

---

# **transliterate Documentation**

***Release 1.8.1***

**Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>**

**Jul 07, 2017**



<b>1</b>	<b>Prerequisites</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage and examples</b>	<b>7</b>
3.1	Simple usage . . . . .	7
3.2	Registering a custom language pack . . . . .	9
3.2.1	Basics . . . . .	9
3.2.2	API in depth . . . . .	10
3.2.2.1	Mappings . . . . .	10
3.2.2.2	Additional . . . . .	11
3.3	Using the lorem ipsum generator . . . . .	11
3.4	Language detection . . . . .	11
3.5	Slugify . . . . .	12
<b>4</b>	<b>Missing a language pack?</b>	<b>13</b>
<b>5</b>	<b>License</b>	<b>15</b>
<b>6</b>	<b>Support</b>	<b>17</b>
<b>7</b>	<b>Author</b>	<b>19</b>
<b>8</b>	<b>Documentation</b>	<b>21</b>
8.1	transliterate package . . . . .	21
8.1.1	Subpackages . . . . .	21
8.1.1.1	transliterate.backports package . . . . .	21
8.1.1.1.1	Submodules . . . . .	21
8.1.1.1.2	transliterate.backports.collections module . . . . .	21
8.1.1.1.3	Module contents . . . . .	22
8.1.1.2	transliterate.contrib package . . . . .	22
8.1.1.2.1	Subpackages . . . . .	22
8.1.1.2.1.1	transliterate.contrib.apps package . . . . .	22
8.1.1.2.1.2	Subpackages . . . . .	22
8.1.1.2.1.3	transliterate.contrib.apps.translipsum package . . . . .	22
8.1.1.2.1.4	Submodules . . . . .	22
8.1.1.2.1.5	transliterate.contrib.apps.translipsum.utils module . . . . .	22

8.1.1.2.1.6	Module contents . . . . .	23
8.1.1.2.1.7	Module contents . . . . .	23
8.1.1.2.1.8	transliterate.contrib.languages package . . . . .	23
8.1.1.2.1.9	Subpackages . . . . .	23
8.1.1.2.1.10	transliterate.contrib.languages.bg package . . . . .	23
8.1.1.2.1.11	Subpackages . . . . .	23
8.1.1.2.1.12	transliterate.contrib.languages.bg.data package . . . . .	23
8.1.1.2.1.13	Submodules . . . . .	23
8.1.1.2.1.14	transliterate.contrib.languages.bg.data.python32 module . . . . .	23
8.1.1.2.1.15	transliterate.contrib.languages.bg.data.standard module . . . . .	23
8.1.1.2.1.16	Module contents . . . . .	23
8.1.1.2.1.17	Submodules . . . . .	23
8.1.1.2.1.18	transliterate.contrib.languages.bg.translit_language_pack module . . . . .	23
8.1.1.2.1.19	Module contents . . . . .	24
8.1.1.2.1.20	transliterate.contrib.languages.el package . . . . .	24
8.1.1.2.1.21	Subpackages . . . . .	24
8.1.1.2.1.22	transliterate.contrib.languages.el.data package . . . . .	24
8.1.1.2.1.23	Submodules . . . . .	24
8.1.1.2.1.24	transliterate.contrib.languages.el.data.python32 module . . . . .	24
8.1.1.2.1.25	transliterate.contrib.languages.el.data.standard module . . . . .	24
8.1.1.2.1.26	Module contents . . . . .	24
8.1.1.2.1.27	Submodules . . . . .	24
8.1.1.2.1.28	transliterate.contrib.languages.el.translit_language_pack module . . . . .	24
8.1.1.2.1.29	Module contents . . . . .	24
8.1.1.2.1.30	transliterate.contrib.languages.he package . . . . .	24
8.1.1.2.1.31	Submodules . . . . .	24
8.1.1.2.1.32	transliterate.contrib.languages.he.translit_language_pack module . . . . .	24
8.1.1.2.1.33	Module contents . . . . .	25
8.1.1.2.1.34	transliterate.contrib.languages.hi package . . . . .	25
8.1.1.2.1.35	Submodules . . . . .	25
8.1.1.2.1.36	transliterate.contrib.languages.hi.translit_language_pack module . . . . .	25
8.1.1.2.1.37	Module contents . . . . .	26
8.1.1.2.1.38	transliterate.contrib.languages.hy package . . . . .	26
8.1.1.2.1.39	Subpackages . . . . .	26
8.1.1.2.1.40	transliterate.contrib.languages.hy.data package . . . . .	26
8.1.1.2.1.41	Submodules . . . . .	26
8.1.1.2.1.42	transliterate.contrib.languages.hy.data.python32 module . . . . .	26
8.1.1.2.1.43	transliterate.contrib.languages.hy.data.standard module . . . . .	26
8.1.1.2.1.44	Module contents . . . . .	26
8.1.1.2.1.45	Submodules . . . . .	26
8.1.1.2.1.46	transliterate.contrib.languages.hy.translit_language_pack module . . . . .	26
8.1.1.2.1.47	Module contents . . . . .	27
8.1.1.2.1.48	transliterate.contrib.languages.ka package . . . . .	27
8.1.1.2.1.49	Subpackages . . . . .	27
8.1.1.2.1.50	transliterate.contrib.languages.ka.data package . . . . .	27
8.1.1.2.1.51	Submodules . . . . .	27
8.1.1.2.1.52	transliterate.contrib.languages.ka.data.python32 module . . . . .	27
8.1.1.2.1.53	transliterate.contrib.languages.ka.data.standard module . . . . .	27
8.1.1.2.1.54	Module contents . . . . .	27
8.1.1.2.1.55	Submodules . . . . .	27
8.1.1.2.1.56	transliterate.contrib.languages.ka.translit_language_pack module . . . . .	27
8.1.1.2.1.57	Module contents . . . . .	28
8.1.1.2.1.58	transliterate.contrib.languages.ru package . . . . .	28
8.1.1.2.1.59	Subpackages . . . . .	28

8.1.1.2.1.60	transliterate.contrib.languages.ru.data package . . . . .	28
8.1.1.2.1.61	Submodules . . . . .	28
8.1.1.2.1.62	transliterate.contrib.languages.ru.data.python32 module . . . . .	28
8.1.1.2.1.63	transliterate.contrib.languages.ru.data.standard module . . . . .	28
8.1.1.2.1.64	Module contents . . . . .	28
8.1.1.2.1.65	Submodules . . . . .	28
8.1.1.2.1.66	transliterate.contrib.languages.ru.translit_language_pack module .	28
8.1.1.2.1.67	Module contents . . . . .	29
8.1.1.2.1.68	transliterate.contrib.languages.uk package . . . . .	29
8.1.1.2.1.69	Subpackages . . . . .	29
8.1.1.2.1.70	transliterate.contrib.languages.uk.data package . . . . .	29
8.1.1.2.1.71	Submodules . . . . .	29
8.1.1.2.1.72	transliterate.contrib.languages.uk.data.python32 module . . . . .	29
8.1.1.2.1.73	transliterate.contrib.languages.uk.data.standard module . . . . .	29
8.1.1.2.1.74	Module contents . . . . .	29
8.1.1.2.1.75	Submodules . . . . .	29
8.1.1.2.1.76	transliterate.contrib.languages.uk.translit_language_pack module .	29
8.1.1.2.1.77	Module contents . . . . .	30
8.1.1.2.1.78	Module contents . . . . .	30
8.1.1.2.2	Module contents . . . . .	30
8.1.1.3	transliterate.tests package . . . . .	30
8.1.1.3.1	Subpackages . . . . .	30
8.1.1.3.1.1	transliterate.tests.data package . . . . .	30
8.1.1.3.1.2	Submodules . . . . .	30
8.1.1.3.1.3	transliterate.tests.data.normal module . . . . .	30
8.1.1.3.1.4	transliterate.tests.data.python32 module . . . . .	30
8.1.1.3.1.5	Module contents . . . . .	30
8.1.1.3.2	Submodules . . . . .	30
8.1.1.3.3	transliterate.tests.base module . . . . .	30
8.1.1.3.4	transliterate.tests.defaults module . . . . .	30
8.1.1.3.5	transliterate.tests.helpers module . . . . .	30
8.1.1.3.6	transliterate.tests.test_transliterate module . . . . .	30
8.1.1.3.7	Module contents . . . . .	32
8.1.2	Submodules . . . . .	32
8.1.3	transliterate.base module . . . . .	32
8.1.4	transliterate.conf module . . . . .	34
8.1.5	transliterate.decorators module . . . . .	34
8.1.6	transliterate.defaults module . . . . .	34
8.1.7	transliterate.discover module . . . . .	34
8.1.8	transliterate.exceptions module . . . . .	35
8.1.9	transliterate.helpers module . . . . .	35
8.1.10	transliterate.utils module . . . . .	35
8.1.11	Module contents . . . . .	36

## 9 Indices and tables 39

## Python Module Index 41



Bi-directional transliterator for Python. Transliterates (unicode) strings according to the rules specified in the language packs (source script <-> target script).

Comes with language packs for the following languages (listed in alphabetical order):

- Armenian
- Bulgarian (beta)
- Georgian
- Greek
- Macedonian (alpha)
- Mongolian (alpha)
- Russian
- Ukrainian (beta)

There are also a number of useful tools included, such as:

- Simple lorem ipsum generator, which allows lorem ipsum generation in the language chosen.
- Language detection for the text (if appropriate language pack is available).
- Slugify function for non-latin texts.





# CHAPTER 1

---

## Prerequisites

---

- Python 2.6.8+, 2.7.+, 3.3.+



## CHAPTER 2

---

### Installation

---

Install with latest stable version from PyPI.

```
$ pip install transliterate
```

or install the latest stable version from bitbucket:

```
$ pip install -e hg+https://bitbucket.org/barseghyanartur/transliterate@stable  
↪ #egg=transliterate
```

or install the latest stable version from github:

```
$ pip install -e git+http://github.com/barseghyanartur/transliterate@stable  
↪ #egg=transliterate
```

That's all. See the *Usage and examples* section for more.



---

## Usage and examples

---

### Simple usage

Required imports

```
from transliterate import translit, get_available_language_codes
```

Original text

```
text = "Lorem ipsum dolor sit amet"
```

Transliteration to Armenian

```
print translit(text, 'hy')  
  
# Output:
```

Transliteration to Georgian

```
print translit(text, 'ka')  
  
# Output:
```

Transliteration to Greek

```
print translit(text, 'el')  
  
# Output: Λορεμ ιψθμ δολορ σιτ αμετ
```

Transliteration to Russian

```
print translit(text, 'ru')  
  
# Output:
```

List of available (registered) languages

```
print get_available_language_codes()

# Output: ['el', 'hy', 'ka', 'ru']
```

Reversed transliterations are transliterations made from target language to source language (in terms they are defined in language packs). In case of reversed transliterations, you may leave out the `language_code` attribute, although if you know it on beforehand, specify it since it works faster that way.

Reversed transliteration from Armenian

```
print translit(u"      ", 'hy', reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Reversed transliteration from Armenian with `language_code` argument left out

```
print translit(u"      ", reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Reversed transliteration from Georgian

```
print translit(u"      ", 'ka', reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Reversed transliteration from Georgian with `language_code` argument left out

```
print translit(u"      ", reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Reversed transliteration from Greek

```
print translit(u"Λορεμ ιψθμ δολορ σιτ αμετ", 'el', reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Reversed transliteration from Greek with `language_code` argument left out

```
print translit(u"Λορεμ ιψθμ δολορ σιτ αμετ", reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Reversed transliteration from Russian (Cyrillic)

```
print translit(u"      ", 'ru', reversed=True)

# Output: Lorm ipsum dolor sit amt
```

Reversed transliteration from Russian (Cyrillic) with `language_code` argument left out

```
print translit(u"      ", reversed=True)

# Output: Lorem ipsum dolor sit amet
```

Testing the decorator

```
from transliterate.decorators import transliterate_function

@transliterate_function(language_code='hy')
def decorator_test(text):
    return text

print decorator_test(u"Lorem ipsum dolor sit amet")

# Output:
```

## Registering a custom language pack

### Basics

Make sure to call the *autodiscover* function before registering your own language packs if you want to use the bundled language packs along with your own custom ones.

```
from transliterate.discover import autodiscover
autodiscover()
```

Then the custom language pack part comes.

```
from transliterate.base import TranslitLanguagePack, registry

class ExampleLanguagePack(TranslitLanguagePack):
    language_code = "example"
    language_name = "Example"
    mapping = (
        u"abcdefghijkl",
        u"1234567890",
    )

registry.register(ExampleLanguagePack)

print get_available_language_codes()

# Output: ['el', 'hy', 'ka', 'ru', 'example']

print translit(text, 'example')

# Output: Lor5m 9psum 4olor s9t 1m5t
```

It's possible to replace existing language packs with your own ones. By default, existing language packs are not force-installed.

To force install a language pack, set the *force* argument to *True* when registering a language pack. In that case, if a language pack with same language code has already been registered, it will be replaced; otherwise, if language pack didn't exist in the registry, it will be just registered.

```
registry.register(ExampleLanguagePack, force=True)
```

Forced language packs can't be replaced or unregistered.

## API in depth

There are 7 class properties that you could/should be using in your language pack, of which 4 are various sorts of mappings.

### Mappings

- **mapping (tuple):** A tuple of two strings, that simply represent the mapping of characters from the source language to the target language. For example, if your source language is Latin and you want to convert “a”, “b”, “c”, “d” and “e” characters to appropriate characters in Russian Cyrillic, your mapping would look as follows:

```
mapping = (u"abcde", u"")
```

Example (taken from the Greek language pack).

```
mapping = (
    u"abgdezhiiklmnxoprstfywABGDEZHIKLMNXOPRSTYFWU",
    u"αβγδεζηκλμνξοπρστυφωθΑΒΓΔΕΖΗΚΛΜΝΞΟΠΡΣΤΥΦΩΘ",
)
```

- **reversed\_specific\_mapping (tuple):** When making reversed translations, the mapping property is still used, but in some cases you need to provide additional rules. This property (reversed\_specific\_mapping) is meant for such cases. Further, is alike the mapping.

Example (taken from the Greek language pack).

```
reversed_specific_mapping = (
    u"θΘ",
    u"υΥ"
)
```

- **pre\_processor\_mapping (dict):** A dictionary of mapping from source language to target language. Use this only in cases if a single character in source language shall be represented by more than one character in the target language.

Example (taken from the Greek language pack).

```
pre_processor_mapping = {
    u"th": u"θ",
    u"ch": u"χ",
    u"ps": u"ψ",
    u"TH": u"Θ",
    u"CH": u"Χ",
    u"PS": u"Ψ",
}
```

- **reversed\_specific\_pre\_processor\_mapping:** Same as pre\_processor\_mapping, but used in reversed translations.

Example (taken from the Armenian language pack)

```
reversed_specific_pre_processor_mapping = {
    u"": u"u",
    u"": u"U"
}
```



## Additional

- `character_ranges` (tuple): A tuple of character ranges (unicode table). Used in language detection. Works only if `detectable` property is set to `True`. Be aware, that language (or shall I better be saying - script) detection is very basic and is based on characters only.
- `detectable` (bool): If set to `True`, language pack would be used for automatic language detection.

## Using the lorem ipsum generator

Note, that due to incompatibility of the original *lorem-ipsum-generator* package with Python 3, when used with Python 3 *transliterate* uses its' own simplified fallback lorem ipsum generator (which still does the job).

Required imports

```
from transliterate.contrib.apps.translipsum import TranslipsumGenerator
```

Generating paragraphs in Armenian

```
g_am = TranslipsumGenerator(language_code='hy')
print g_am.generate_paragraph()

# Output:
# . .
# . U , .
# .
```

Generating sentence in Georgian

```
g_ka = TranslipsumGenerator(language_code='ka')
print g_ka.generate_sentence()

# Output:  c  c.
```

Generating sentence in Greek

```
g_el = TranslipsumGenerator(language_code='el')
print g_el.generate_sentence()

# Output: Νες ρρας αμετ, ελιτ νεστιβθλθμ εθ, αενεαν ναμ, τελλθσ ναριθσ.
```

Generating sentence in Russian (Cyrillic)

```
g_ru = TranslipsumGenerator(language_code='ru')
print g_ru.generate_sentence()

# Output:  cc, c q  .
```

## Language detection

Required imports

```
from transliterate import detect_language
```

Detect Armenian text

```
detect_language(u'      ')\n\n# Output: hy
```

Detect Georgian text

```
detect_language(u'      ')\n\n# Output: ka
```

Detect Greek text

```
detect_language(u'Λορεμ ιψθμ δολορ σιτ αμετ')\n\n# Output: el
```

Detect Russian (Cyrillic) text

```
detect_language(u'      ')\n\n# Output: ru
```

## Slugify

Required imports

```
from transliterate import slugify
```

Slugify Armenian text

```
slugify(u'      ')\n\n# Output: lorem-ipsum-dolor-sit-amet
```

Slugify Georgian text

```
slugify(u'      ')\n\n# Output: lorem-ipsum-dolor-sit-amet
```

Slugify Greek text

```
slugify(u'Λορεμ ιψθμ δολορ σιτ αμετ')\n\n# Output: lorem-ipsum-dolor-sit-amet
```

Slugify Russian (Cyrillic) text

```
slugify(u'      ')\n\n# Output: lorem-ipsum-dolor-sit-amet
```

## CHAPTER 4

---

### Missing a language pack?

---

Missing a language pack for your own language? Contribute to the project by making one and it will appear in a new version (which will be released very quickly).



## CHAPTER 5

---

License

---

GPL 2.0/LGPL 2.1



## CHAPTER 6

---

### Support

---

For any issues contact me at the e-mail given in the *Author* section.





## CHAPTER 7

---

Author

---

Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>



Contents:

## transliterate package

### Subpackages

#### transliterate.backports package

#### Submodules

#### transliterate.backports.collections module

**class** transliterate.backports.collections.**Counter** (*iterable=None, \*\*kws*)

Bases: dict

Dict subclass for counting hashable objects. Sometimes called a bag or multiset. Elements are stored as dictionary keys and their counts are stored as dictionary values.

```
>>> Counter('zyzygy')
Counter({'y': 3, 'z': 2, 'g': 1})
```

**copy()**

Like dict.copy() but returns a Counter instance instead of a dict.

**elements()**

Iterator over elements repeating each as many times as its count.

```
>>> c = Counter('ABCABC')
>>> sorted(c.elements())
['A', 'A', 'B', 'B', 'C', 'C']
```

If an element's count has been set to zero or is a negative number, `elements()` will ignore it.

**classmethod** `fromkeys` (*iterable*, *v=None*)

**most\_common** (*n=None*)

List the *n* most common elements and their counts from the most common to the least. If *n* is `None`, then list all element counts.

```
>>> Counter('abracadabra').most_common(3)
[('a', 5), ('r', 2), ('b', 2)]
```

**update** (*iterable=None*, *\*\*kws*)

Like `dict.update()` but add counts instead of replacing them.

Source can be an iterable, a dictionary, or another `Counter` instance.

```
>>> c = Counter('which')
>>> c.update('witch')           # add elements from another iterable
>>> d = Counter('watch')
>>> c.update(d)                 # add elements from another counter
>>> c['h']                      # four 'h' in which, witch, and watch
4
```

## Module contents

### transliterate.contrib package

#### Subpackages

#### transliterate.contrib.apps package

#### Subpackages

#### transliterate.contrib.apps.translipsum package

#### Submodules

#### transliterate.contrib.apps.translipsum.utils module

**class** `transliterate.contrib.apps.translipsum.utils.Generator` (*\*args*, *\*\*kwargs*)  
Bases: `object`

Fallback lorem ipsum generator for Python 3.

**generate\_paragraph** (*num\_sentences=4*)

**generate\_sentence** ()

**generate\_word** ()



```
pre_processor_mapping = {u'ch': u'\u0447', u'zh': u'\u0436', u'ya': u'\u044f', u'Ts': u'\u0426', u'Sht': u'\u0429',  
reversed_specific_mapping = (u'\u044c\u044a\u042a', u'yaA')
```

## Module contents

## transliterate.contrib.languages.el package

## Subpackages

## transliterate.contrib.languages.el.data package

## Submodules

## transliterate.contrib.languages.el.data.python32 module

## transliterate.contrib.languages.el.data.standard module

## Module contents

## Submodules

## transliterate.contrib.languages.el.translit\_language\_pack module

```
class transliterate.contrib.languages.el.translit_language_pack.GreekLanguagePack
    Bases: transliterate.base.TranslitLanguagePack
```

Language pack for Greek language. See [http://en.wikipedia.org/wiki/Greek\\_alphabet](http://en.wikipedia.org/wiki/Greek_alphabet) for details.

```
character_ranges = ((880, 1023), (7936, 8191))
```

```
detectable = True
```

```
language_code = 'el'
```

```
language_name = 'Greek'
```

[illegible]

```
pre_processor_mapping = {u'iy': u'\u03b7\u03c5', u'Ps': u'\u03a8', u'u': u'\u03c5', u'ch': u'\u03c7', u'Oy': u'\u0394'}
```

```
reversed_specific_mapping = (u'\u03c2\u03ac\u03ad\u03ae\u03af\u03cd\u03cc\u03ce\u03ca\u03cb\u0390\u03b0\u03
```

## Module contents

## transliterate.contrib.languages.he package

## Submodules

## transliterate.contrib.languages.he.translit\_language\_pack module

```
class transliterate.contrib.languages.he.translit_language_pack.HebrewLanguagePack
    Bases: transliterate.base.TranslitLanguagePack
```













## Module contents

## Module contents

## Module contents

## transliterate.tests package

## Subpackages

## transliterate.tests.data package

## Submodules

## transliterate.tests.data.normal module

## transliterate.tests.data.python32 module

## Module contents

## Submodules

## transliterate.tests.base module

## transliterate.tests.defaults module

## transliterate.tests.helpers module

`transliterate.tests.helpers.print_info` (*func*)  
Prints some useful info.

`transliterate.tests.helpers.py2only` (*func*)  
Skips the test on Python 3.

## transliterate.tests.test\_transliterate module

```
class transliterate.tests.test_transliterate.TransliterateTest (methodName='runTest')
    Bases: unittest.case.TestCase
    Tests of transliterate.utils.translit.

    setUp ()

    test_01_get_available_language_codes (*args, **kwargs)
    test_02_translit_latin_to_armenian (*args, **kwargs)
    test_03_translit_latin_to_georgian (*args, **kwargs)
    test_04_translit_latin_to_greek (*args, **kwargs)
    test_06_translit_latin_to_bulgarian_cyrillic (*args, **kwargs)
```

```
test_06_translit_latin_to_cyrillic(*args, **kwargs)
test_06_translit_latin_to_mongolian_cyrillic(*args, **kwargs)
test_06_translit_latin_to_ukrainian_cyrillic(*args, **kwargs)
test_07_translit_armenian_to_latin(*args, **kwargs)
test_08_translit_georgian_to_latin(*args, **kwargs)
test_09_translit_greek_to_latin(*args, **kwargs)
test_11_translit_bulgarian_cyrillic_to_latin(*args, **kwargs)
test_11_translit_cyrillic_to_latin(*args, **kwargs)
test_11_translit_mongolian_cyrillic_to_latin(*args, **kwargs)
test_11_translit_ukrainian_cyrillic_to_latin(*args, **kwargs)
test_12_function_decorator(*args, **kwargs)
test_13_method_decorator(*args, **kwargs)
test_14_function_decorator(*args, **kwargs)
test_15_register_custom_language_pack(*args, **kwargs)
test_16_translipsum_generator_armenian(*args, **kwargs)
test_17_translipsum_generator_georgian(*args, **kwargs)
test_18_translipsum_generator_greek(*args, **kwargs)
test_20_translipsum_generator_bulgarian_cyrillic(*args, **kwargs)
test_20_translipsum_generator_cyrillic(*args, **kwargs)
test_20_translipsum_generator_mongolian_cyrillic(*args, **kwargs)
test_20_translipsum_generator_ukrainian_cyrillic(*args, **kwargs)
test_21_language_detection_armenian(*args, **kwargs)
test_22_language_detection_georgian(*args, **kwargs)
test_23_language_detection_greek(*args, **kwargs)
test_25_false_language_detection_cyrillic(*args, **kwargs)
test_25_language_detection_cyrillic(*args, **kwargs)
test_26_slugify_armenian(*args, **kwargs)
test_27_slugify_georgian(*args, **kwargs)
test_28_slugify_greek(*args, **kwargs)
test_30_slugify_bulgarian_cyrillic(*args, **kwargs)
test_30_slugify_cyrillic(*args, **kwargs)
test_30_slugify_mongolian_cyrillic(*args, **kwargs)
test_30_slugify_ukrainian_cyrillic(*args, **kwargs)
test_31_override_settings(*args, **kwargs)
test_32_auto_translit_reversed(*args, **kwargs)
test_33_register_unregister(*args, **kwargs)
```

## Module contents

## Submodules

### transliterate.base module

**class** transliterate.base.**TranslitLanguagePack**

Bases: object

Base language pack. The attributes below shall be defined in every language pack.

**language\_code:** Language code (obligatory). Example value: 'hy', 'ru'. **language\_name:** Language name (obligatory). Example value: 'Armenian', 'Russian'.

**character\_ranges:** Character ranges that are specific to the language. When making a pack, check [this](#) page for the ranges.

**mapping:** Mapping (obligatory). A tuple, consisting of two strings (source and target). Example value: (u'abc', u'').

**reversed\_specific\_mapping:** Specific mapping (one direction only) used when transliterating from target script to source script (reversed transliteration).

**pre\_processor\_mapping:** Pre processor mapping (optional). A dictionary mapping for letters that can't be represented by a single latin letter.

**reversed\_specific\_pre\_processor\_mapping:** Pre processor mapping ( optional). A dictionary mapping for letters that can't be represented by a single latin letter (reversed transliteration).

#### example

```
>>> class ArmenianLanguagePack(TranslitLanguagePack):
>>>     language_code = "hy"
>>>     language_name = "Armenian"
>>>     character_ranges = ((0x0530, 0x058F), (0xFB10, 0xFB1F))
>>>     mapping = (
>>>         u"abgdezilxkhmjnpsvtrcq&ofABGDEZILXKHMJNPSVTRCQOF", # Source script
>>>         u"", # Target script
>>>     )
>>>     reversed_specific_mapping = (
>>>         u"",
>>>         u"rR"
>>>     )
>>>     pre_processor_mapping = {
>>>         # lowercase
>>>         u"e'": u"",
>>>         u"y": u"",
>>>         u"th": u"",
>>>         u"jh": u"",
>>>         u"ts": u"",
>>>         u"dz": u"",
>>>         u"gh": u"",
>>>         u"tch": u"",
>>>         u"sh": u"",
>>>         u"vo": u"",
>>>         u"ch": u",
```

```

>>>     u"dj": u"",
>>>     u"ph": u"",
>>>     u"u": u"",
>>>
>>>     # uppercase
>>>     u"E'": u"",
>>>     u"Y": u"",
>>>     u"Th": u"",
>>>     u"Jh": u"",
>>>     u"Ts": u"",
>>>     u"Dz": u"",
>>>     u"Gh": u"",
>>>     u"Tch": u"",
>>>     u"Sh": u"",
>>>     u"Vo": u"",
>>>     u"Ch": u"",
>>>     u"Dj": u"",
>>>     u"Ph": u"",
>>>     u"U": u""
>>> }
>>> reversed_specific_pre_processor_mapping = {
>>>     u"": u"u",
>>>     u"": u"U"
>>> }

```

Note, that in Python 3 you won't be using u prefix before the strings.

**character\_ranges** = None

**characters** = None

**classmethod contains** (*character*)

Checks if given character belongs to the language pack.

**Return bool**

**detect** (*text*, *num\_words*=None)

Heavy language detection, which is activated for languages that are harder detect (like Russian Cyrillic and Ukrainian Cyrillic).

**Parameters**

- **value** (*unicode*) – Input string.
- **num\_words** (*int*) – Number of words to base decision on.

**Return bool** True if detected and False otherwise.

**detectable** = False

**language\_code** = None

**language\_name** = None

**make\_strict** (*value*, *reversed*=False)

Strips out unnecessary characters from the string.

**Parameters**

- **value** (*string*) –
- **reversed** (*bool*) –

**Return string**

```
mapping = None
pre_processor_mapping = None
pre_processor_mapping_keys = []
reversed_characters = None
reversed_pre_processor_mapping_keys = []
reversed_specific_mapping = None
reversed_specific_pre_processor_mapping = None
reversed_specific_pre_processor_mapping_keys = []
suggest (value, reversed=False, limit=None)
    Suggest possible variants (some sort of auto-complete).
```

#### Parameters

- **value** (*str*) –
- **limit** (*int*) – Limit number of suggested variants.

#### Return list

```
translit (value, reversed=False, strict=False, fail_silently=True)
    Transliterates the given value according to the rules set in the transliteration pack.
```

#### Parameters

- **value** (*str*) –
- **reversed** (*bool*) –
- **strict** (*bool*) –

#### Return str

## transliterate.conf module

## transliterate.decorators module

```
transliterate.decorators.transliterate_function
    alias of TransliterateFunction
```

```
transliterate.decorators.transliterate_method
    alias of TransliterateMethod
```

## transliterate.defaults module

## transliterate.discover module

```
transliterate.discover.autodiscover()
    Auto-discovers the language packs in contrib/apps.
```



## transliterate.exceptions module

**exception** transliterate.exceptions.**LanguageCodeError**

Bases: exceptions.Exception

Exception raised when language code is left empty or has incorrect value.

**exception** transliterate.exceptions.**ImproperlyConfigured**

Bases: exceptions.Exception

Exception raised when developer didn't configure the code properly.

**exception** transliterate.exceptions.**LanguagePackNotFound**

Bases: exceptions.Exception

Exception raised when language pack is not found for the language code given.

**exception** transliterate.exceptions.**LanguageDetectionError**

Bases: exceptions.Exception

Exception raised when language can't be detected for the text given.

**exception** transliterate.exceptions.**InvalidRegistryItemType**

Bases: exceptions.ValueError

Raised when an attempt is made to register an item in the registry which does not have a proper type.

## transliterate.helpers module

transliterate.helpers.**PROJECT\_DIR** (*base*)

## transliterate.utils module

transliterate.utils.**translit** (*value*, *language\_code=None*, *reversed=False*, *strict=False*)

Transliterates the text for the language given. Language code is optional in case of reversed translations (from some script to latin).

### Parameters

- **value** (*str*) –
- **language\_code** (*str*) –
- **reversed** (*bool*) – If set to True, reversed translation is made.
- **strict** (*bool*) – If given, all that are not found in the transliteration pack, are simply stripped out.

### Return str

transliterate.utils.**suggest** (*value*, *language\_code=None*, *reversed=False*, *limit=None*)

Suggest possible variants.

### Parameters

- **value** (*str*) –
- **language\_code** (*str*) –
- **reversed** (*bool*) – If set to True, reversed translation is made.
- **limit** (*int*) – Limit number of suggested variants.

**Return list**

`transliterate.utils.detect_language(text, num_words=None, fail_silently=True, heavy_check=False)`

Detects the language from the value given based on ranges defined in active language packs.

**Parameters**

- **value** (*unicode*) – Input string.
- **num\_words** (*int*) – Number of words to base decision on.
- **fail\_silently** (*bool*) –
- **heavy\_check** (*bool*) – If given, heavy checks would be applied when simple checks don't give any results. Heavy checks are language specific and do not apply to a common logic. Heavy language detection is defined in the `detect` method of each language pack.

**Return str** Language code.

`transliterate.utils.slugify(text, language_code=None)`

Slugifies the given text. If no `language_code` is given, auto-detects the language code from text given.

**Parameters** **text** (*unicode*) –

**Return str**

## Module contents

`transliterate.translit(value, language_code=None, reversed=False, strict=False)`

Transliterates the text for the language given. Language code is optional in case of reversed translations (from some script to latin).

**Parameters**

- **value** (*str*) –
- **language\_code** (*str*) –
- **reversed** (*bool*) – If set to True, reversed translation is made.
- **strict** (*bool*) – If given, all that are not found in the transliteration pack, are simply stripped out.

**Return str**

`transliterate.get_available_language_codes()`

Gets list of language codes for registered language packs.

**Return list**

`transliterate.detect_language(text, num_words=None, fail_silently=True, heavy_check=False)`

Detects the language from the value given based on ranges defined in active language packs.

**Parameters**

- **value** (*unicode*) – Input string.
- **num\_words** (*int*) – Number of words to base decision on.
- **fail\_silently** (*bool*) –
- **heavy\_check** (*bool*) – If given, heavy checks would be applied when simple checks don't give any results. Heavy checks are language specific and do not apply to a common logic. Heavy language detection is defined in the `detect` method of each language pack.

**Return str** Language code.

`transliterate.slugify(text, language_code=None)`

Slugifies the given text. If no `language_code` is given, auto-detects the language code from text given.

**Parameters** `text` (*unicode*) –

**Return str**

`transliterate.get_available_language_packs()`

Gets list of registered language packs.

**Return list**



## CHAPTER 9

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### t

transliterate, 36  
transliterate.backports, 22  
transliterate.backports.collections, 21  
transliterate.base, 32  
transliterate.conf, 34  
transliterate.contrib, 30  
transliterate.contrib.apps, 23  
transliterate.contrib.apps.translipsisum, 23  
transliterate.contrib.apps.translipsisum.utils, 22  
transliterate.contrib.languages, 30  
transliterate.contrib.languages.bg, 24  
transliterate.contrib.languages.bg.data, 23  
transliterate.contrib.languages.bg.data.python32, 23  
transliterate.contrib.languages.bg.data.standard, 23  
transliterate.contrib.languages.bg.translit\_language\_pack, 23  
transliterate.contrib.languages.el, 24  
transliterate.contrib.languages.el.data, 24  
transliterate.contrib.languages.el.data.python32, 24  
transliterate.contrib.languages.el.data.standard, 24  
transliterate.contrib.languages.el.translit\_language\_pack, 24  
transliterate.contrib.languages.he, 25  
transliterate.contrib.languages.he.translit\_language\_pack, 24  
transliterate.contrib.languages.hi, 26  
transliterate.contrib.languages.hi.translit\_language\_pack, 25  
transliterate.contrib.languages.hy, 27  
transliterate.contrib.languages.hy.data, 26  
transliterate.contrib.languages.hy.data.python32, 26  
transliterate.contrib.languages.hy.data.standard, 26  
transliterate.contrib.languages.hy.translit\_language\_pack, 26  
transliterate.contrib.languages.ka, 28  
transliterate.contrib.languages.ka.data, 27  
transliterate.contrib.languages.ka.data.python32, 27  
transliterate.contrib.languages.ka.data.standard, 27  
transliterate.contrib.languages.ka.translit\_language\_pack, 27  
transliterate.contrib.languages.ru, 29  
transliterate.contrib.languages.ru.data, 28  
transliterate.contrib.languages.ru.data.python32, 28  
transliterate.contrib.languages.ru.data.standard, 28  
transliterate.contrib.languages.ru.translit\_language\_pack, 28  
transliterate.contrib.languages.uk, 30  
transliterate.contrib.languages.uk.data, 29  
transliterate.contrib.languages.uk.data.python32, 29  
transliterate.contrib.languages.uk.data.standard, 29  
transliterate.contrib.languages.uk.translit\_language\_pack, 29  
transliterate.decorators, 34  
transliterate.defaults, 34  
transliterate.discover, 34  
transliterate.exceptions, 35  
transliterate.helpers, 35  
transliterate.tests, 32

`transliterate.tests.base`, [30](#)  
`transliterate.tests.data`, [30](#)  
`transliterate.tests.data.normal`, [30](#)  
`transliterate.tests.data.python32`, [30](#)  
`transliterate.tests.defaults`, [30](#)  
`transliterate.tests.helpers`, [30](#)  
`transliterate.tests.test_transliterate`,  
    [30](#)  
`transliterate.utils`, [35](#)



## A

ArmenianLanguagePack (class in transliterate.contrib.languages.hy.translit\_language\_pack), 26

autodiscover() (in module transliterate.discover), 34

## B

BulgarianLanguagePack (class in transliterate.contrib.languages.bg.translit\_language\_pack), 23

## C

character\_ranges (transliterate.base.TranslitLanguagePack attribute), 33

character\_ranges (transliterate.contrib.languages.bg.translit\_language\_pack.BulgarianLanguagePack attribute), 23

character\_ranges (transliterate.contrib.languages.el.translit\_language\_pack.GreekLanguagePack attribute), 24

character\_ranges (transliterate.contrib.languages.he.translit\_language\_pack.HebrewLanguagePack attribute), 25

character\_ranges (transliterate.contrib.languages.hi.translit\_language\_pack.HindiLanguagePack attribute), 25

character\_ranges (transliterate.contrib.languages.hy.translit\_language\_pack.ArmenianLanguagePack attribute), 26

character\_ranges (transliterate.contrib.languages.ka.translit\_language\_pack.GeorgianLanguagePack attribute), 27

character\_ranges (transliterate.contrib.languages.ru.translit\_language\_pack.RussianLanguagePack attribute), 28

character\_ranges (transliterate.contrib.languages.uk.translit\_language\_pack.UkrainianLanguagePack attribute), 29

characters (transliterate.base.TranslitLanguagePack attribute), 33

contains() (transliterate.base.TranslitLanguagePack class method), 33

copy() (transliterate.backports.collections.Counter method), 21

Counter (class in transliterate.backports.collections), 21

## D

detect() (transliterate.base.TranslitLanguagePack method), 33

detect\_language() (in module transliterate), 36

detect\_language() (in module transliterate.utils), 36

detectable (transliterate.base.TranslitLanguagePack attribute), 33

detectable (transliterate.contrib.languages.bg.translit\_language\_pack.BulgarianLanguagePack attribute), 23

detectable (transliterate.contrib.languages.el.translit\_language\_pack.GreekLanguagePack attribute), 24

detectable (transliterate.contrib.languages.he.translit\_language\_pack.HebrewLanguagePack attribute), 25

detectable (transliterate.contrib.languages.hi.translit\_language\_pack.HindiLanguagePack attribute), 25

detectable (transliterate.contrib.languages.hy.translit\_language\_pack.ArmenianLanguagePack attribute), 26

detectable (transliterate.contrib.languages.ka.translit\_language\_pack.GeorgianLanguagePack attribute), 27

detectable (transliterate.contrib.languages.ru.translit\_language\_pack.RussianLanguagePack attribute), 28

## E

elements() (transliterate.backports.collections.Counter method), 21

## F

fromkeys() (transliterate.backports.collections.Counter class method), 22

## G

generate\_paragraph() (transliterate-

ate.contrib.apps.translipsisum.TranslipsisumGenerator (transliterate.contrib.apps.translipsisum.translit\_language\_pack.ArmenianLanguagePack attribute), 23  
 generate\_paragraph() (transliterate.contrib.apps.translipsisum.utils.Generator language\_code (transliterate.contrib.languages.ka.translit\_language\_pack.GeorgianLanguagePack attribute), 22  
 generate\_sentence() (transliterate.contrib.apps.translipsisum.TranslipsisumGenerator language\_code (transliterate.contrib.languages.ru.translit\_language\_pack.RussianLanguagePack attribute), 27  
 generate\_sentence() (transliterate.contrib.apps.translipsisum.utils.Generator language\_code (transliterate.contrib.languages.uk.translit\_language\_pack.UkrainianLanguagePack attribute), 22  
 generate\_word() (transliterate.contrib.apps.translipsisum.utils.Generator language\_name (transliterate.base.TranslitLanguagePack attribute), 22  
 Generator (class in transliterate.contrib.apps.translipsisum.utils), 22  
 GeorgianLanguagePack (class in transliterate.contrib.languages.ka.translit\_language\_pack), language\_name (transliterate.contrib.languages.el.translit\_language\_pack.GreekLanguagePack attribute), 27  
 get\_available\_language\_codes() (in module transliterate), language\_name (transliterate.contrib.languages.he.translit\_language\_pack.HebrewLanguagePack attribute), 36  
 get\_available\_language\_packs() (in module transliterate), language\_name (transliterate.contrib.languages.hi.translit\_language\_pack.HindiLanguagePack attribute), 37  
 GreekLanguagePack (class in transliterate.contrib.languages.el.translit\_language\_pack), language\_name (transliterate.contrib.languages.hy.translit\_language\_pack.ArmenianLanguagePack attribute), 24  
 H  
 HebrewLanguagePack (class in transliterate.contrib.languages.he.translit\_language\_pack), language\_name (transliterate.contrib.languages.ka.translit\_language\_pack.GeorgianLanguagePack attribute), 24  
 HindiLanguagePack (class in transliterate.contrib.languages.hi.translit\_language\_pack), language\_name (transliterate.contrib.languages.ru.translit\_language\_pack.RussianLanguagePack attribute), 25  
 I  
 ImproperlyConfigured, 35  
 InvalidRegistryItemType, 35  
 L  
 language\_code (transliterate.base.TranslitLanguagePack attribute), 33  
 language\_code (transliterate.contrib.languages.bg.translit\_language\_pack.BulgarianLanguagePack attribute), 23  
 language\_code (transliterate.contrib.languages.el.translit\_language\_pack.GreekLanguagePack attribute), 24  
 language\_code (transliterate.contrib.languages.he.translit\_language\_pack.HebrewLanguagePack attribute), 25  
 language\_code (transliterate.contrib.languages.hi.translit\_language\_pack.HindiLanguagePack attribute), 25  
 M  
 make\_strict() (transliterate.base.TranslitLanguagePack method), 33  
 mapping (transliterate.base.TranslitLanguagePack attribute), 33  
 mapping (transliterate.contrib.languages.bg.translit\_language\_pack.BulgarianLanguagePack attribute), 23  
 mapping (transliterate.contrib.languages.el.translit\_language\_pack.GreekLanguagePack attribute), 24  
 mapping (transliterate.contrib.languages.he.translit\_language\_pack.HebrewLanguagePack attribute), 25



method), 30		method), 31	
test_04_translit_latin_to_greek()	(transliterate.tests.test_transliterate.TransliterateTest method), 30	test_18_translipsum_generator_greek()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_06_translit_latin_to_bulgarian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 30	test_20_translipsum_generator_bulgarian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_06_translit_latin_to_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 30	test_20_translipsum_generator_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_06_translit_latin_to_mongolian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_20_translipsum_generator_mongolian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_06_translit_latin_to_ukrainian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_20_translipsum_generator_ukrainian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_07_translit_armenian_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_21_language_detection_armenian()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_08_translit_georgian_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_22_language_detection_georgian()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_09_translit_greek_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_23_language_detection_greek()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_11_translit_bulgarian_cyrillic_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_25_false_language_detection_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_11_translit_cyrillic_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_25_language_detection_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_11_translit_mongolian_cyrillic_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_26_slugify_armenian()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_11_translit_ukrainian_cyrillic_to_latin()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_27_slugify_georgian()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_12_function_decorator()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_28_slugify_greek()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_13_method_decorator()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_30_slugify_bulgarian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_14_function_decorator()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_30_slugify_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_15_register_custom_language_pack()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_30_slugify_mongolian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_16_translipsum_generator_armenian()	(transliterate.tests.test_transliterate.TransliterateTest method), 31	test_30_slugify_ukrainian_cyrillic()	(transliterate.tests.test_transliterate.TransliterateTest method), 31
test_17_translipsum_generator_georgian()	(transliterate.tests.test_transliterate.TransliterateTest	test_31_override_settings()	(transliterate.tests.test_transliterate.TransliterateTest

- method), 31
  - test\_32\_auto\_translit\_reversed() (transliterate.tests.test\_transliterate.TransliterateTest method), 31
  - test\_33\_register\_unregister() (transliterate.tests.test\_transliterate.TransliterateTest method), 31
  - TranslipsisumGenerator (class in transliterate.contrib.apps.translipsisum), 23
  - translit() (in module transliterate), 36
  - translit() (in module transliterate.utils), 35
  - translit() (transliterate.base.TranslitLanguagePack method), 34
  - transliterate (module), 36
  - transliterate.backports (module), 22
  - transliterate.backports.collections (module), 21
  - transliterate.base (module), 32
  - transliterate.conf (module), 34
  - transliterate.contrib (module), 30
  - transliterate.contrib.apps (module), 23
  - transliterate.contrib.apps.translipsisum (module), 23
  - transliterate.contrib.apps.translipsisum.utils (module), 22
  - transliterate.contrib.languages (module), 30
  - transliterate.contrib.languages.bg (module), 24
  - transliterate.contrib.languages.bg.data (module), 23
  - transliterate.contrib.languages.bg.data.python32 (module), 23
  - transliterate.contrib.languages.bg.data.standard (module), 23
  - transliterate.contrib.languages.bg.translit\_language\_pack (module), 23
  - transliterate.contrib.languages.el (module), 24
  - transliterate.contrib.languages.el.data (module), 24
  - transliterate.contrib.languages.el.data.python32 (module), 24
  - transliterate.contrib.languages.el.data.standard (module), 24
  - transliterate.contrib.languages.el.translit\_language\_pack (module), 24
  - transliterate.contrib.languages.he (module), 25
  - transliterate.contrib.languages.he.translit\_language\_pack (module), 24
  - transliterate.contrib.languages.hi (module), 26
  - transliterate.contrib.languages.hi.translit\_language\_pack (module), 25
  - transliterate.contrib.languages.hy (module), 27
  - transliterate.contrib.languages.hy.data (module), 26
  - transliterate.contrib.languages.hy.data.python32 (module), 26
  - transliterate.contrib.languages.hy.data.standard (module), 26
  - transliterate.contrib.languages.hy.translit\_language\_pack (module), 26
  - transliterate.contrib.languages.ka (module), 28
  - transliterate.contrib.languages.ka.data (module), 27
  - transliterate.contrib.languages.ka.data.python32 (module), 27
  - transliterate.contrib.languages.ka.data.standard (module), 27
  - transliterate.contrib.languages.ka.translit\_language\_pack (module), 27
  - transliterate.contrib.languages.ru (module), 29
  - transliterate.contrib.languages.ru.data (module), 28
  - transliterate.contrib.languages.ru.data.python32 (module), 28
  - transliterate.contrib.languages.ru.data.standard (module), 28
  - transliterate.contrib.languages.ru.translit\_language\_pack (module), 28
  - transliterate.contrib.languages.uk (module), 30
  - transliterate.contrib.languages.uk.data (module), 29
  - transliterate.contrib.languages.uk.data.python32 (module), 29
  - transliterate.contrib.languages.uk.data.standard (module), 29
  - transliterate.contrib.languages.uk.translit\_language\_pack (module), 29
  - transliterate.decorators (module), 34
  - transliterate.defaults (module), 34
  - transliterate.discover (module), 34
  - transliterate.exceptions (module), 35
  - transliterate.helpers (module), 35
  - transliterate.tests (module), 32
  - transliterate.tests.base (module), 30
  - transliterate.tests.data (module), 30
  - transliterate.tests.data.normal (module), 30
  - transliterate.tests.data.python32 (module), 30
  - transliterate.tests.defaults (module), 30
  - transliterate.tests.helpers (module), 30
  - transliterate.tests.test\_transliterate (module), 30
  - transliterate.utils (module), 35
  - transliterate\_function (in module transliterate.decorators), 34
  - transliterate\_method (in module transliterate.decorators), 34
  - TransliterateTest (class in transliterate.tests.test\_transliterate), 30
  - TranslitLanguagePack (class in transliterate.base), 32
- ## U
- UkrainianLanguagePack (class in transliterate.contrib.languages.uk.translit\_language\_pack), 29
  - update() (transliterate.backports.collections.Counter method), 22