

# reStructuredText

[docutils.sf.net](http://docutils.sf.net)

Django-UserGroup Hamburg 04.08.2009

Arne Brodowski

[www.arnebrodowski.de](http://www.arnebrodowski.de)

# Was ist ReST?

- ReST ist eine Auszeichnungssprache (Markup)
- ReST soll in der Textform gut lesbar zu sein
- ReST soll gut in andere Formate konvertierbar sein (HTML, XML, LaTeX)

# Ein Beispiel

Eine Überschrift

=====

Lorem ipsum\_ dolor sit amet,  
consectetur **\*\*adipisicing\*\*** elit.

- Punkt 1
- Punkt 2

.. \_ipsum: <http://example.com>

# Wer benutzt ReST?

- Viele Python Projekte sind in ReST dokumentiert, u.a. Python selbst und Django.
- Oft anzutreffen in Python Docstrings

# ReST + Django

- Ein Template-Filter ist in `django.contrib.markup` schon enthalten.
- In der Praxis zeigt sich, dass das oft nicht reicht.

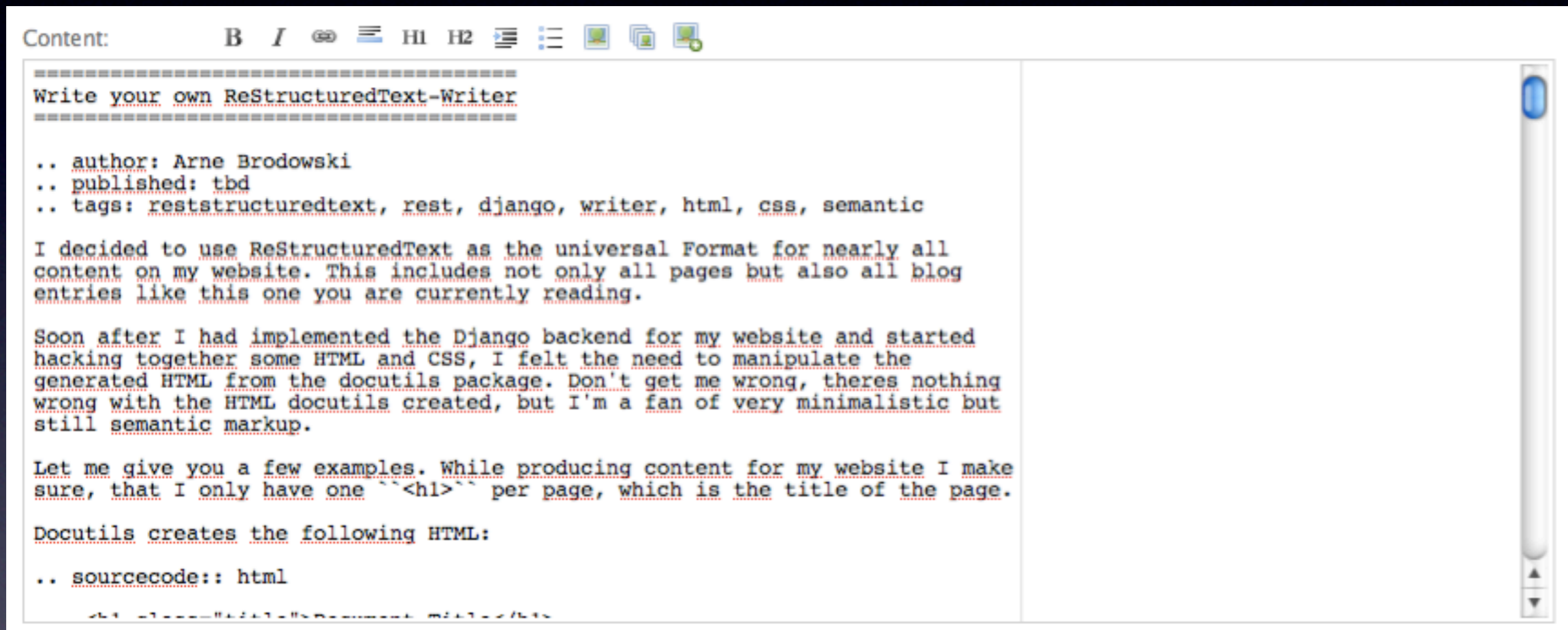
# Template

```
{% block content %}
```

```
{{ object.text|restructuredtext }}
```

```
{% endblock %}
```

# Administration



The screenshot shows a web editor window titled "Content:". The editor has a toolbar with icons for bold (B), italic (I), link, list, H1, H2, and other formatting options. The main content area displays a document with the following text:

```
=====
Write your own ReStructuredText-Writer
=====

.. author: Arne Brodowski
.. published: tbd
.. tags: reststructuredtext, rest, django, writer, html, css, semantic

I decided to use ReStructuredText as the universal Format for nearly all
content on my website. This includes not only all pages but also all blog
entries like this one you are currently reading.

Soon after I had implemented the Django backend for my website and started
hacking together some HTML and CSS, I felt the need to manipulate the
generated HTML from the docutils package. Don't get me wrong, theres nothing
wrong with the HTML docutils created, but I'm a fan of very minimalistic but
still semantic markup.

Let me give you a few examples. While producing content for my website I make
sure, that I only have one ``<h1>`` per page, which is the title of the page.

Docutils creates the following HTML:

.. sourcecode:: html
<h1 class="title">Document #111</h1>
```

<http://files.arnebrodowski.de/software/javascript/jquery.resteditor.js>

# Bilder einfügen

- JS Editor öffnet ein Popup Bildern
- Primary Key wird in den ReST-Text geschrieben
- Ein Templatetag ersetzt #<pk> mit dem richtigen Pfad bevor ReST zu HTML konvertiert wird.



# Die Ausgabe anpassen

```
from docutils.core import publish_parts
```

```
parts = publish_parts(source=value,  
                      writer=CleanHTMLWriter(),  
                      settings_overrides=docutils_settings)  
return parts[...]
```

# Writer & Translator

```
class CleanHTMLWriter(html4css1.Writer):
```

```
    def __init__(self):
```

```
        html4css1.Writer.__init__(self)
```

```
        self.translator_class = CleanHTMLTranslator
```

```
class CleanHTMLTranslator(html4css1.HTMLTranslator):
```

```
    ...
```

# Translator anpassen

```
class CleanHTMLTranslator(html4css1.HTMLTranslator):  
    def visit_container(self, node):  
        self.body.append(  
            self.starttag(node, 'div', CLASS='container'))  
  
    def depart_container(self, node):  
        self.body.append("\n</div>\n")
```

...

# Mehr Informationen

- [docutils.sf.net](http://docutils.sf.net)
- [sphinx.pocoo.org](http://sphinx.pocoo.org)
- [pygments.org](http://pygments.org)

The End