

Migrando para o *php7*

Guia de Referência Rápida
Prof. Er Galvão Abbott

Table of Contents

Licença de uso.....	3
Introdução.....	3
Novas funcionalidades.....	4
PHP 7.0.....	4
1 – Operador de NULL Coalesce (??).....	4
2 – Operador de Comparação Combinada (Spaceship) (<=>).....	4
3 – Classes Anônimas.....	5
4 – Divisão Precisa de inteiros.....	5
5 – Group Use Declarations.....	5
6 – Retorno em Generators.....	6
7 – Scalar Type Hints.....	6
8 – Return Type Declarations.....	7
9 – Configuração de sessões via session_start.....	7
10 – Desserialização Filtrada.....	7
11 – Mecanismos CSPRNG confiáveis.....	8
12 – Arrays como valores de constantes.....	8
PHP 7.1.....	9
1 – Retorno do tipo void.....	9
2 – Tipos Nullable.....	9
3 – Visibilidade para constantes de classes.....	9
4 – Short syntax para list().....	10
Quebras de Compatibilidade Reversa.....	10
PHP 7.0.....	10
1 – Remoção de defaults múltiplos no switch.....	10
2 – Remoção das tags alternativas.....	10
3 – Remoção de expressões regulares POSIX.....	11
4 – Remoção de funções mysql_*.....	11
5 – Remoção do warning da configuração date.timezone.....	11
6 – Remoção dos construtores no estilo PHP4.....	11
PHP 7.1.....	12
1 – Remoção das funções mcrypt_*.....	12
2 – Remoção do hashing para session ID.....	12
Referências.....	13
Artigos.....	13
Links.....	13

Licença de uso

Este material é licenciado por Er Galvão Abbott sob uma licença Creative Commons Share-alike 4.0 Internacional – CC By-SA 4.0).

Resumidamente: você pode utilizar, compartilhar e adaptar este material, contanto que o crédito ao autor seja mantido e o seu material seja disponibilizado sob uma licença também share-alike.

Texto completo da licença: [aqui](#).

Introdução

Lançada no final de 2015, a versão 7.0 da linguagem PHP se tornou rapidamente a mais importante em sua história desde o lançamento da versão 5.3, seis anos antes. Dentre muitas novidades, o aumento significativo de performance (a versão 7 é estimada minimamente como duas vezes mais rápida do que a versão 5), aliado ao fato de que a computação em nuvem e virtualização tornaram trivial o controle dos ambientes além do desenvolvimento fez com que esta versão se tornasse irresistível para os desenvolvedores e empresas.

Apesar de tudo isso, muitas das novas funcionalidades da linguagem – por serem muito recentes – e quebras de compatibilidade reversa tornam a migração muitas vezes em um processo doloroso. Neste curso conversaremos em detalhe sobre tudo isso, e este material dispõe-se a ser uma referência rápida sobre as mudanças mais relevantes da linguagem para quando surgirem dúvidas.

Novas funcionalidades

PHP 7.0

1 – Operador de NULL Coalesce (??)

Antes do PHP7:

```
echo ((isset($a) and !is_null($a)) ? $a : 'no');
```

Após o PHP7:

```
echo $a ?? 'no';
```

2 – Operador de Comparação Combinada (Spaceship) (<=>)

Antes do PHP7:

```
if ($a < $b) {  
    return -1;  
} else if ($a == $b) {  
    return 0;  
} else {  
    return 1;  
}
```

Após o PHP7:

```
return $a <=> $b;
```

3 – Classes Anônimas

Antes do PHP7: Não era possível.

Após o PHP7:

```
$a = new class() {  
    public $x;  
    public $y;  
  
    public function __construct()  
    {  
        $this->x = 2;  
        $this->y = 3;  
    }  
};  
  
echo $a->x; // Resultado: 2
```

4 – Divisão Precisa de inteiros

Antes do PHP7:

```
echo (int)3/2;
```

Depois do PHP7:

```
echo intdiv(3, 2);
```

5 – Group Use Declarations

Antes do PHP7:

```
use Foo\Bar\Baz;  
use Foo\Bar\Quux;
```

Após o PHP7:

```
use Foo\Bar\  
    Baz,  
    Quux  
};
```

6 – Retorno em Generators

Antes do PHP7: Não era possível.

Após o PHP7:

```
function foo()
{
    yield 'a';
    yield 'b';
    return 'c';
}

$bar = foo();
echo $bar→current(); // 'a'
$bar→next();
echo $bar→current(); // 'b'
echo $bar→getReturn(); // 'c'
```

7 – Scalar Type Hints

Antes do PHP7: Não era possível. Apenas tipos compostos (Arrays e Objetos específicos).

Após o PHP7:

Tipos escalares para parâmetros de funções/métodos/closures: int, float, string e bool

```
function soma(float $x, float $y)
{
    return $x + $y;
}
```

Podem ser usados de forma estrita ou fraca (default), através de uma instrução declare().

Veremos os detalhes de cada implementação em sala de aula.

8 – Return Type Declarations

Antes do PHP7: Não era possível

Após o PHP7: Definição de tipagem de retorno, escalar ou composto.

```
function soma(float $x, float $y): float
{
    return $x + $y;
}
```

Podem ser usados de forma estrita ou fraca (default), através de uma instrução declare().

Veremos os detalhes de cada implementação em sala de aula.

9 – Configuração de sessões via session_start

Antes do PHP7: Só era possível via definição de configurações (php.ini, .user.ini, .htaccess, etc...)

Após o PHP7:

```
session_start([
    'name' => 'minhaSessao',
    'cookie_httponly' => '1',
]);
```

10 – Desserialização Filtrada

Antes do PHP7:

```
class foo
{
    public function __destruct()
    {
        echo 'Fail!' . PHP_EOL;
    }
}

$foo = new foo();
$bar = serialize($foo);
var_dump(unserialize($bar)); // Método destruidor é invocado
```

Após o PHP7:

```
class foo
{
    public function __destruct()
    {
        echo 'Fail!' . PHP_EOL;
    }
}

$foo = new foo();
$bar = serialize($foo);
var_dump(unserialize($bar, ['allowed_classes' => false])); // Método destruidor é ignorado
var_dump(unserialize($bar, ['allowed_classes' => ['bar']])); // Método destruidor é ignorado
var_dump(unserialize($bar, ['allowed_classes' => true])); // Método destruidor é invocado
var_dump(unserialize($bar, ['allowed_classes' => ['foo']])); // Método destruidor é invocado
```

11 – Mecanismos CSPRNG confiáveis

Antes do PHP7:

```
$int = rand(1, 10);
$int = mt_rand(1, 10);
$bin = openssl_random_pseudo_bytes(10);
```

Após o PHP7:

```
$int = random_int(1, 10);
$bin = random_bytes(10);
```

12 – Arrays como valores de constantes

Antes do PHP7: Não era possível.

Após o PHP7:

```
define('DB', [
    'HOST' => 'localhost',
    'USER' => 'foo',
]);
```


PHP 7.1

1 – Retorno do tipo void

Antes do PHP7.1: Não era possível.

Após o PHP7.1:

```
public function setId(int $id): void
{
    $this->id = $id;
}
```

2 – Tipos Nullable

Antes do PHP7.1: Não era possível.

Após o PHP7.1:

```
public function setID(?int $id): void
{
    if ($id === NULL) {
        $id = 1;
    }

    $this->id = $id;
}
```

3 – Visibilidade para constantes de classes

Antes do PHP7.1: Não era possível.

Após o PHP7.1:

```
class Foo
{
    public const BAR = 1;
    protected const BAZ = 'foo';
    private const QUUX = true;
}
```

4 – Short syntax para list()

Antes do PHP7.1: Somente era possível com list():

```
$a = ['foo', 'bar', 'baz'];  
list($m1, $m2, $m3) = $a;
```

Após o PHP7.1:

```
$a = ['foo', 'bar', 'baz'];  
[$m1, $m2, $m3] = $a;
```

Quebras de Compatibilidade Reversa

PHP 7.0

1 – Remoção de defaults múltiplos no switch

Antes do PHP7:

```
$f = 'foo';  
switch ($f) {  
    case 'bar':  
        echo 'é bar';  
        break;  
    default:  
        echo 'Não é bar';  
        break;  
    default:  
        echo 'Não é baz';  
        break;  
}
```

Após o PHP7:

```
Fatal error: Switch statements may only contain one default clause in %s on line %d
```

2 – Remoção das tags alternativas

Antes do PHP7: Era possível cercar código php com ASP tags (“<%”) ou mesmo tags “<script>”.

Após o PHP7:

Estas tags não estão mais disponíveis, incluindo a “Short ASP echo” (“<%=”).

3 – Remoção de expressões regulares POSIX

Antes do PHP7:

```
$foo = 'a1b2c3';  
var_dump(ereg('a', $foo)); // 1
```

Após o PHP7:

```
Warning: Uncaught Error: Call to undefined function ereg() in
```

4 – Remoção de funções mysql_*

Antes do PHP7:

```
$conn = mysql_connect('localhost', 'foo', 'bar');  
mysql_select_db('mydb');  
  
$sql = 'SELECT * FROM tabela';  
$result = mysql_query($sql);  
  
// ... etc...
```

Após o PHP7:

Apenas mysqli_* e PDO são suportados. Veremos mais detalhes e exemplos em sala de aula.

5 – Remoção do warning da configuração date.timezone

Antes do PHP7: Era emitido um warning quando se tentava usar datas sem a definição de timezone setada no php.ini.

Após o PHP7: Este setting é automaticamente setado para UTC (ou GMT) por default.

6 – Remoção dos construtores no estilo PHP4

Antes do PHP7.1: Era possível utilizar o “construtor clássico” de OO (nome do método === nome da classe).

Após o PHP7.1: Apenas o nome especial __construct é reconhecido.

PHP 7.1

1 – Remoção das funções mcrypt_*

Antes do PHP7.1: Era possível utilizar a extensão mcrypt para questões de criptografia.

Após o PHP7.1: A extensão foi removida, favorecendo o uso de openssl. Veremos a implementação disso em sala de aula.

2 – Remoção do hashing para session ID

Antes do PHP7.1: Session IDs eram gerados através de hashing.

Após o PHP7.1: Com implementações melhoradas de CSPRNG, não há mais necessidade do uso de hashing para a geração de Session IDs, portanto as seguintes configurações de sessão foram removidas:

- hash_function
- hash_bits_per_character
- entropy_file
- entropy_length

E duas novas configurações foram adicionadas:

- sid_length (default: 26)
- sid_bits_per_character (default: 5)

Referências

Este material se destina a ser uma referência rápida para as alterações onde provavelmente se encontrarão maiores problemas ao migrar. Existem muitas outras alterações pelas quais a linguagem passou/está passando que não se encontram aqui. Além disso, muitos detalhes interessantes e discussões mais longas sobre a linguagem estão disponíveis na web. Abaixo, alguns links que podem ser interessantes.

Artigos

- [PHP7: More strict! \(but only if you want it to be\)](#), by myself!
- [PHP7 is evolving! What's new?](#), by myself (again!)
- [PHP7 Reference](#), by Thomas Punt
- [PHP7 feature freeze](#), by Phil Sturgeon
- [An Exceptional Change in PHP 7.0](#), by Davey Shafik

Links

Todo o processo decisório de cada versão da linguagem é completamente transparente e tratado em detalhes. Acompanhe, informe-se e conheça:

<https://wiki.php.net/rfc>