

---

# **read\_the\_docks\_test**

***Release 0.0.2***

**Paul K.**

**Jan 14, 2021**



**CONTENTS:**

- 1 Erstellen einer Doku für readTheDocks mit Sphinx: 3**
  - 1.1 Lokal: . . . . . 3
  - 1.2 Diese Seite als “raw code”: . . . . . 3
- 2 Bilder Integrieren 5**
  - 2.1 Bilder werden so hinzugefügt: . . . . . 5
- 3 Cheat Sheet 7**
  - 3.1 Section Structure . . . . . 7
  - 3.2 Body Elements . . . . . 7
  - 3.3 Inline Markup . . . . . 10
  - 3.4 Directive Quick Reference . . . . . 10
  - 3.5 Interpreted Text Role Quick Reference . . . . . 11
- Bibliography 13**



Link zum [Code](#) (GitLab).



## ERSTELLEN EINER DOKU FÜR READTHEDOCKS MIT SPHINX:

### 1.1 Lokal:

#### 1.1.1 (Python Environment)

- im Git-Repository:
- install sphinx:

```
pip install sphinx
```

```
sphinx-quickstart docs
```

- das “docs” erstellt automatisch einen Ordner “docs” in dem die Doku-Files und Make-Files (make.bat) liegen werden
  - ==> standards bzw. geeignete Einstellungen verwenden
- Um nun aus den .txt oder .rst (restructuredtext) Dateien die fertige Doku-Seite (i.d.R. html) zu machen verwende:

```
cd docs
make html

+ ==> das index.html file ist im /docs/build/html ordner verfügbar
      es kann z.B. mit Firefox geöffnet werden
```

- Nun im ReadTheDocs-Admin Center das Repo hinzufügen
- Dann WebHook im GitLab einrichten
- Branches usw. im Admin-Center verwalten

### 1.2 Diese Seite als “raw code”:

```
=====
Erstellen einer Doku für readTheDocks mit Sphinx:
=====
```

```
(Links to) StructuredText:
=====
```

```
* https://github.com/DevDungeon/reStructuredText-Documentation-Reference
```

(continues on next page)

(continued from previous page)

```
Lokal:
=====

(Python Environment)
-----

- im Git-Repository:
    * install sphinx:

.. code-block::

    pip install sphinx

* "docs" erstellt automatisch einen Ordner "docs" in dem die Doku-Files und Make-
↳Files (make.bat) liegen werden

.. code-block::

    sphinx-quickstart docs

* ==>
    + standards verwenden & Projektnamen (+ aktuelle Projektversion) eingeben_
↳ (... Min. 33:27 (kurz) oder 17:00 (lang - mit erklärung) .. )

* Um aus den .txt oder .rst (restructuredtext) files die fertige Doku-Seite (i.d.R._
↳html) zu machen:

.. code-block::

    cd docs
    make html

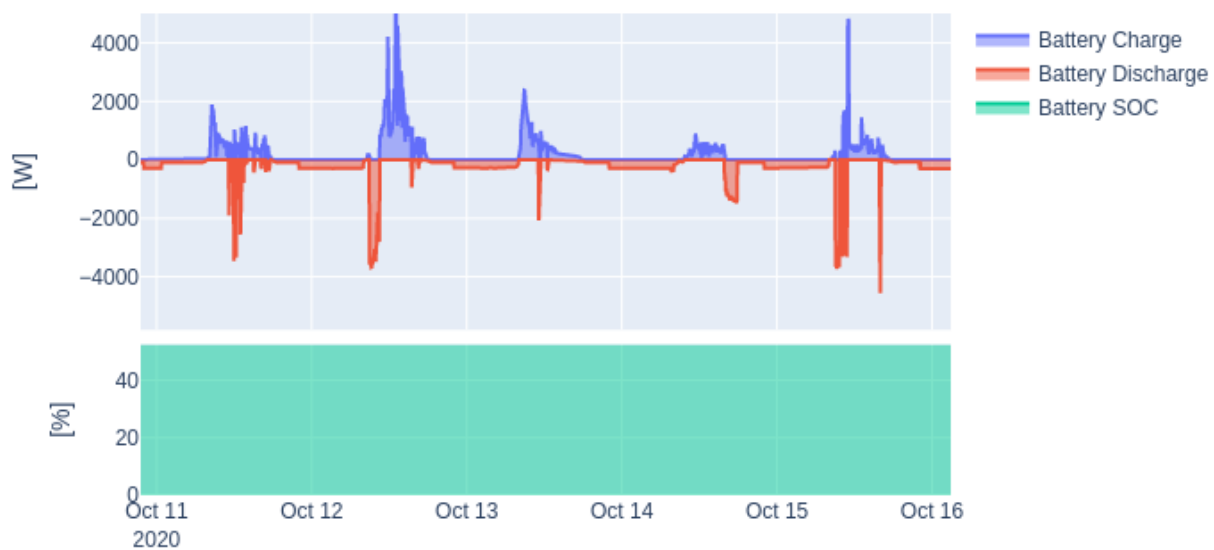
* make html wird auch automatisch von readthedocks ausgeführt - aber so können wir_
↳das Ergebnis lokal anschauen..

* ==>
    + das index.html file ist im /docs/build/html    ordner verfügbar und_
↳kann z.B. mit Firefox geöffnet werden
```



## BILDER INTEGRIEREN

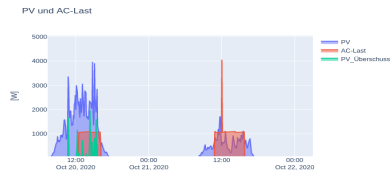
Leistung und State of Charge



## 2.1 Bilder werden so hinzugefügt:

```
.. image:: images/example2.png

.. oder komplizierter:
.. image:: images/example2.png
   :height: 100
   :width: 200
   :alt: alternate text
```



## CHEAT SHEET

The `reStructuredText` Cheat Sheet: Syntax Reminders

**Date** \$Date: 2013-02-20 02:10:53 +0100 (Mi, 20. Feb 2013) \$

**Revision** \$Revision: 7612 \$

**Description** This is a “docinfo block”, or bibliographic field list

---

**Note:** If you are reading this as HTML, please read [cheatsheet.txt](#) instead to see the input syntax examples!

---

### 3.1 Section Structure

Section titles are underlined or overlined & underlined.


### 3.2 Body Elements

Grid table:

Paragraphs are flush-left, separated by blank lines. Block quotes are indented.	Literal block, preceded by “::”:  Indented <b>or:</b>  > Quoted
<pre>&gt;&gt;&gt; print 'Doctest block' Doctest block</pre>	
Line blocks preserve line breaks & indents. [new in 0.3.6] Useful for addresses, verse, and adornment-free lists; long lines can be wrapped with continuation lines.	

Simple tables:

List Type	Examples (syntax in the <a href="#">text source</a> )
Bullet list	<ul style="list-style-type: none"><li>• items begin with “-“, “+”, or “*”</li></ul>
Enumerated list	<ol style="list-style-type: none"><li>1. items use any variation of “1.”, “A)”, and “(i)”</li><li>2. also auto-enumerated</li></ol>
Definition list	<b>Term is flush-left</b> [optional classifier] Definition is indented, no blank line between
Field list	<b>field name</b> field body
Option list	<b>-o</b> at least 2 spaces between option & description

Explicit Markup	Examples (visible in the <a href="#">text source</a> )
Footnote	
Citation	
Hyperlink Target	
Anonymous Target	
	
Directive (“::”)	
Substitution Def	
Comment	
Empty Comment	(“..” on a line by itself, with blank lines before & after, used to separate indentation contexts)

## 3.3 Inline Markup

*emphasis*; **strong emphasis**; *interpreted text*; *interpreted text with role*; inline literal text; standalone hyperlink, <http://docutils.sourceforge.net>; named reference, `reStructuredText`; anonymous reference; footnote reference,<sup>1</sup>; citation reference, `[CIT2002]`; like an inline directive; inline internal target.

## 3.4 Directive Quick Reference

See <http://docutils.sf.net/docs/ref/rst/directives.html> for full info.

Directive Name	Description (Docutils version added to, in [brackets])
attention	Specific admonition; also “caution”, “danger”, “error”, “hint”, “important”, “note”, “tip”, “warning”
admonition	Generic titled admonition: <code>.. admonition:: By The Way</code>
image	<code>.. image:: picture.png</code> ; many options possible
figure	Like “image”, but with optional caption and legend
topic	<code>.. topic:: Title</code> ; like a mini section
sidebar	<code>.. sidebar:: Title</code> ; like a mini parallel document
parsed-literal	A literal block with parsed inline markup
rubric	<code>.. rubric:: Informal Heading</code>
epigraph	Block quote with class=“epigraph”
highlights	Block quote with class=“highlights”
pull-quote	Block quote with class=“pull-quote”
compound	Compound paragraphs [0.3.6]
container	Generic block-level container element [0.3.10]
table	Create a titled table [0.3.1]
list-table	Create a table from a uniform two-level bullet list [0.3.8]
csv-table	Create a table from CSV data [0.3.4]
contents	Generate a table of contents
sectnum	Automatically number sections, subsections, etc.
header, footer	Create document decorations [0.3.8]
target-notes	Create an explicit footnote for each external target
math	Mathematical notation (input in LaTeX format)
meta	HTML-specific metadata
include	Read an external reST file as if it were inline
raw	Non-reST data passed untouched to the Writer
replace	Replacement text for substitution definitions
unicode	Unicode character code conversion for substitution defs
date	Generates today’s date; for substitution defs
class	Set a “class” attribute on the next element
role	Create a custom interpreted text role [0.3.2]
default-role	Set the default interpreted text role [0.3.10]
title	Set the metadata document title [0.3.10]

<sup>1</sup> Manually numbered or [#] auto-numbered (even [#labelled]) or [\*] auto-symbol

## 3.5 Interpreted Text Role Quick Reference

See <http://docutils.sf.net/docs/ref/rst/roles.html> for full info.

Role Name	Description
emphasis	Equivalent to <i>emphasis</i>
literal	Equivalent to <code>literal</code> but processes backslash escapes
math	Mathematical notation (input in LaTeX format)
PEP	Reference to a numbered Python Enhancement Proposal
RFC	Reference to a numbered Internet Request For Comments
raw	For non-reST data; cannot be used directly (see docs) [0.3.6]
strong	Equivalent to <b>strong</b>
sub	Subscript
sup	Superscript
title	Title reference (book, etc.); standard default role





## BIBLIOGRAPHY

[CIT2002] A citation.