
php-pseudolocale Documentation

Release 1.1.0

Jan 30, 2021

CONTENTS

1	Installation	3
2	Examples	5
3	License	7

The library is used to generate readable strings for ASCII latin to test Unicode support and/or detect missing translation strings. The result is a mix of extended latin, cyrillic, greek, armenian, georgian, japanese scripts with some math symbols.

INSTALLATION

Install by composer

```
composer require arokettu/pseudolocale
```


EXAMPLES

```
<?php
use Arokettu\Pseudolocale\Pseudolocale;

$text = 'The Quick Brown Fox Jumps Over The Lazy Dog';

// Lowercase letters:
echo Pseudolocale::pseudolocalize(strtolower($text));
// [-- ũıç ø ø ũš ø å øø --]

// Uppercase letters:
echo Pseudolocale::pseudolocalize(strtoupper($text));
// [-- ßØÑ ø ø ŁÅ ĐØ --]

// Replace only lowercase letters:
echo Pseudolocale::pseudolocalize($text, Pseudolocale::REPLACE_LOWERCASE);
// [-- T Qũıç Bø Fø Jũš O T Łå Dø --]

// Replace only uppercase letters:
echo Pseudolocale::pseudolocalize($text, Pseudolocale::REPLACE_UPPERCASE);
// [-- he uick ßrown ox umps Øver he Łazy Đog --]

// Redefine prefix and postfix:
echo Pseudolocale::pseudolocalize($text, Pseudolocale::REPLACE_ALL, '<', '>');
// or better, use new PHP 8 syntax:
echo Pseudolocale::pseudolocalize($text, prefix: '<', postfix: '>');
// < ũıç ßø ø ũš ø Łå Đø>

// Since 1.1 the library does not replace sprintf replacement patterns by default:
echo Pseudolocale::pseudolocalize("It's %d to go alone. Take %s with you");
// [-- 'š %d ø ø åø. å %s ı øū --]
// You can use your own patterns
echo Pseudolocale::pseudolocalize(
    'No, %username%! I am your %relative%!',
    Pseudolocale::REPLACE_ALL,
    '[-- ', ' --]',
    '/%\w+%/');
);
// PHP 8 is preferred:
echo Pseudolocale::pseudolocalize('No, %username%! I am your %relative%!',
    ↪regexPreserve: '/%\w+%/');
// [-- Ñø, %username%! å øū %relative%! --]
```

CHAPTER THREE

LICENSE

The library is available as open source under the terms of the [MIT License](#). See LICENSE.md