



What's New in PHP 7.3

Andy Carter

andy-carter.com | [@drmonkeyninja](https://twitter.com/drmonkeyninja)

PHP Supported Versions

	Initial Release	Active Support Until	Security Support Until
5.6 	28 Aug 2014	19 Jan 2018	31 Dec 2018
7.1	1 Dec 2016	1 Dec 2018	1 Dec 2019
7.2	30 Nov 2017	30 Nov 2019	30 Nov 2020
7.3	6 Dec 2018	6 Dec 2020	6 Dec 2021

Source: <http://php.net/supported-versions.php>

Trailing commas in function calls

```
// For a long time now we've been able to leave a trailing comma
// at the end of an array

$array = [
    'Bugs Bunny',
    'Tweety',
    'Tasmanian Devil',
    'Porky Pig',
];
```

“Having a trailing comma after the last defined array entry, while unusual, is a valid syntax.” - <http://php.net/manual/en/function.array.php>

Trailing commas in function calls

```
// In PHP 7.3 this is now allowed
```

```
$data = foo(  
    'Bugs Bunny',  
    'Daffy Duck',  
);
```

```
// ...but this still isn't
```

```
function foo($a, $b,) {  
    return "$a $b";  
}
```

New array methods `array_key_first` & `array_key_last`

```
$items = [  
    'name' => 'Daffy Duck',  
    'alias' => 'Duck Dodgers',  
    'species' => 'American black duck',  
];  
  
// Instead of this  
  
reset($items);  
$firstKey = key($items);  
var_dump($firstKey); // 'name'  
  
end($items);  
$lastKey = key($items);  
var_dump($lastKey); // 'species'  
  
// You can now do  
  
$firstKey = array_key_first($items);  
var_dump($firstKey); // 'name'  
  
$lastKey = array_key_last($items);  
var_dump($lastKey); // 'species'
```

New `is_countable`

```
$items = null;

echo sprintf('There are %d items', count($items));
// error Warning: count(): Parameter must be an array or an object
// that implements Countable

// With PHP 7.3 we can check if variable is Countable

if (is_countable($items)) {
    echo sprintf('There are %d items', count($items));
}
```

New Heredoc/Nowdoc syntax

```
$name = 'Bugs';

$str = <<<EOD
Hello $name. This is      // Hello Bugs. This is
a heredoc example.      // a heredoc example.
EOD;

$str = <<<'EOD'
Hello $name. This is      // Hello $name. This is
a nowdoc example.        // a nowdoc example.
EOD;
```

New Heredoc/Nowdoc syntax

- The closing identifier for doc strings is no longer required to be followed by a semicolon or newline
- Additionally the closing identifier may be indented, in which case the indentation will be stripped from all lines in the doc string

```
$strings = [  
    'Looney Toons', <<<EOD  
    Steven Spielberg Presents Tiny Toon Adventures  
    EOD, 'Animaniacs',  
];
```


New Heredoc/Nowdoc syntax

```
// Instead of this
```

```
$query = <<<SQL  
SELECT *  
FROM `contacts`  
WHERE `telephone` IS NOT NULL;  
SQL;
```

```
// You can now do
```

```
$query = <<<SQL  
    SELECT *  
    FROM `contacts`  
    WHERE `telephone` IS NOT NULL;  
SQL;
```

New Heredoc/Nowdoc syntax

```
// This will cause a syntax error in PHP 7.3
```

```
$str = <<<F00  
abcdefg  
    F00  
F00;
```

```
// This can be resolved using a different heredoc label
```

```
$str = <<<BAR  
abcdefg  
    F00  
BAR;
```

JsonException

PHP 7.3 adds a new **JSON_THROW_ON_ERROR** option for `json_encode` and `json_decode`.

```
json_encode($data, JSON_THROW_ON_ERROR);  
  
json_decode("invalid json", null, 512, JSON_THROW_ON_ERROR);  
  
// Throws JsonException
```

JsonException

PHP 7.3 adds a new **JSON_THROW_ON_ERROR** option for `json_encode` and `json_decode`.

```
try {  
    json_decode("invalid json", null, 512, JSON_THROW_ON_ERROR);  
} catch (\JsonException $exception) {  
    echo $exception->getMessage(); // 'Syntax error'  
}
```

list reference assignment

```
$array = ['Bugs Bunny', 'Daffy Duck'];  
  
// list() and its shorthand [] syntax now support references  
  
list($a, &$b) = $array;  
  
$b = 'Porky Pig';  
  
// $array = ['Bugs Bunny', 'Porky Pig'];
```

Undefined variables in compact

compact creates an array containing variables and their values

```
$title = 'Hello world';
$body = 'Lorem ipsum';
$comments = [];

$array = compact('title', 'body', 'comments');

print_r($array);
// [
//     'title' => 'Hello world',
//     'body' => 'Lorem ipsum',
//     'comments' => [],
// ]
```

Undefined variables in compact

Undefined variables passed to compact are no longer ignored

```
$foo = 'Sylvester';  
compact('foo', 'bar');  
  
// Notice: compact(): Undefined variable: bar
```

Case-insensitive Constants 🙌

This is now deprecated.

```
// Prior to PHP 7.3 you could define case-insensitive constants  
  
define("GREETING", "Hello you.", true);  
echo GREETING; // outputs "Hello you."  
echo Greeting; // outputs "Hello you."
```


String Search Functions

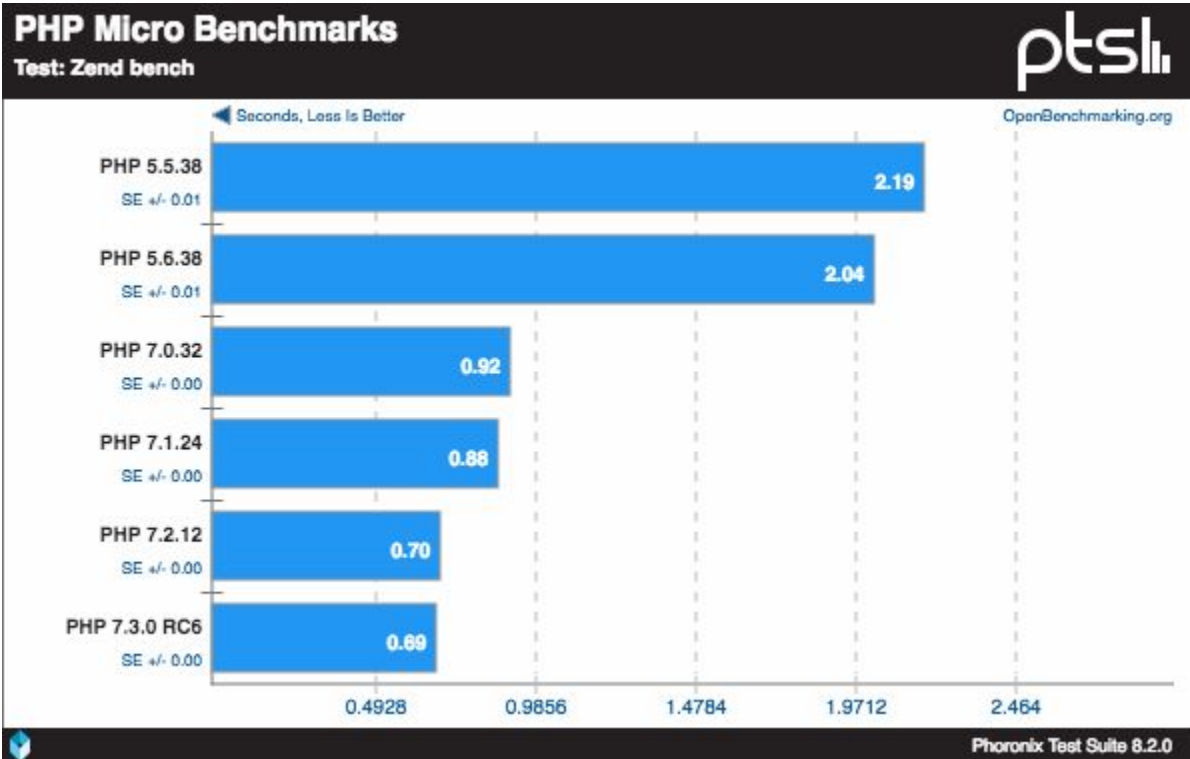
Passing a non-string needle to string search functions is deprecated.

The affected functions are: `strpos`, `strpos`, `stripos`, `stripos`, `strstr`, `strchr`, `strchr`, and `stristr`.

```
$haystack = 'Hello World 123!';
$needle = 2;

// In PHP 7.3 this will error
if (strpos($haystack, $needle) !== false) {
    echo 'Needle found';
}

// To fix cast the needle to a string
if (strpos($haystack, (string)$needle) !== false) {
    echo 'Needle found';
}
```



Source: www.phoronix.com

Coming to PHP 7.4?

- Preloading
- Typed properties
- Null Coalescing Assignment operator (??=)
- Foreign Function Interface



<http://php.net/manual/en/migration73.php>