

# 附录 快速索引

名称	作用	所在章节	页码
os 模块 —— 系统操作			
_exit() 方法	退出指定状态的进程	3.2	P58
abort() 方法	发送一个程序请求异常终止信号	3.3	P60
cpu_count() 方法	获取系统中的 CPU 数量	1.8	P44
environb 属性	获取当前环境变量的值（字节）	2.3	P55
environ 属性	获取当前环境变量的值（字符串）	2.2	P53
exec 族方法	执行新程序	1.3	P35
get_handle_inheritable() 方法	获取句柄的可继承标志	4.5	P70
get_inheritable() 方法	获取文件描述符的可继承标志	4.7	P72
get_terminal_size() 方法	获取终端窗口的大小	1.9	P45
getenv() 方法	获取环境变量的值	2.1	P49
getlogin() 方法	获取系统的登录用户名	1.6	P43
getpid() 方法	获取当前进程 ID	3.4	P61
getppid() 方法	获取父进程 ID	3.5	P63
isatty() 方法	判断是否连接到 TTY 设备	1.7	P43
kill() 方法	杀死进程	3.6	P63
linesep 属性	操作系统中分隔行的字符串	1.4	P41
name 属性	获取依赖操作系统的模块名称	1.5	P42
popen() 方法	执行系统命令	1.1	P2
set_handle_inheritable() 方法	设置句柄的可继承标志	4.4	P69
set_inheritable() 方法	设置文件描述符的可继承标志	4.6	P71
strerror() 方法	根据错误码返回错误信息	4.1	P65
supports_bytes_environ 属性	判断原生环境类型是否为字节型	4.2	P66
system() 方法	执行本地命令（不读取结果）	1.2	P13
times() 方法	返回当前的全局进程时间	3.1	P57
urandom() 方法	生成随机字节字符串	4.3	P67
sys 模块 —— 系统相关的属性和方法			
__breakpointhook__ 对象	breakpointhook() 函数的初始值	12.15	P205

名称	作用	所在章节	页码
<code>__excepthook__</code> 对象	<code>excepthook()</code> 函数的初始值	12.14	P203
<code>__interactivehook__</code> 属性	交互模式下启动解释器	6.14	P117
<code>__stderr__</code> 属性	<code>stderr</code> 值的文件对象	5.11	P99
<code>__stdin__</code> 属性	<code>stdin</code> 值的文件对象	5.9	P98
<code>__stdout__</code> 属性	<code>stdout</code> 值的文件对象	5.1	P99
<code>_clear_type_cache()</code> 方法	清除内部类型缓存	12.8	P195
<code>_current_frames()</code> 方法	将线程标识符映射到最顶层堆栈	10.4	P164
<code>_debugmallocstats()</code> 方法	打印关于 CPython 内存分配器状态的低级信息	7.9	P131
<code>_enablelegacywindowsfsencoding()</code> 方法	改变默认文件系统编码和错误模式	9.4	P150
<code>_getframe()</code> 方法	返回一个框架对象	12.13	P203
<code>_xoptions</code>	获取 <code>-X</code> 命令行选项的字典	12.12	P201
<code>api_version</code> 属性	Python 解释器的 C API 版本号	6.3	P104
<code>argv</code> 属性	程序命令行参数列表	6.8	P110
<code>base_exec_prefix</code> 属性	获取 Python 安装目录	8.3	P140
<code>base_prefix</code> 属性	获取 Python 虚拟环境目录	8.4	P141
<code>breakpointhook()</code> 方法	<code>breakpoint()</code> 的钩子函数	12.11	P199
<code>builtin_module_names</code> 属性	获取 Python 解释器中的所有模块名称	6.11	P114
<code>byteorder</code> 属性	本机字节顺序指示器	9.3	P149
<code>call_tracing()</code> 方法	递归调用 / 调试代码	10.7	P169
<code>copyright</code> 属性	Python 版权信息	6.4	P105
<code>displayhook()</code> 方法	打印表达式结果	12.7	P193
<code>dllhandle</code> 属性	针对 Python DLL 的整数句柄	7.11	P135
<code>dont_write_bytecode</code> 属性	是否写字节码文件	9.6	P153
<code>exc_info()</code> 方法	捕获正在处理的异常信息	11.1	P179
<code>excepthook()</code> 方法	发生未捕获异常时调用	11.3	P182
<code>exec_prefix</code> 属性	获取 Python 安装文件目录	8.5	P142
<code>executable</code> 属性	Python 解释器的绝对路径	8.1	P137
<code>exit()</code> 方法	退出 Python	6.7	P108
<code>flags</code> 属性	启动 Python 时的命令行选项	6.9	P112
<code>float_info</code> 属性	包含浮点类型的信息	12.4	P190

名称	作用	所在章节	页码
float_repr_style 属性	表示 repr() 函数输出的浮点数	12.5	P192
get_asyncgen_hooks() 方法	返回 asyncgen_hooks 对象	10.13	P177
get_coroutine_origin_tracking_depth() 方法	获取当前的协程源跟踪深度	10.11	P175
get_coroutine_wrapper() 方法	返回 None 或一个包装器	10.9	P173
getallocatedblocks() 方法	获取当前分配的内存块数量	7.8	P130
getcheckinterval() 方法	设置和获取解释器的检查间隔	6.15	P118
getdefaultencoding() 方法	获取默认字符串编码名称	9.1	P147
getfilesystemencodeerrors() 方法	获取文件名转换时的错误模式名称	11.2	P182
getfilesystemencoding() 方法	获取文件系统使用的编码名称	9.2	P147
getprofile() 方法	获取探查器函数	12.9	P196
getrecursionlimit() 方法	获取递归限制的值	10.5	P166
getrefcount() 方法	获取对象的引用计数	7.6	P127
getsizeof() 方法	返回对象占用内存大小	7.7	P129
getswitchinterval() 方法	获取线程切换间隔	10.1	P157
gettrace() 方法	获取设置的跟踪函数	7.5	P126
getwindowsversion() 方法	获取 Windows 版本信息	7.1	P121
hash_info 属性	获取哈希信息	9.5	P151
hexversion 属性	十六进制的版本号	6.5	P106
implementation 属性	当前运行的解释器的信息	6.6	P107
int_info 属性	Python 内部整数表示的信息	12.2	P189
intern() 方法	获取字符串或副本	12.6	P193
is_finalizing() 方法	获取是否关闭 Python 解释器	6.1	P113
maxsize 属性	最大整数值	12.3	P190
maxunicode 属性	能够表示 Unicode 码的最大整数	9.7	P155
meta_path	元路径查找器对象的列表	8.7	P145
modules 属性	将模块名称映射到已加载模块的字典	12.1	P187
path_hooks 属性	为路径创建 finder	8.8	P145
path_importer_cache 属性	查找程序对象缓存的字典	7.1	P133
path 属性	指定模块搜索路径	8.2	P137
platform 属性	获取系统标识符	7.2	P122

名称	作用	所在章节	页码
prefix 属性	表示特定于站点的目录前缀	8.6	P144
ps1 属性	解释器的主要提示	6.12	P115
ps2 属性	解释器的辅助提示	6.13	P116
set_asyncgen_hooks() 方法	设置事件循环中的异步生成终结器和迭代拦截器	10.12	P176
set_coroutine_origin_tracking_depth() 方法	允许启用或禁用协同源跟踪	10.1	P174
set_coroutine_wrapper() 方法	允许拦截协同程序对象的创建	10.8	P171
setcheckinterval() 方法	设置解释器的检查间隔	6.16	P119
setprofile() 方法	设置函数以允许对 Python 源码分析	12.1	P197
setrecursionlimit() 方法	设置递归限制的值	10.6	P167
setswitchinterval() 方法	设置线程切换间隔	10.2	P159
settrace() 方法	设置系统的跟踪功能	7.4	P125
stderr 属性	用于标准错误的文件对象	5.7	P94
stdin 属性	Python 程序的标准输入流	5.1	P74
stdout 属性	Python 程序的标准输出流	5.4	P88
sys.stderr.write() 方法	标准错误输出	5.8	P95
sys.stdin.read() 方法	读取标准输入流的前 n 个字符	5.2	P77
sys.stdin.readline() 方法	读取全部的标准输入流	5.3	P80
sys.stdout.flush() 方法	刷新标准输出流	5.6	P92
sys.stdout.write() 方法	输出标准输出流	5.5	P90
thread_info 属性	包含有关线程实现信息的结构序列	10.3	P162
tracebacklimit 属性	发生未处理异常时打印的最大回溯信息级别数	11.4	P184
version_info 属性	包含 Python 版本信息各组件的元组	6.1	P101
version 属性	Python 解释器的版本号等信息	6.2	P103
warnoptions 属性	警告框架的实现细节	11.5	P186
winver 属性	Windows 平台上形成注册表项的版本号	7.3	P124