

# **My Project**

**version test**

**Author Name**

**July 07, 2015**



# Contents

<b>Sphinx Test Document</b>	<b>1</b>
Module-specific markup	1
Inline Markup	1
Substitutions	2
Paragraph-level markup	2
Syntax Highlighting	2
Misc Markup	3
Index-Generating Markup	3
Glossary	4
<b>Indices and tables</b>	<b>5</b>
<b>Index</b>	<b>7</b>
<b>Python Module Index</b>	<b>9</b>



# Sphinx Test Document

This is a test document

## Module-specific markup

This document tests sphinx markup from section "Module-specific markup" in the sphinx docs, using the snippets from the manual.

The rst2pdf output should be a hybrid between the LaTeX and HTML outputs.

```
parrot.spam (eggs)
parrot.ham (eggs)
    Spam or ham the foo.
```

```
Timer.repeat ([, repeat=3[, number=1000000]])
```

```
class parrot.Foo
```

```
    quux ()
```

```
class parrot.Bar
```

```
    Bar.quux ()
```

```
parrot.compile (source[, filename[, symbol]])
```

```
parrot.compile (source[, filename, symbol])
```

```
parrot.compile (source : string[, filename, symbol]) → ast object
```

```
parrot.format_exception (etype, value, tb[, limit=None])
```

Format the exception with a traceback.

### Parameters:

- **etype** -- exception type
- **value** -- exception value
- **tb** -- traceback object
- **limit** (*integer or None*) -- maximum number of stack frames to show

**Return type:** list of strings

```
-m <module>, --module <module>
```

Run a module as a script.

```
-r
```

Work recursively.

```
-r revision
```

Specify the revision to work upon.

**rm -r** should refer to the first option, while **svn -r** should refer to the second one.

```
opcode
```

Describes a Python bytecode instruction.

## Inline Markup

**parrot** should link to the parrot module

**parrot.ham()** hams the eggs and links to the definition

**:cfunc:PyType\_GenericAlloc()** is C and links to the definition

**ls** is a OS-level command

Ctrl-C interrupts

From: `ralsina@kde.org` is my email

`ls(1)` explains `ls`

Start ■ Programs If you don't have an arrow between start and programs, your font doesn't have `u2023`

`text/plain`

`comp.os.advocacy.linux`

**notepad**

`^ERR`

`term1` and `term2` link to the glossary. so does `Term number 1`

## Substitutions

Release: test

Version: test

Today: July 07, 2015

## Paragraph-level markup

### Note

This function is not suitable for sending spam e-mails.

New in version 2.5: *New in version 2.5:* The *spam* parameter.

Changed in version 2.5: *Changed in version 2.5:* The *spam* parameter.

Seealso

**Module `zipfile`**

Documentation of the `zipfile` standard module.

**GNU tar manual, Basic Tar Format**

Documentation for tar archive files, including GNU tar extensions.

### LICENSE AGREEMENT

- A list of
- short items
- that should be
- displayed
- horizontally

```
try_stmt ::= try1_stmt | try2_stmt
try1_stmt ::= "try" ":" suite
              ("except" [expression ["," target]] ":" suite)+
              ["else" ":" suite]
              ["finally" ":" suite]
try2_stmt ::= "try" ":" suite
              "finally" ":" suite
```

## Syntax Highlighting

```
"Some code without linenos."
```

```
1 "Some code with linenos."
```

```
"This has less than 5 lines, so no linenos"
```

```
1 "This has more than 5 lines, so it has linenos"
2 "This has more than 5 lines, so it has linenos"
3 "This has more than 5 lines, so it has linenos"
4 "This has more than 5 lines, so it has linenos"
5 "This has more than 5 lines, so it has linenos"
6 "This has more than 5 lines, so it has linenos"
7 "This has more than 5 lines, so it has linenos"
```

```
"This is before a highlight:: none, so it has highlight"
```

```
"This is after a highlight:: none, so no highlight"
```

```
# -*- coding: utf-8 -*-
# Sphinx markup documentation build configuration file, created by
#
# All configuration values have a default; values that are commented out
# serve to show the default.

import sys, os

# If extensions (or modules to document with autodoc) are in another directory,
# add these directories to sys.path here. If the directory is relative to the
# documentation root, use os.path.abspath to make it absolute, like shown here.
#sys.path.append(os.path.abspath('.'))
```

## Misc Markup

Section author: Roberto Alsina <[alsina@netmanagers.com.ar](mailto:alsina@netmanagers.com.ar)>

This shows in PDF, not in HTML

## Index-Generating Markup

**term1**

A term in 1st place

**term2**

A term in 2nd place

**term1** and **term2** link to the glossary

In the Index, there should be several entries pointed here, generated by this:

```
.. index::
   single: execution; context
   module: __main__
   module: sys
   triple: module; search; path
```

## Glossary

This is a section called glossary. It should not be confused with the glossary up there (Issue 190)

THis link: **glossary** should link to the glossary, not the section called glossary.



## Indices and tables

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



# Index

[\\_\\_main\\_\\_](#)  
module

## *E*

**execution**  
context

## *M*

**module**  
[\\_\\_main\\_\\_](#)  
search path  
sys

## *P*

**path**  
module search

## *S*

**search**  
path, module  
**sys**  
module

## *T*

term1  
term2



# Python Module Index

## *p*

### **parrot** (Unix, Windows)

Analyze and reanimate dead parrots.