
Typus Documentation

Release 0.2

Murad Byashimov

Nov 04, 2018

Contents

1	Web API	3
2	Installation	5
3	Usage	7
4	What it does	9
5	Documentation	11
6	Compatibility	13
7	Changelog	15
	7.1 0.2	15
	7.2 0.1	15
8	Contents	17
	8.1 What it's for?	17
	8.2 Processors	17
	8.3 Utils	18
9	Indices and tables	19

Typus is a typography tool. It means you can write text the way you use to and let it handle all that formatting headache:

```
"I don't feel very much like Pooh today..." said Pooh.  
"There there," said Piglet. "I'll bring you tea and honey until you do."  
- A.A. Milne, Winnie-the-Pooh
```

```
"I don't feel very much like Pooh today..." said Pooh.  
"There there," said Piglet. "I'll bring you tea and honey until you do."  
-- A.A. Milne, Winnie-the-Pooh
```

Copy & paste this example to your rich text editor. Result may depend on the font of your choice. For instance, there is a tiny non-breaking space between A. A. you can see with Helvetica:

**"I don't feel very much like Pooh today..." said Pooh.
"There there," said Piglet. "I'll bring you tea and honey until you do."
— A.A. Milne, Winnie-the-Pooh**

Try out the [demo](#).

CHAPTER 1

Web API

A tiny [web-service](#) for whatever legal purpose it may serve.

CHAPTER 2

Installation

```
$ pip install git+git://github.com/byashimov/typus.git#egg=typus
```


Currently Typus supports English and Russian languages only. But it doesn't mean it can't handle more. I'm quite sure it covers Serbian and Turkmen.

In fact, Typus doesn't make difference between languages. It works with text. If you use Cyrillic then only relative processors will affect that text. In another words, give it a try if your language is not on the list

Here is a short example:

```
>>> from typus import en_typus, ru_typus
...
>>> # Underscore is for nbsp in debug mode
>>> en_typus("Beautiful is better than ugly." (c) Tim Peters.', debug=True)
'"Beautiful is_better than ugly." @_Tim Peters.'
>>> # Cyrillic '' in '()'
>>> ru_typus('« , .» () .', debug=True)
'« , .» @_ .'

```

The only difference between `en_typus` and `ru_typus` are in quotes they set: `""` for English and `«„»` for Russian. Both of them handle mixed text and that is pretty awesome.

Typus is highly customizable. Not only quotes can be replaced but almost everything. For instance, if you don't use html tags you can skip `EscapeHtml` processor which makes your Typus a little faster.

CHAPTER 4

What it does

- Replaces regular quotes "foo 'bar' baz" with typographic pairs: “foo ‘bar’ baz”. Quotes style depends on language and your Typus configuration.
- Replaces regular dash foo - bar with mdash or ndash or minus. Depends on case: plain text, digit rage, phone nubers, etc.
- Replaces complex symbols such as (c) with unicode characters: ©. Cyrillic analogs are supported too.
- Replaces vulgar fractions 1/2 with unicode characters: ½.
- Turns multiply symbol to a real one: 3x3 becomes 3×3.
- Replaces quotes with primes: 2' 4" becomes 2′ 4″.
- Puts non-breaking spaces.
- Puts ruble symbol.
- Trims spaces at the end of lines.
- and much more.

CHAPTER 5

Documentation

Docs are hosted on readthedocs.org.

See also:

Oh, there is also an outdated Russian article I should not probably suggest, but since all docs are in English, this [link](#) might be quite helpful.

CHAPTER 6

Compatibility

Tested on Python 3.6, 3.7.

7.1 0.2

- Python 3.6 and higher are supported only. However should work on any 3.x branch. That's because 3.6 string formatting is used in tests to make them easier to read and write.
- `EnRuExpressions` is no longer a mixin but processor.
- Better, cleaner tests with `pytest`.
- Minor fixes and improvements.

7.2 0.1

- Initial release.

8.1 What it's for?

Well, when you write text you make sure it's grammatically correct. Typography is *an aesthetic* grammar. Everything you type should be typographed in order to respect the reader. For instance, when you write “you’re” you put *apostrophe* instead of *single quote*, because of the same reason you place dot at the end of sentence instead of comma, even though they look similar.

Unfortunately all typographic characters are well hidden in your keyboard layout which makes them almost impossible to use. Fortunately Typus can do that for you.

8.1.1 The anatomy

`typus.core.TypusCore` runs *Processors* to do the job which can be plugged in for desired configuration. Here is a quick example:

```
from typus.core import TypusCore
from typus.processors import EnQuotes

class MyTypus(TypusCore):
    processors = (EnQuotes, )

my_typus = MyTypus()
assert my_typus('"quoted text"') == '"quoted text'"
```

`typus.core.TypusCore` runs `typus.processors.EnQuotes` processor which improves *quotes* only.

8.2 Processors

Processors are the core of Typus. Multiple processors are nested and chained in one single function to do things which may depend on the result returned by inner processors. Say, we set `EscapeHtml` and `MyTrimProcessor`, this is

how it works:

```
extract html tags
    pass text further if condition is true
        do something and return
    return the text
put tags back and return
```

In python:

```
from typus.core import TypusCore
from typus.processors import BaseProcessor, EscapeHtml

class MyTrimProcessor(BaseProcessor):
    def run(self, text, **kwargs):
        # When processor is initiated it gets typus instance
        # as the first argument so you can access to it's configuration
        # any time
        if self.typus.trim:
            trimmed = text.strip()
        else:
            trimmed = text
        return self.run_other(trimmed, **kwargs)

class MyTypus(TypusCore):
    # This becomes a single function. EscapeHtml goes first
    processors = (EscapeHtml, MyTrimProcessor)

    # Set it to `False` to disable trimming
    trim = True

my_typus = MyTypus()
assert my_typus(' test ') == 'test'
```

8.2.1 Built-in processors

8.3 Utils

CHAPTER 9

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)