

PHP

Hypertext Preprocessor

Programmer's - Beyond IT

Flávio Medeiros

Marcos Couto



O?QUE É PHP?

- PHP é um acrônimo recursivo para PHP: Hypertext Preprocessor
- Linguagem server-side interpretada para desenvolvimento web
- Começou a ser desenvolvida em 1995 por Rasmus Lerdorf
- Atualmente se encontra na versão 7.3.3 (a versão 7.4 está em desenvolvimento)

PARADIGMA DO PHP

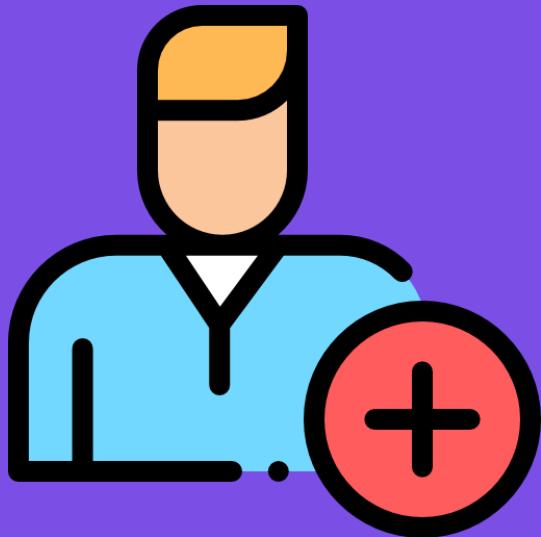
PHP é multi-paradigma, uma linguagem flexível.

- Procedural
- Orientada à Objetos
- Funcional



PHP É "TIPADO"?

A partir da versão 7 o PHP permite tipagem dos dados.



Fracamente tipada

CRIANDO UM ARQUIVO PHP



```
<?php
```

```
...
```

```
?>
```

CRIANDO VARIÁVEIS

```
1 <?php
2
3     $integer = 10;
4
5     $floatOrDouble = 10.5;
6
7     $boolean = true;
8
9     $string = 'Hello';
10
11    $stringInterpolation = "{$string} World!";
12
13    $array = array(
14        'key1' => 'value1',
15        'key2' => 'value2'
16    );
17
18    $array2 = [
19        'key1' => 'value1',
20        'key2' => 'value2',
21    ];
22
23    $object = new stdClass();
24
25 ?>
```

CRIANDO FUNÇÕES

```
1 <?php
2
3     function soma($a, $b)
4     {
5
6         return $a + $b;
7
8     }
9
10
11    echo soma(10, 30); // resulta em 40
12
13 ?>
```

CONDIÇÕES

```
1 <?php
2
3 function verificaMaior($a, $b)
4 {
5
6     $resultado;
7
8     if ($a > $b) {
9         $resultado = 'A é maior que B';
10    }
11    elseif ($a < $b) {
12        $resultado = 'B é maior que A';
13    }
14    elseif ($a == $b) {
15        $resultado = 'As variaveis A e B são iguais';
16    }
17 }
18
19 return $resultado;
20 }
21
22 echo verificaMaior(30, 20);
23 // A é maior que B
24 echo verificaMaior(10, 20);
25 // B é maior que A
26 echo verificaMaior(50, 50);
27 // As variaveis A e B são iguais
28
```

OPERADORES TERNÁRIOS

```
1 $a = true;
2 echo ($a) ? 'A condição é verdadeira' : 'A condição é falsa';
3 //A condicao é verdadeira
4
5 $names = [
6     0 => 'Programmers',
7     1 => '',
8 ];
9 echo ($names[0]) ?: 'Padrão';
10 //Programmers
11
12 echo ($names[1]) ?: 'Padrão';
13 //Padrão
14
15 echo ($names[2]) ?: 'Padrão';
16 //Gera erro
17
18 echo ($names[0]) ?? 'Padrão';
19 //Programmers
20
21 echo ($names[1]) ?? 'Padrão';
22 //
23
24 echo ($names[2]) ?? 'Padrão';
25 //Padrão
```

CRIANDO UMA CLASSE

```
1 <?php
2
3 class Person {
4
5     protected $id;
6     public $name;
7     public $age;
8
9
10    public function __construct($name, $age)
11    {
12        $this->name = $name;
13        $this->age = $age;
14    }
15
16 }
17
18
```

UTILIZANDO A CLASSE



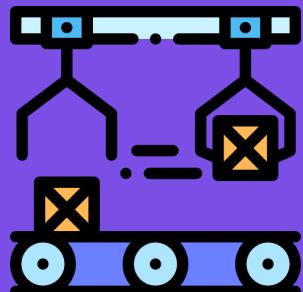
```
1 <?php
2
3 $person = new Person("Flávio Medeiros", 21);
4
5 echo "Olá eu sou o {$person->name} e tenho {$person->age} anos";
6 //Olá eu sou o Flávio Medeiros e tenho 21 anos
7
8 $person->name = "Marcos Couto";
9 $person->age = 24;
10
11 echo "Olá eu sou o {$person->name} e tenho {$person->age} anos";
12 //Olá eu sou o Marcos Couto e tenho 24 anos
13
```

EXTENDENDO UMA CLASSE

```
1 <?php
2
3 class Student extends Person {
4
5     public $room;
6
7     public function __construct($name, $age, $room)
8     {
9         $this->name = $name;
10        $this->age = $age;
11        $this->room = $room;
12    }
13
14 }
15
16
```

TIPANDO DADOS

```
1 // PHP > 7.0
2
3 class Person {
4
5     public $name;
6     public $age;
7
8     public function __construct(string $name, int $age): void
9     {
10         $this->name = $name;
11         $this->age = $age;
12     }
13 }
14
15
16 // PHP > 7.4
17
18 class Person {
19
20     public string $name;
21     public int $age;
22
23     ...
24
25 }
```



GERENCIADOR DE DEPENDÊNCIAS



Programmer's - Beyond IT



FRAMEWORKS

- Laravel
- Zend Framework
- Phalcon
- Symfony
- CakePHP
- CodeIgniter



Laravel

- Framework MVC
- CLI - Artisan
- ORM - Eloquent
- Migrations, Seed, Factory
- Rotas
- Autenticação própria
- View - Blade
- Validation
- Compiling Assets - Mix
- Web Sockets - Echo
- Teste Unitário - PHPUnit
- Teste Front-end - Dusk

Criando um projeto com Laravel

Premissa: o Composer deve estar instalado

```
● ● ●  
laravel new blog
```