

xverbatim

display source code or execute it

一个显示代码或者执行代码的环境

万泽¹ | wanze²

版本： 1.0

¹作者：湖南常德人氏

²编者：wanze. 邮箱：a358003542@gmail.com。

开头说的话

开头说的话

目 录

开头说的话	i
目录	ii
1 章节开始	1
2 bash 语言	2
3 python 语言	3
4 c 语言	4
5 lisp 语言	5
6 tex 语言	6
7 php 语言	9
8 html 语言	10
9 java 语言	11
10 perl 语言	12
11 后缀名总览 txt mode	13

章节开始

bash 语言

```
1 a=1
2 echo $a
```

1

python 语言

```
1 for i in range(1, 5):  
2     print (i)  
3 else:  
4     print ("The for loop is over")
```

1

2

3

4

The for loop is over

C 语言

```
1#include <stdio.h>
2main()
3{
4printf ("Hello World!\n");
5}
```

Hello World!

lisp 语言

```
1 (format t "hello")
```

hello

tex 语言

```
1 % !Mode:: "TeX:UTF-8"% 確保文檔 utf-8 編碼
2 % 新加入的命令如下: addchtoc addsectoc reduline printendnotes hlabel
3 % 新加入的環境如下: common-format fig scalefig
4
5 \documentclass[11pt,oneside]{book}
6 \newlength{\textpt}
7 \setlength{\textpt}{11pt}
8 \newif\ifphone
9 \phonefalse
10
11 \usepackage{myconfig}
12 \usepackage{mytitle}
13
14 \begin{document}
15 \frontmatter
16
17 \titlea{書籍}
18 \titleb{使用\LaTeX 排版}
19 \titlec{一種良好的風格}
```

```
20 \author{作者}
21 \authorinfo{作者: }
22 \editor{编者}
23 \email{a358003542@gmail.com}
24 \editorinfo{编者: }
25 \version{1.0}
26 \titleLC
27
28 \addchtoc{开头说的话}
29 \chapter*{开头说的话}
30 \begin{common-format}
31 开头说的话
32
33 % 这里空一行。
34
35 \end{common-format}
36
37
38 \addchtoc{目录}
39 \setcounter{tocdepth}{2}
40 \tableofcontents
41
42 \begin{common-format}
43 \mainmatter
44
45 \chapter{章节开始}
46
```

47 开始写作。。。

48

49 % 这里空一行

50

51 \end{common-format}

52 \end{document}

php 语言

```
1 <?php
2 echo "<html>
3 <head>
4 <title>Hello World Page</title>
5 </head>
6 <body>
7 Hello World!
8 </body>
9 </html>"
10 ?>
```

html 语言

```
1 <HTML>
2 <HEAD>
3 <TITLE>Hello, World Page!</TITLE>
4 </HEAD>
5
6 <BODY>
7 Hello, World!
8 </BODY>
9 </HTML>
```

java 语言

```
1 class HelloWorld {  
2     public static void main (String args[]) {  
3         for (;;) {  
4             System.out.print("Hello World ");  
5         }  
6     }  
7 }
```

perl 语言

```
1 print "Hello, World!\n" while (1);
```

后缀名总览 txt mode

```
1 Pygments version 1.6, (c) 2006-2013 by Georg Brandl.
2
3 Lexers:
4 ~~~~~
5 * Clipper, XBase:
6     FoxPro (filenames *.PRG, *.prg)
7 * Cucumber, cucumber, Gherkin, gherkin:
8     Gherkin (filenames *.feature)
9 * RobotFramework, robotframework:
10    RobotFramework (filenames *.txt, *.robot)
11 * abap:
12    ABAP (filenames *.abap)
13 * ada, ada95ada2005:
14    Ada (filenames *.adb, *.ads, *.ada)
15 * ahk:
16    autohotkey (filenames *.ahk, *.ahkl)
17 * antlr-as, antlr-actionscript:
18    ANTLR With ActionScript Target (filenames *.G, *.g)
19 * antlr-cpp:
```



```
20     ANTLR With CPP Target (filenames *.G, *.g)
21 * antlr-csharp, antlr-c#:
22     ANTLR With C# Target (filenames *.G, *.g)
23 * antlr-java:
24     ANTLR With Java Target (filenames *.G, *.g)
25 * antlr-objc:
26     ANTLR With ObjectiveC Target (filenames *.G, *.g)
27 * antlr-perl:
28     ANTLR With Perl Target (filenames *.G, *.g)
29 * antlr-python:
30     ANTLR With Python Target (filenames *.G, *.g)
31 * antlr-ruby, antlr-rb:
32     ANTLR With Ruby Target (filenames *.G, *.g)
33 * antlr:
34     ANTLR
35 * apacheconf, aconf, apache:
36     ApacheConf (filenames .htaccess,
37     apache.conf, apache2.conf)
38 * applescript:
39     AppleScript (filenames *.applescript)
40 * as, actionscript:
41     ActionScript (filenames *.as)
42 * as3, actionscript3:
43     ActionScript 3 (filenames *.as)
44 * aspectj:
45     AspectJ (filenames *.aj)
46
```

```

47 * aspx-cs:
48     aspx-cs (filenames *.aspx, *.asax,
49         *.ascx, *.ashx, *.asmx, *.axd)
50 * aspx-vb:
51     aspx-vb (filenames *.aspx, *.asax,
52         *.ascx, *.ashx, *.asmx, *.axd)
53 * asy, asymptote:
54     Asymptote (filenames *.asy)
55 * autoit, Autoit:
56     AutoIt (filenames *.au3)
57 * awk, gawk, mawk, nawk:
58
59     Awk (filenames *.awk)
60 * basemake:
61     Base Makefile
62 * bash, sh, ksh:
63     Bash (filenames *.sh, *.ksh, *.bash, *.ebuild,
64         *.eclass, .bashrc, bashrc, .bash_*, bash_*)
65 * bat:
66     Batchfile (filenames *.bat, *.cmd)
67 * bbcode:
68     BBCode
69 * befunge:
70     Befunge (filenames *.befunge)
71 * blitzmax, bmax:
72     BlitzMax (filenames *.bmx)
73 * boo:

```

```
74     Boo (filenames *.boo)
75 * brainfuck, bf:
76     Brainfuck (filenames *.bf, *.b)
77 * bro:
78     Bro (filenames *.bro)
79 * bugs, winbugs, openbugs:
80     BUGS (filenames *.bug)
81 * c-objdump:
82     c-objdump (filenames *.c-objdump)
83 * c:
84     C (filenames *.c, *.h, *.idc)
85 * ca65:
86     ca65 (filenames *.s)
87 * cbmbas:
88     CBM BASIC V2 (filenames *.bas)
89 * ceylon:
90     Ceylon (filenames *.ceylon)
91 * cfengine3, cf3:
92     CFEngine3 (filenames *.cf)
93 * cfm:
94     Coldfusion HTML (filenames *.cfm, *.cfml, *.cfc)
95 * cfs:
96     cfstatement
97 * cheetah, spitfire:
98     Cheetah (filenames *.tmpl, *.spt)
99 * clojure, clj:
100    Clojure (filenames *.clj)
```

```

101 * cmake:
102     CMake (filenames *.cmake, CMakeLists.txt)
103 * cobol:
104     COBOL (filenames *.cob, *.COB, *.cpy, *.CPY)
105 * cobolfree:
106     COBOLFree (filenames *.cbl, *.CBL)
107 * coffee-script, coffeescript:
108     CoffeeScript (filenames *.coffee)
109 * common-lisp, cl:
110     Common Lisp (filenames *.cl, *.lisp, *.el)
111 * console:
112     Bash Session (filenames *.sh-session)
113 * control:
114     Debian Control file (filenames control)
115 * coq:
116     Coq (filenames *.v)
117 * cpp, c++:
118     C++ (filenames *.cpp, *.hpp, *.c++, *.h++, *.cc, *.hh, *.cxx, *.hxx, *.C, *.H, *.cp, *.
119 * cpp-objdump, c++-objdump, cxx-objdump:
120     cpp-objdump (filenames *.cpp-objdump, *.c++-objdump, *.cxx-objdump)
121 * croc:
122     Croc (filenames *.croc)
123 * csharp, c#:
124     C# (filenames *.cs)
125 * css+django, css+jinja:
126     CSS+Django/Jinja
127 * css+erb, css+ruby:

```

```
128     CSS+Ruby
129 * css+genshitext, css+genshi:
130     CSS+Genshi Text
131 * css+lasso:
132     CSS+Lasso
133 * css+mako:
134     CSS+Mako
135 * css+mako:
136     CSS+Mako
137 * css+myghty:
138     CSS+Myghty
139 * css+php:
140     CSS+PHP
141 * css+smarty:
142     CSS+Smarty
143 * css:
144     CSS (filenames *.css)
145 * cuda, cu:
146     CUDA (filenames *.cu, *.cuh)
147 * cython, pyx:
148     Cython (filenames *.pyx, *.pxd, *.pxi)
149 * d-objdump:
150     d-objdump (filenames *.d-objdump)
151 * d:
152     D (filenames *.d, *.di)
153 * dart:
154     Dart (filenames *.dart)
```

```
155 * delphi, pas, pascal, objectpascal:
156     Delphi (filenames *.pas)
157 * dg:
158     dg (filenames *.dg)
159 * diff, udiff:
160     Diff (filenames *.diff, *.patch)
161 * django, jinja:
162     Django/Jinja
163 * dpatch:
164     Darcs Patch (filenames *.dpatch, *.darcspatch)
165 * dtd:
166     DTD (filenames *.dtd)
167 * duel, Duel Engine, Duel View, JBST, jbst, JsonML+BST:
168     Duel (filenames *.duel, *.jbst)
169 * dylan-console, dylan-repl:
170     Dylan session (filenames *.dylan-console)
171 * dylan-lid, lid:
172     DylanLID (filenames *.lid, *.hdp)
173 * dylan:
174     Dylan (filenames *.dylan, *.dyl, *.intr)
175 * ec:
176     eC (filenames *.ec, *.eh)
177 * ecl:
178     ECL (filenames *.ecl)
179 * elixir, ex, exs:
180     Elixir (filenames *.ex, *.exs)
181 * erb:
```

```
182     ERB

183 * erl:

184     Erlang erl session (filenames *.erl-sh)

185 * erlang:

186     Erlang (filenames *.erl, *.hrl, *.es, *.escript)

187 * evoque:

188     Evoque (filenames *.evoque)

189 * factor:

190     Factor (filenames *.factor)

191 * fan:

192     Fantom (filenames *.fan)

193 * fancy, fy:

194     Fancy (filenames *.fy, *.fancypack)

195 * felix, flx:

196     Felix (filenames *.flx, *.flxh)

197 * fortran:

198     Fortran (filenames *.f, *.f90, *.F, *.F90)

199 * fsharp:

200     FSharp (filenames *.fs, *.fsi)

201 * gas:

202     GAS (filenames *.s, *.S)

203 * genshi, kid, xml+genshi, xml+kid:

204     Genshi (filenames *.kid)

205 * genshitext:

206     Genshi Text

207 * glsl:

208     GLSL (filenames *.vert, *.frag, *.geo)
```

```
209 * gnuplot:
210     Gnuplot (filenames *.plot, *.plt)
211 * go:
212     Go (filenames *.go)
213 * gooddata-cl:
214     GoodData-CL (filenames *.gdc)
215 * gosu:
216     Gosu (filenames *.gs, *.gsx, *.gsp, *.vark)
217 * groff, nroff, man:
218     Groff (filenames *.[1234567], *.man)
219 * groovy:
220     Groovy (filenames *.groovy)
221 * gst:
222     Gosu Template (filenames *.gst)
223 * haml, HAML:
224     Haml (filenames *.haml)
225 * haskell, hs:
226     Haskell (filenames *.hs)
227 * haxml, hxml:
228     Hxml (filenames *.hxml)
229 * html+cheetah, html+spitfire:
230     HTML+Cheetah
231 * html+django, html+jinja:
232     HTML+Django/Jinja
233 * html+evoque:
234     HTML+Evoque (filenames *.html)
235 * html+genshi, html+kid:
```


236 HTML+Genshi

237 * html+lasso:

238 HTML+Lasso

239 * html+mako:

240 HTML+Mako

241 * html+mako:

242 HTML+Mako

243 * html+myghty:

244 HTML+Myghty

245 * html+php:

246 HTML+PHP (filenames *.phtml)

247 * html+smarty:

248 HTML+Smarty

249 * html+velocity:

250 HTML+Velocity

251 * html:

252 HTML (filenames *.html, *.htm, *.xhtml, *.xslt)

253 * http:

254 HTTP

255 * hx, haXe:

256 haXe (filenames *.hx)

257 * hybris, hy:

258 Hybris (filenames *.hy, *.hyb)

259 * idl:

260 IDL (filenames *.pro)

261 * iex:

262 Elixir iex session

```
263 * ini, cfg:
264     INI (filenames *.ini, *.cfg)
265 * io:
266     Io (filenames *.io)
267 * ioke, ik:
268     Ioke (filenames *.ik)
269 * irc:
270     IRC logs (filenames *.weechatlog)
271 * jade, JADE:
272     Jade (filenames *.jade)
273 * jags:
274     JAGS (filenames *.jag, *.bug)
275 * java:
276     Java (filenames *.java)
277 * jlcon:
278     Julia console
279 * js+cheetah, javascript+cheetah, js+spitfire, javascript+spitfire:
280     JavaScript+Cheetah
281 * js+django, javascript+django, js+jinja, javascript+jinja:
282     JavaScript+Django/Jinja
283 * js+erb, javascript+erb, js+ruby, javascript+ruby:
284     JavaScript+Ruby
285 * js+genshitext, js+genshi, javascript+genshitext, javascript+genshi:
286     JavaScript+Genshi Text
287 * js+lasso, javascript+lasso:
288     JavaScript+Lasso
289 * js+mako, javascript+mako:
```

```
290     JavaScript+Mako
291 * js+mako, javascript+mako:
292     JavaScript+Mako
293 * js+myghty, javascript+myghty:
294     JavaScript+Myghty
295 * js+php, javascript+php:
296     JavaScript+PHP
297 * js+smarty, javascript+smarty:
298     JavaScript+Smarty
299 * js, javascript:
300     JavaScript (filenames *.js)
301 * json:
302     JSON (filenames *.json)
303 * jsp:
304     Java Server Page (filenames *.jsp)
305 * julia, jl:
306     Julia (filenames *.jl)
307 * kconfig, menuconfig, linux-config, kernel-config:
308     Kconfig (filenames Kconfig, *Config.in*, external.in*, standard-modules.in)
309 * koka:
310     Koka (filenames *.kk, *.kki)
311 * kotlin:
312     Kotlin (filenames *.kt)
313 * lasso, lassoscript:
314     Lasso (filenames *.lasso, *.lasso[89])
315 * lhs, literate-haskell:
316     Literate Haskell (filenames *.lhs)
```

```
317 * lighty, lighttpd:
318     Lighttpd configuration file
319 * live-script, livescript:
320     LiveScript (filenames *.ls)
321 * llvm:
322     LLVM (filenames *.ll)
323 * logos:
324     Logos (filenames *.x, *.xi, *.xm, *.xmi)
325 * logtalk:
326     Logtalk (filenames *.lgt)
327 * lua:
328     Lua (filenames *.lua, *.wlua)
329 * make, makefile, mf, bsdmake:
330     Makefile (filenames *.mak, Makefile, makefile, Makefile.*, GNUmakefile)
331 * mako:
332     Mako (filenames *.mao)
333 * mako:
334     Mako (filenames *.mao)
335 * maql:
336     MAQL (filenames *.maql)
337 * mason:
338     Mason (filenames *.m, *.mhtml, *.mc, *.mi, autohandler, dhandler)
339 * matlab:
340     Matlab (filenames *.m)
341 * matlabsession:
342     Matlab session
343 * minid:
```

```
344     MiniD (filenames *.md)
345 * modelica:
346     Modelica (filenames *.mo)
347 * modula2, m2:
348     Modula-2 (filenames *.def, *.mod)
349 * monkey:
350     Monkey (filenames *.monkey)
351 * moocode:
352     MOOCode (filenames *.moo)
353 * moon, moonscript:
354     MoonScript (filenames *.moon)
355 * mscgen, msc:
356     Mscgen (filenames *.msc)
357 * mupad:
358     MuPAD (filenames *.mu)
359 * mxml:
360     MXML (filenames *.mxml)
361 * myghty:
362     Myghty (filenames *.myt, autodelegate)
363 * mysql:
364     MySQL
365 * nasm:
366     NASM (filenames *.asm, *.ASM)
367 * nemerle:
368     Nemerle (filenames *.n)
369 * newlisp:
370     NewLisp (filenames *.lsp, *.nl)
```

```
371 * newspeak:
372     Newspeak (filenames *.ns2)
373 * nginx:
374     Nginx configuration file
375 * nimrod, nim:
376     Nimrod (filenames *.nim, *.nimrod)
377 * nsis, nsi, nsh:
378     NSIS (filenames *.nsi, *.nsh)
379 * numpy:
380     NumPy
381 * objdump:
382     objdump (filenames *.objdump)
383 * objective-c++, objectivec++, obj-c++, objc++:
384     Objective-C++ (filenames *.mm, *.hh)
385 * objective-c, objectivec, obj-c, objc:
386     Objective-C (filenames *.m, *.h)
387 * objective-j, objectivej, obj-j, objj:
388     Objective-J (filenames *.j)
389 * ocaml:
390     OCaml (filenames *.ml, *.mli, *.mll, *.mly)
391 * octave:
392     Octave (filenames *.m)
393 * ooc:
394     Ooc (filenames *.ooc)
395 * opa:
396     Opa (filenames *.opa)
397 * openedge, abl, progress:
```

```
398     OpenEdge ABL (filenames *.p, *.cls)
399 * perl, pl:
400     Perl (filenames *.pl, *.pm)
401 * php, php3, php4, php5:
402     PHP (filenames *.php, *.php[345], *.inc)
403 * plpgsql:
404     PL/pgSQL
405 * postgresql, postgres:
406     PostgreSQL SQL dialect
407 * postscript:
408     PostScript (filenames *.ps, *.eps)
409 * pot, po:
410     Gettext Catalog (filenames *.pot, *.po)
411 * pov:
412     POV-Ray (filenames *.pov, *.inc)
413 * powershell, posh, ps1:
414     PowerShell (filenames *.ps1)
415 * prolog:
416     Prolog (filenames *.prolog, *.pro, *.pl)
417 * properties:
418     Properties (filenames *.properties)
419 * protobuf:
420     Protocol Buffer (filenames *.proto)
421 * psql, postgresql-console, postgres-console:
422     PostgreSQL console (psql)
423 * puppet:
424     Puppet (filenames *.pp)
```

```
425 * py3tb:
426     Python 3.0 Traceback (filenames *.py3tb)
427 * pycon:
428     Python console session
429 * pypylog, pypy:
430     PyPy Log (filenames *.pypylog)
431 * pytb:
432     Python Traceback (filenames *.pytb)
433 * python, py, sage:
434     Python (filenames *.py, *.pyw, *.sc, SConstruct, SConscript, *.tac, *.sage)
435 * python3, py3:
436     Python 3
437 * qml, Qt Meta Language, Qt modeling Language:
438     QML (filenames *.qml)
439 * racket, rkt:
440     Racket (filenames *.rkt, *.rktl)
441 * ragel-c:
442     Ragel in C Host (filenames *.rl)
443 * ragel-cpp:
444     Ragel in CPP Host (filenames *.rl)
445 * ragel-d:
446     Ragel in D Host (filenames *.rl)
447 * ragel-em:
448     Embedded Ragel (filenames *.rl)
449 * ragel-java:
450     Ragel in Java Host (filenames *.rl)
451 * ragel-objc:
```



```
452     Ragel in Objective C Host (filenames *.rl)
453 * ragel-ruby, ragel-rb:
454     Ragel in Ruby Host (filenames *.rl)
455 * ragel:
456     Ragel
457 * raw:
458     Raw token data
459 * rb, ruby, duby:
460     Ruby (filenames *.rb, *.rbw, Rakefile, *.rake, *.gemspec, *.rbx, *.duby)
461 * rbcon, irb:
462     Ruby irb session
463 * rconsole, rout:
464     RConsole (filenames *.Rout)
465 * rd:
466     Rd (filenames *.Rd)
467 * rebol:
468     REBOL (filenames *.r, *.r3)
469 * redcode:
470     Redcode (filenames *.cw)
471 * registry:
472     reg (filenames *.reg)
473 * rhtml, html+erb, html+ruby:
474     RHTML (filenames *.rhtml)
475 * rst, rest, restructuredtext:
476     reStructuredText (filenames *.rst, *.rest)
477 * rust:
478     Rust (filenames *.rs, *.rc)
```

```
479 * sass, SASS:
480     Sass (filenames *.sass)
481 * scala:
482     Scala (filenames *.scala)
483 * scaml, SCAML:
484     Scaml (filenames *.scaml)
485 * scheme, scm:
486     Scheme (filenames *.scm, *.ss)
487 * scilab:
488     Scilab (filenames *.sci, *.sce, *.tst)
489 * scss:
490     SCSS (filenames *.scss)
491 * shell-session:
492     Shell Session (filenames *.shell-session)
493 * smali:
494     Smali (filenames *.smali)
495 * smalltalk, squeak:
496     Smalltalk (filenames *.st)
497 * smarty:
498     Smarty (filenames *.tpl)
499 * sml:
500     Standard ML (filenames *.sml, *.sig, *.fun)
501 * snobol:
502     Snobol (filenames *.snobol)
503 * sourceslist, sources.list:
504     Debian Sourcelist (filenames sources.list)
505 * sp:
```

```
506     SourcePawn (filenames *.sp)
507 * spec:
508     RPMSpec (filenames *.spec)
509 * splus, s, r:
510     S (filenames *.S, *.R, .Rhistory, .Rprofile)
511 * sql:
512     SQL (filenames *.sql)
513 * sqlite3:
514     sqlite3con (filenames *.sqlite3-console)
515 * squidconf, squid.conf, squid:
516     SquidConf (filenames squid.conf)
517 * ssp:
518     Scalate Server Page (filenames *.ssp)
519 * stan:
520     Stan (filenames *.stan)
521 * systemverilog, sv:
522     systemverilog (filenames *.sv, *.svh)
523 * tcl:
524     Tcl (filenames *.tcl)
525 * tcsh, csh:
526     Tcsh (filenames *.tcsh, *.csh)
527 * tea:
528     Tea (filenames *.tea)
529 * tex, latex:
530     TeX (filenames *.tex, *.aux, *.toc)
531 * text:
532     Text only (filenames *.txt)
```

```

533 * trac-wiki, moin:
534     MoinMoin/Trac Wiki markup
535 * treetop:
536     Treetop (filenames *.treetop, *.tt)
537 * ts:
538     TypeScript (filenames *.ts)
539 * urbiscript:
540     UrbiScript (filenames *.u)
541 * vala, vapi:
542     Vala (filenames *.vala, *.vapi)
543 * vb.net, vbnet:
544     VB.net (filenames *.vb, *.bas)
545 * velocity:
546     Velocity (filenames *.vm, *.fhtml)
547 * verilog, v:
548     verilog (filenames *.v)
549 * vgl:
550     VGL (filenames *.rpf)
551 * vhd1:
552     vhd1 (filenames *.vhd1, *.vhd)
553 * vim:
554     VimL (filenames *.vim, .vimrc, .exrc, .gvimrc, _vimrc, _exrc, _gvimrc, vimrc, gvimrc)
555 * xml+cheetah, xml+spitfire:
556     XML+Cheetah
557 * xml+django, xml+jinja:
558     XML+Django/Jinja
559 * xml+erb, xml+ruby:

```

```
560     XML+Ruby
561 * xml+evoque:
562     XML+Evoque (filenames *.xml)
563 * xml+lasso:
564     XML+Lasso
565 * xml+mako:
566     XML+Mako
567 * xml+mako:
568     XML+Mako
569 * xml+myghty:
570     XML+Myghty
571 * xml+php:
572     XML+PHP
573 * xml+smarty:
574     XML+Smarty
575 * xml+velocity:
576     XML+Velocity
577 * xml:
578     XML (filenames *.xml, *.xsl, *.rss, *.xslt, *.xsd, *.wsdl)
579 * xquery, xqy, xq, xql, xqm:
580     XQuery (filenames *.xqy, *.xquery, *.xq, *.xql, *.xqm)
581 * xslt:
582     XSLT (filenames *.xsl, *.xslt, *.xpl)
583 * xtend:
584     Xtend (filenames *.xtend)
585 * yaml:
586     YAML (filenames *.yaml, *.yml)
```

```

587
588 Formatters:
589 ~~~~~
590 * bbcode, bb:
591     Format tokens with BBcodes.
592     These formatting codes are used by
593     many bulletin boards, so you can highlight
594     your sourcecode with pygments before posting it there.
595 * bmp, bitmap:
596     Create a bitmap image from source code.
597     This uses the Python Imaging Library to generate a
598     pixmap from the source code. (filenames *.bmp)
599 * gif:
600     Create a GIF image from source code.
601     This uses the Python Imaging Library to
602     generate a pixmap from the source code. (filenames *.gif)
603 * html:
604     Format tokens as HTML 4
605     '<span>' tags within a '<pre>' tag, wrapped in a '<div>' tag.
606     The '<div>'s CSS class can be set by the 'cssclass' option. (filenames *.html, *.ht
607 * img, IMG, png:
608     Create a PNG image from source code.
609     This uses the Python Imaging Library to generate a
610     pixmap from the source code. (filenames *.png)
611 * jpg, jpeg:
612     Create a JPEG image from source code. This uses
613     the Python Imaging Library to generate a pixmap from

```

```
614     the source code. (filenames *.jpg)
615 * latex, tex:
616     Format tokens as LaTeX code. This needs the 'fancyvrb'
617     and 'color' standard packages. (filenames *.tex)
618 * raw, tokens:
619     Format tokens as a raw representation for storing
620     token streams. (filenames *.raw)
621 * rtf:
622     Format tokens as RTF markup.
623     This formatter automatically outputs full RTF documents
624     with color information and other useful stuff. Perfect for
625     Copy and Paste into Microsoft® Word® documents. (filenames *.rtf)
626 * svg:
627     Format tokens as an SVG graphics file. This formatter
628     is still experimental. Each line of code is a '<text>'
629     element with explicit 'x' and 'y' coordinates containing
630     '<tspan>' elements with the individual token styles. (filenames *.svg)
631 * terminal, console:
632     Format tokens with ANSI color sequences, for output in
633     a text console. Color sequences are terminated at newlines,
634     so that paging the output works correctly.
635 * terminal256, console256, 256:
636     Format tokens with ANSI color
637     sequences, for output in a
638     256-color terminal or console.
639     Like in 'TerminalFormatter'
640     color sequences are terminated
```

641 at newlines, so that paging
642 the output works correctly.
643 * text, null:
644 Output the text unchanged without
645 any formatting. (filenames *.txt)
