

Programando em PHP

Introdução ao PHP

PSR

Criação de API com Laravel

Quem sou eu

Luiz Amaral

Desenvolvedor PHP e JS

Github: <https://github.com/amaralluiz>

Linkedin: <https://www.linkedin.com/in/amaralluiz/>

Blog: <https://www.amaralluiz.dev/>

Email: contato@amaralluiz.dev

Variáveis

Tipos

Inteiro

Ponto Flutuante

String

Array

Objeto

Null

Booleano

Operadores

Operadores

Atribuição

Aritmético

Relacionais

Lógicos

Estruturas de Controle

Estruturas de Controle

Estruturas de Seleção

Estruturas de Repetição

Estruturas de Seleção

If / else if / else

```
if(expressão1) {  
    // comandos se a expressão é verdadeira  
} else if(expressão2) {  
    // comandos se a expressão2 for verdadeira  
} else {  
    // comandos se expressão 1 e 2 forem  
    falsas  
}
```

Estruturas de Seleção

Switch

```
<?php
switch($variavel) {
    case valor1:
        // comandos
        break;
    case valor2:
        // comandos
        break;
    default:
        // comandos
}
```

Estruturas de Repetição

For

```
<?php
for(expr1; expr2; expr3) {
    // comandos
}
```

```
<?php
for($i = 0; $i <= 10; $i++) {
    echo $i .PHP_EOL;
}
```

Estruturas de Repetição

While

```
<?php
while(expr1) {
    // comandos
}
```

```
<?php
$i = 0;
while($i <= 10) {
    echo $i . PHP_EOL;
    $i++;
}
```

Manipulação de Arrays

Criando array

```
<?php
```

```
// Primeira forma, utilizando array()
```

```
$scores = array('vermelho', 'azul', 'verde');
```

```
// Segunda forma, utilizando sintaxe resumida
```

```
$scores = ['vermelho', 'azul', 'verde'];
```

```
// Terceira forma, adicionando valores
```

```
$scores[] = 'vermelho';
```

```
$scores[] = 'azul';
```

```
$scores[] = 'verde';
```

Acessando arrays

```
<?php  
echo $cores[0]; // resultado = vermelho  
echo $cores[1]; // resultado = azul  
echo $cores[2]; // resultado = verde
```

Arrays associativos

```
<?php
```

```
// Utilizando array()
```

```
$scores = array(  
    'vermelho' => 'FF0000',  
    'azul' => '0000FF',  
    'verde' => '00FF00',  
);
```

```
// Utilizando sintaxe resumida e adicionando  
valores
```

```
$scores = [];  
$scores['vermelho'] = 'FF0000';  
$scores['azul'] = '0000FF';  
$scores['verde'] = '00FF00';
```


Arrays associativos

```
<?php
```

```
echo $cores['vermelho']; // resultado FF0000
```

```
echo $cores['azul']; // resultado '0000FF'
```

```
echo $cores['verde']; // resultado '00FF00'
```

Iterações

```
<?php
```

```
$frutas = array();
```

```
$frutas['cor'] = 'vermelho';
```

```
$frutas['sabor'] = 'doce';
```

```
$frutas['formato'] = 'redondo';
```

```
$frutas['nome'] = 'maçã';
```

```
foreach($frutas as $chave => $valor) {  
    echo "$chave => $valor \n";  
}
```

Arrays

Multidimensionais

```
$carros = array(
    'Palio' => array(
        'cor' => 'azul',
        'potencia' => '1.0',
        'opcionais' => 'Ar Cond.'
    ),
    'Corsa' => array(
        'cor' => 'cinza',
        'potencia' => '1.3',
        'opcionais' => 'mp3'
    ),
    'Gol' => array(
        'cor' => 'branco',
        'potencia' => '1.0',
        'opcionais' => 'metalica'
    )
);

echo $carros['Palio']['opcionais'];
```

Arrays multidimensionais

```
$carros = [];  
$carros['Palio']['cor'] = 'azul';  
$carros['Palio']['potencia'] = '1.0';  
$carros['Palio']['opcionais'] = 'Ar Cond.';  
$carros['Corsa']['cor'] = 'cinza';  
$carros['Corsa']['potencia'] = '1.3';  
$carros['Corsa']['opcionais'] = 'mp3';  
$carros['Gol']['potencia'] = 'branco';  
$carros['Gol']['cor'] = '1.0';  
$carros['Gol']['opcionais'] = 'metalica';  
  
echo $carros['Palio']['opcionais'];
```

Iteração em Arrays multidimensionais

```
<?php
```

```
$carros = [];  
$carros['Palio']['cor'] = 'azul';  
$carros['Palio']['potencia'] = '1.0';  
$carros['Palio']['opcionais'] = 'Ar Cond.';  
$carros['Corsa']['cor'] = 'cinza';  
$carros['Corsa']['potencia'] = '1.3';  
$carros['Corsa']['opcionais'] = 'mp3';  
$carros['Gol']['potencia'] = 'branco';  
$carros['Gol']['cor'] = '1.0';  
$carros['Gol']['opcionais'] = 'metalica';  
  
foreach ($carros as $modelo =>  
$caracteristicas) {  
    echo "> modelo $modelo\n";  
    foreach ($caracteristicas as  
$caracteristica => $valor) {  
        echo " - caracteristica  
$caracteristica => $valor\n";  
    }  
}
```

Funções para manipulação de Arrays

array_merge

```
<?php
$a = ['verde', 'azul'];
$b = ['vermelho', 'amarel'];
$c = array_merge($a, $b);
var_dump($c);
```

array_push

```
<?php
$a = ['verde', 'azul'];
$b = ['vermelho', 'amarelo'];
array_push($a, $b);
var_dump($a);
```


Implode e explode

```
<?php
```

```
// Explode
```

```
$string = "10/05/2015";
```

```
var_dump(explode("/", $string));
```

```
// Implode
```

```
$array = [10, 5, 2015];
```

```
var_dump(implode('/', $array));
```

Funções

Funções

```
<?php
```

```
// criação
```

```
function nomeDaFuncao (arg1, arg2, arg3)
{
    $valor = $arg1 + $arg2 + $arg3;
    return $valor;
}
```

```
// chamada
```

```
nomeDaFuncao(arg1, arg2, arg3);
```



PSR

Criação de API com Laravel

Protocolo HTTP

HTTP é um protocolo que permite a obtenção de recursos, como documentos HTML. É a base de qualquer troca de dados na Web e um protocolo cliente-servidor, o que significa que as requisições são iniciadas pelo destinatário, geralmente um navegador da Web.

O que é uma API?

API, do inglês Application Programming Interface, é um conjunto de rotinas e padrões estabelecidos por um software para a utilização de suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implantação do software, mas apenas usar seus serviços.

API Web é o conjunto de interfaces no contexto de desenvolvimento Web. É um conjunto definido de mensagens de requisição e resposta HTTP, geralmente nos formatos XML ou JSON. Atualmente a arquitetura mais utilizada é o modelo REST.

Mas o que é REST?

REST é um estilo de arquitetura de software que define um conjunto de restrições a serem usadas para a criação de web services. Foi criada em 2000 por Roy Fielding como uma tese de doutorado.

Princípios:

- Um *protocolo cliente/servidor sem estado*: cada mensagem HTTP contém toda a informação necessária para compreender o pedido. Como resultado, nem o cliente e nem o servidor necessitam gravar nenhum estado das comunicações entre mensagens. Na prática, muitas aplicações baseadas em HTTP utilizam cookies e outros mecanismos para manter o estado da sessão
- Um conjunto de *operações bem definidas* que se aplicam a todos os *recursos* de informação: HTTP em si define um pequeno conjunto de operações, as mais importantes são **POST**, **GET**, **PUT** e **DELETE**. Com frequência estas operações são combinadas com operações CRUD para a persistência de dados, onde POST não se encaixa exatamente neste esquema.
- Uma *sintaxe universal* para identificar os recursos. No sistema REST, cada recurso é unicamente direcionado através da sua URI.
- O *uso de hipermídia*, tanto para a informação da aplicação como para as transições de estado da aplicação: a representação deste estado em um sistema REST são tipicamente HTML ou XML. Como resultado disto, é possível navegar com um recurso REST a muitos outros, simplesmente seguindo ligações sem requerer o uso de registros ou outra infraestrutura adicional.