
pyfoobar

Jul 29, 2019

Contents

1	sphinx	1
2	Sphinxmarkdown	3
3	markdown rst	5
3.1	5
4	rst, reST, reStructuredText, reST, standard reST markup	7
5		9
5.1	9
5.2	wheel	9
5.3	install	10
5.4	10
6		11
6.1	ep	11
6.2	11
7		13
7.1	reference	13
7.2	13
7.3	vscode	14
7.4	lint	14
7.5	print	14
7.6	test	14
7.7	ci	15
7.8	profile	15
7.9	API	15
8		17
8.1	inspect	17
8.2	functools	18
8.3	ABCMeta	18
8.4	util	18
9	(asyncio)	19

10 functional programming	21
10.1 generator	21
10.2 list comprehension, listcomp	21
10.3 built-in	21
10.4 libs	22
11	23
12	25
12.1 stdlib	25
12.2 extern lib	25
12.3 framework	25
12.4 math	26
12.5 net	26
12.6 frameworks	26
13 (frameworks)	27
13.1 RxPy	27
14 (Qt)	29
14.1 Qt.py	29
14.2 gevent	29
14.3 qconsole	29
15 (math)	31
15.1 numpy	31
15.2 multiprocessing	31
16 (net.Scrapy)	33
16.1 Scrapy shell	33
16.2 Other	33
17	35
17.1 Neo4j	35
17.2 sqlite	35
18	37
19 TODOS	39
20 rST	41
21 Indices and tables	43
Python Module Index	45
Index	47

CHAPTER 1

sphinx

```
http://openalea.gforge.inria.fr/doc/openalea/doc/_build/html/source/sphinx/rest_
↪syntax.html
```


CHAPTER 2

Sphinxmarkdown

This is my answer, taken from [StackOverflow](#)

example

document

You can use Markdown and reStructuredText in the same Sphinx project. How to do this is succinctly documented on [Read The Docs](#). Install `recommonmark` (`pip install recommonmark`) and then edit `conf.py`:

```
from recommonmark.parser import CommonMarkParser

source_parsers = {
    '.md': CommonMarkParser,
}

source_suffix = ['.rst', '.md']
```



```
class recommonmark.transform.AutoStructify(*args, **kwargs)
    Bases: docutils.transforms.Transform

    Automatically try to transform blocks to sphinx directives.

    This class is designed to handle AST generated by CommonMarkParser.
```

```
import recommonmark.transform
github_doc_root = 'https://github.com/rtfd/recommonmark/tree/master/doc/'
def setup(app):
    app.add_config_value('recommonmark_config', {
        'url_resolver': lambda url: github_doc_root + url,
        'auto_toc_tree_section': 'Contents',
    }, True)
    app.add_transform(recommonmark.transform.AutoStructify)
```

```
### ``eval_rst
### .. autoclass:: recommonmark.transform.AutoStructify
###    :show-inheritance:
### ``
```

- *Lang*
- *Libs*
- *Algo*

3.1

Line numbers and highlights

emphasis-lines: highlights the lines.

linenos: shows the line numbers as well.
caption: shown at the top of the code block.
name: may be referenced with *:ref:* later.

Listing 1: An example code-block with everything turned on.

```
1 # Comment line
2 import System
3 System.run_emphasis_line
4 # Long lines in code blocks create a auto horizontal scrollbar
5 System.exit!
```

rst, reST, reStructuredText, reST, standard reST markup

```
https://zh-sphinx-doc.readthedocs.io/en/latest/rest.html
```

English Version

```
http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
```

```
https://my-study-restructuredtext.readthedocs.io/en/latest/  
http://openalea.gforge.inria.fr/doc/openalea/doc/_build/html/source/sphinx/rest_  
→syntax.html
```

sphinx-quickstart

```
doc/content.rst  
sphinx-build docs/ docs/build  
http://www.sphinx-doc.org/en/stable/usage/quickstart.html
```

sphinx.ext.autodoc

```
https://www.sphinx-doc.org/en/master/usage/extensions/autodoc.html
```

5.1

:

```
distutils, setuptools, distribute, distutils, distlib
```

:

```
pbr, https://pypi.org/project/pbr/
```

5.2 wheel

5.2.1 jarwheel

```
# package a wheel
python setup.py bdist_wheel

# run cmd line from wheel
python wheel-0.21.xxx.whl/wheel -h

# another example
python foobar.zip
# eq
PYTHONPATH=foobar.zip python -m __main__

# another example
python foobar.zip/mymod
PYTHONPATH=foobar.zip python -m mymod.__main__
```

5.2.2

```
python setup.py bdist_wheel upload -r testpypi
```

5.3 install

easy_install, Egg

5.3.1 pip

piphttps://pip.readthedocs.io/en/stable/user_guide

```
# installpi
pip install numpy --timeout 60
pip install -i https://testpypi.python.org/pypi ceilometer
```

```
pip freeze
```

5.4

```
setup.py sdist
python setup.py register -r testpypi
python setup.py sdist upload -r testpypi
```

6.1 ep

Entry points are a simple way for distributions to “advertise” Python objects (such as functions or classes) for use by other distributions.

6.1.1 entry_point_inspector:

```
epi group list
epi group show gui_scripts
epi ep show gui_scripts spyder
```

6.1.2 pkg_resources

6.2

6.2.1 stevedore

stevedore

Python makes loading code dynamically easy, allowing you to configure and extend your application by discovering and loading extensions (“plugins”) at runtime. Many applications implement their own library for doing this, using **import** or `importlib`. `stevedore` avoids creating yet another extension mechanism by building on top of `setuptools` entry points. The code for managing entry points tends to be repetitive, though, so `stevedore` provides manager classes for implementing common patterns for using dynamically loaded extensions.

Pythonstevedore

<http://yansu.org/2013/06/09/learn-python-stevedore-module-in-detail.html>

7.1 reference

<https://github.com/jobbole/awesome-python-cn>

7.2

7.2.1 virtual env

```
virtualenv venv --system-site-package
```

7.2.2 pip

config windows: C:\Users\tony\AppData\Roaming\pip\pip.ini

```
[global]
timeout = 60
index-url = https://pypi.tuna.tsinghua.edu.cn/simple

trusted-host = pypi.python.org
               pypi.org
               files.pythonhosted.org
               pypi.tuna.tsinghua.edu.cn
```

7.3 vscode

7.3.1 config for windows

```
{
  "cmake-tools-helper.cmake_download_path": "c:\\Users\\tony\\.
↪vscode\\extensions\\maddouri.cmake-tools-helper-0.2.1\\cmake_download",
  "workbench.startupEditor": "newUntitledFile",
  "workbench.colorTheme": "Electron Highlighter",
  "terminal.integrated.shell.windows": "C:\\WINDOWS\\System32\\cmd.exe",
  "terminal.integrated.shellArgs.windows" : ["/K",
↪ "C:\\Users\\tony\\Anaconda3\\Scripts\\activate.bat", "C:\\Users\\tony\\Anaconda3"],
  "git.enableSmartCommit": true,
  "git.autofetch": true,
  "editor.suggestSelection": "first",
  "vsintellicode.modify.editor.suggestSelection": "automaticallyOverrodeDefaultValue
↪",
  "python.jediEnabled": false
}
```

7.3.2 config for linux

linux path: /home/tony/.pip/pip.conf

```
[global]
index-url = http://mirrors.aliyun.com/pypi/simple/

[install]
trusted-host=mirrors.aliyun.com
```

7.4 lint

7.5 print

```
from pprint import pprint
```

7.6 test

7.6.1 unit test

test_

7.6.2 nosetests

7.6.3 tox

7.7 ci

TODO

gerrit JenkinsCI Travis Zuul: Sonar

7.8 profile

```
examples.profile.use_cprofile.note()
python -m cProfile -s time myscript.py
KCacheGrindValgrind
pyprof2calltreeProfileValgrind -o
pypyCPython
python -m memory_profiler script.py
memory_profiler
using memoryview
time.perf_counter()
timeit
profile
```

```
examples.profile.use_cprofile.x()
```

```
>>> dis.dis(x)
dispython
```

7.9 API

sphinx

```
.. deprecated:: 1.1
```

warnings warnings.warn()

```
warnings.warn("turn_left is deprecated sin Version 1.1, use trun instead")
```

debtcollector

```
from debtcollector import moves
@move.moved_method("turn", version="1.1")
```



```
class examples.lang.slots.Point (x,y)
    Bases: object

    x
    y

examples.lang.slots.use_namedtuple ()
    collections.namedtuple("Foobar", ["x", "y"])

class examples.lang.api.UseDebtcollector
    Bases: object
```

<https://docs.openstack.org/debtcollector/latest/user/index.html>

A collection of Python deprecation patterns and strategies that help you collect your technical debt in a non-destructive manner. The goal of this library is to provide well documented developer facing deprecation patterns that start of with a basic set and can expand into a larger set of patterns as time goes on. The desired output of these patterns is to apply the warnings module to emit DeprecationWarning or PendingDeprecationWarning or similar derivative to developers using libraries (or potentially applications) about future deprecations.

```
trun (direction='left')
```

```
trun_left ()
```

Deprecated since version 1.1.

8.1 inspect

inspect - Get useful information from live Python objects.

8.2 functools

8.3 ABCMeta

sixPythonPython 2.x Python 3.x @six.add_metaclass(MetaClass) PythonmetaclassPythonabcsix
@six.add_metaclass(abc.ABCMeta)

```
import six
import abc

@six.add_metaclass(abc.ABCMeta)
class PluginBase(object) :
    pass
```

8.3.1 super(), mro()

mro method resolution order

8.4 util

8.4.1 print

```
num = 1
print(f"coroutine_{num} start")
```

8.4.2 import

examples.lang.import.**import_module**()
import module from module file

examples.lang.import.**import_source**()
import from source file

8.4.3 function

examples.lang.fun.**func** (a: int, b: dict) → set
function with annotation

CHAPTER 9

(asyncio)

CHAPTER 10

functional programming

10.1 generator

`inspect.isgeneratorfunction()` `inspect.getgeneratorstate()`

`genexpr`

```
l = [1,2,3,4]
g = (i for i in l)
```

10.2 list comprehension, listcomp

```
[ str(i) for i in [1,2,4]]
{ x:x.upper() for x in ["hello", "world"]}
{ x.upper() for x in ["hello", "world"]}
```

10.3 built-in

```
map(func, iterable)
filter(func or None, iterable)
enumerate(iterable[,start])
sorted(iterable, key=None, reverse=False)
any(iterable)
all(iterable)
zip(iter1[,iter2])
```

10.4 libs

10.4.1 first

```
from first import first
first([-1,0,1,2], key=lambda x: x>0)

def greater_than_zero(n):
    return n > 0

first(iterable, key = greater_than_zero)
```

10.4.2 functools

['RLock', 'WRAPPER_ASSIGNMENTS', 'WRAPPER_UPDATES', 'cmp_to_key', 'get_cache_token', 'lru_cache', 'namedtuple', 'partial', 'partialmethod', 'recursive_repr', 'reduce', 'singledispatch', 'total_ordering', 'update_wrapper', 'wraps']

```
import functools
from first import first

def greater_than(number, min=0):
    return number > min

first(iterable, key = functools.partial(greater_than, min = 42))
```

10.4.3 operator

['abs', 'add', 'and_', 'attrgetter', 'concat', 'contains', 'countOf', 'delitem', 'eq', 'floordiv', 'ge', 'getitem', 'gt', 'iadd', 'iand', 'iconcat', 'ifloordiv', 'ilshift', 'imatmul', 'imod', 'imul', 'index', 'indexOf', 'inv', 'invert', 'ior', 'ipow', 'irshift', 'is_', 'is_not', 'isub', 'itemgetter', 'itrueidiv', 'ixor', 'le', 'length_hint', 'lshift', 'lt', 'matmul', 'methodcaller', 'mod', 'mul', 'ne', 'neg', 'not_', 'or_', 'pos', 'pow', 'rshift', 'setitem', 'sub', 'trueidiv', 'truth', 'xor']

```
import operator
import functools import partial
import first
frist(iterable, key = partial(operator.le, 0))
```

10.4.4 itertools

['accumulate', 'chain', 'combinations', 'combinations_with_replacement', 'compress', 'count', 'cycle', 'dropwhile', 'filterfalse', 'groupby', 'islice', 'permutations', 'product', 'repeat', 'starmap', 'takewhile', 'tee', 'zip_longest']

CHAPTER 11

`examples.algorithm.use_collection()`

set add,clear,copy,difference,difference_update,discard,intersection, intersection_update, isdisjoint,issubset,issuperset,pop,remove, symmetric_difference,symmetric_difference_update,union,update

dict clear,copy,fromkeys,get,items,keys,pop,popitem,setdefault,update,values

list append,clear,copy,count,extend,index,insert,pop,remove,reverse,sort

tuple count,index

`examples.algorithm.use_bisect.use_bisect()`

bisect

blistbintree

`examples.algorithm.use_defaultdict.add_animal_in_family(species, animal, family)`
`collections.defaultdict`

`examples.algorithm.memoization.memoized_sin`
`functools.lru_cache(maxsize)`

12.1 stdlib

12.1.1 os

```
import os
# Determine path to this module
OPENSOT_PATH = os.path.dirname(os.path.realpath(__file__))
```

12.1.2 datetime

```
datetime, pytz, ios8601
```

12.2 extern lib

pythonAPI

gui-lib

12.3 framework

12.3.1 gevent

examples.libs.use_gevent.**asyncfun**()
start a group async task

examples.libs.use_gevent.**joinall**()

examples.libs.use_gevent.**use_Actor**()
Actor: a higher level concurrency model popularized by the language Erlang

```
examples.libs.use_gevent.use_Greenlet()  
    Greenlet : thread  
examples.libs.use_gevent.use_zmq()  
    gevent-zmq
```

12.4 math

math

12.5 net

net

12.6 frameworks

framework

13.1 RxPy

`examples.frm.rxpy.userx()`
document <https://rxpy.readthedocs.io/en/latest/>
reactive programming in python: <https://auth0.com/blog/reactive-programming-in-python/>
ReactiveX : <http://reactivex.io/>
observable, observer/subscribe, subject
data stream
toolbox = filter, create, transform, unify
event-driven programming: handling any event to trigger the corresponding action
reactive: wrap data into the reactive system as events

14.1 Qt.py

14.1.1

<https://github.com/codito/pyqtkeybind>

14.2 gevent

14.3 qconsole

```
class examples.libs.qt.iconsole.MainWindow
```

```
    Bases: PyQt5.QtWidgets.QMainWindow
```

```
    A window that contains a single Qt console.
```

```
    shutdown_kernel()
```

```
examples.libs.qt.iconsole.make_jupyter_widget_with_kernel()
```

```
    Start a kernel, connect to it, and create a RichJupyterWidget to use it
```

```
examples.libs.qt.inprocess.show()
```

```
    An example of embedding a RichJupyterWidget with an in-process kernel. We recommend using a kernel in a separate process as the normal option - see embed_qtconsole.py for more information. In-process kernels are not well supported. To run this example:
```

```
    python3 inprocess_qtconsole.py
```


CHAPTER 15

(math)

15.1 numpy

15.2 multiprocessing

16.1 Scrapy shell

testing XPath or CSS expressions

scrapy.cfg:

```
[settings]
shell = bpython
```

```
from scrapy.shell import inspect_response
inspect_response(response, self)
```

<https://docs.scrapy.org/en/latest/topics/extensions.html#debugger-extension>

```
class scrapy.extensions.debug.Debugger
Invokes a Python debugger inside a running Scrapy process when a SIGUSR2 signal is
↳received. After the debugger is exited, the Scrapy process continues running
↳normally.
```

16.2 Other

16.2.1 ScreenShot

Splash - A javascript rendering service Splash is a javascript rendering service. It's a lightweight web browser with an HTTP API, implemented in Python 3 using Twisted and QT5. The (twisted) QT reactor is used to make the service fully asynchronous allowing to take advantage of webkit concurrency via QT main loop.

CHAPTER 17

ORM RMDB NoSQL

17.1 Neo4j

<http://neo4j.com.cn/public/docs/index.html>

17.2 sqlite

CHAPTER 18

- Full Stack Python

<https://www.fullstackpython.com/>

- Nodeclub

Node.js MongoDB <http://cnodejs.org/> <https://github.com/cnodejs/nodeclub/>

- Neo4j

<http://neo4j.com.cn/>

CHAPTER 19

TODOS

CHAPTER 20

rST

:

```
https://github.com/zhaojiedi1992/My_Study_ReStructuredText
```

:

```
=====
=====
=====
-----
-----
-----
\n\n\n\n\n
\n\n\n\n\n
\n\n\n\n\n
```


CHAPTER 21

Indices and tables

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

e

- `examples.algorithm`, [23](#)
- `examples.algorithm.memoization`, [23](#)
- `examples.algorithm.use_bisect`, [23](#)
- `examples.algorithm.use_defaultdict`, [23](#)
- `examples.frm.rxpy`, [27](#)
- `examples.lang.api`, [17](#)
- `examples.lang.fun`, [18](#)
- `examples.lang.import`, [18](#)
- `examples.lang.slots`, [17](#)
- `examples.libs.qt.iconsole`, [29](#)
- `examples.libs.qt.inprocess`, [29](#)
- `examples.libs.use_gevent`, [25](#)
- `examples.profile.use_cprofile`, [15](#)

A

`add_animal_in_family()` (in module *examples.algorithm.use_defaultdict*), 23
`asyncfun()` (in module *examples.libs.use_gevent*), 25
`AutoStructify` (class in *recommonmark.transform*), 5

E

`examples.algorithm` (module), 23
`examples.algorithm.memoization` (module), 23
`examples.algorithm.use_bisect` (module), 23
`examples.algorithm.use_defaultdict` (module), 23
`examples.frm.rxpy` (module), 27
`examples.lang.api` (module), 17
`examples.lang.fun` (module), 18
`examples.lang.import` (module), 18
`examples.lang.slots` (module), 17
`examples.libs.qt.iconconsole` (module), 29
`examples.libs.qt.inprocess` (module), 29
`examples.libs.use_gevent` (module), 25
`examples.profile.use_cprofile` (module), 15

F

`func()` (in module *examples.lang.fun*), 18

I

`import_module()` (in module *examples.lang.import*), 18
`import_source()` (in module *examples.lang.import*), 18

J

`joinall()` (in module *examples.libs.use_gevent*), 25

M

`MainWindow` (class in *examples.libs.qt.iconconsole*), 29

`make_jupyter_widget_with_kernel()` (in module *examples.libs.qt.iconconsole*), 29
`memoized_sin` (in module *examples.algorithm.memoization*), 23

N

`note()` (in module *examples.profile.use_cprofile*), 15

P

`Point` (class in *examples.lang.slots*), 17

S

`show()` (in module *examples.libs.qt.inprocess*), 29
`shutdown_kernel()` (*examples.libs.qt.iconconsole.MainWindow* method), 29

T

`trun()` (*examples.lang.api.UseDebtcollector* method), 17
`trun_left()` (*examples.lang.api.UseDebtcollector* method), 17

U

`use_Actor()` (in module *examples.libs.use_gevent*), 25
`use_bisect()` (in module *examples.algorithm.use_bisect*), 23
`use_collection()` (in module *examples.algorithm*), 23
`use_Greenlet()` (in module *examples.libs.use_gevent*), 26
`use_namedtuple()` (in module *examples.lang.slots*), 17
`use_zmq()` (in module *examples.libs.use_gevent*), 26
`UseDebtcollector` (class in *examples.lang.api*), 17
`userx()` (in module *examples.frm.rxpy*), 27

X

`x` (*examples.lang.slots.Point* attribute), 17

`x()` (in module *examples.profile.use_cprofile*), [15](#)

Y

`y` (*examples.lang.slots.Point* attribute), [17](#)