se ambos números forem par, printe a msg;
Os números informados XX, XX são pares.
3. Faça um script que receba 5 números, imprima a soma, e caso a soma for maior que a quantidade de números informados, printe uma string informado isto.



Atividades

1. Faça um script que calcule a área de um quadrado, em seguida mostre o dobro desta área para o usuário.
2. Faça um script que peça a temperatura em graus Fahrenheit, transforme e mostre a temperatura em graus Celsius.
$$C = (5 * (F - 32) / 9)$$
3. Faça um script que peça o tamanho de um arquivo para download (em MB) e a velocidade de um link de Internet (em Mbps), calcule e informe o tempo aproximado de download do arquivo usando este link (em minutos).



Dúvidas?

