# **Python的platform模块**

* [python](http://so.csdn.net/so/search/s.do?q=python&t=blog" \t "http://blog.csdn.net/xc_tsao/article/details/_blank) /
* [platform](http://so.csdn.net/so/search/s.do?q=platform&t=blog" \t "http://blog.csdn.net/xc_tsao/article/details/_blank)

python中，platform模块给我们提供了很多方法去获取操作系统的信息

如：

import platform

platform.platform() #获取操作系统名称及版本号，'Linux-3.13.0-46-generic-i686-with-Deepin-2014.2-trusty'

platform.version() #获取操作系统版本号，'#76-Ubuntu SMP Thu Feb 26 18:52:49 UTC 2015'

platform.architecture() #获取操作系统的位数，('32bit', 'ELF')

platform.machine() #计算机类型，'i686'

platform.node() #计算机的网络名称，'XF654'

platform.processor() #计算机处理器信息，''i686'

platform.uname() #包含上面所有的信息汇总，('Linux', 'XF654', '3.13.0-46-generic', '#76-Ubuntu SMP Thu Feb 26 18:52:49 UTC 2015', 'i686', 'i686')

还可以获得计算机中python的一些信息：

import platform

platform.python\_build()

platform.python\_compiler()

platform.python\_branch()

platform.python\_implementation()

platform.python\_revision()

platform.python\_version()

platform.python\_version\_tuple()

**[python]** [view plain](http://blog.csdn.net/xc_tsao/article/details/44007143" \o "view plain) [copy](http://blog.csdn.net/xc_tsao/article/details/44007143" \o "copy)

1. #!/usr/bin/env python
2. #coding=utf-8
4. #platform\_mode.py
6. **import** platform
8. '''''
9. python中，platform模块给我们提供了很多方法去获取操作系统的信息
10. 如：
11. import platform
12. platform.platform()        #获取操作系统名称及版本号，'Linux-3.13.0-46-generic-i686-with-Deepin-2014.2-trusty'
13. platform.version()         #获取操作系统版本号，'#76-Ubuntu SMP Thu Feb 26 18:52:49 UTC 2015'
14. platform.architecture()    #获取操作系统的位数，('32bit', 'ELF')
15. platform.machine()         #计算机类型，'i686'
16. platform.node()            #计算机的网络名称，'XF654'
17. platform.processor()       #计算机处理器信息，''i686'
18. platform.uname()           #包含上面所有的信息汇总，('Linux', 'XF654', '3.13.0-46-generic', '#76-Ubuntu SMP Thu Feb 26 18:52:49 UTC 2015', 'i686', 'i686')
20. 还可以获得计算机中python的一些信息：
21. import platform
22. platform.python\_build()
23. platform.python\_compiler()
24. platform.python\_branch()
25. platform.python\_implementation()
26. platform.python\_revision()
27. platform.python\_version()
28. platform.python\_version\_tuple()
29. '''
31. #global var
32. #是否显示日志信息
33. SHOW\_LOG = True
35. **def** get\_platform():
36. '''''获取操作系统名称及版本号'''
37. **return** platform.platform()
39. **def** get\_version():
40. '''''获取操作系统版本号'''
41. **return** platform.version()
43. **def** get\_architecture():
44. '''''获取操作系统的位数'''
45. **return** platform.architecture()
47. **def** get\_machine():
48. '''''计算机类型'''
49. **return** platform.machine()
51. **def** get\_node():
52. '''''计算机的网络名称'''
53. **return** platform.node()
55. **def** get\_processor():
56. '''''计算机处理器信息'''
57. **return** platform.processor()
59. **def** get\_system():
60. '''''获取操作系统类型'''
61. **return** platform.system()
63. **def** get\_uname():
64. '''''汇总信息'''
65. **return** platform.uname()
67. **def** get\_python\_build():
68. ''''' the Python build number and date as strings'''
69. **return** platform.python\_build()
71. **def** get\_python\_compiler():
72. '''''Returns a string identifying the compiler used for compiling Python'''
73. **return** platform.python\_compiler()
75. **def** get\_python\_branch():
76. '''''Returns a string identifying the Python implementation SCM branch'''
77. **return** platform.python\_branch()
79. **def** get\_python\_implementation():
80. '''''Returns a string identifying the Python implementation. Possible return values are: ‘CPython’, ‘IronPython’, ‘Jython’, ‘PyPy’.'''
81. **return** platform.python\_implementation()
83. **def** get\_python\_version():
84. '''''Returns the Python version as string 'major.minor.patchlevel'
85. '''
86. **return** platform.python\_version()
88. **def** get\_python\_revision():
89. '''''Returns a string identifying the Python implementation SCM revision.'''
90. **return** platform.python\_revision()
92. **def** get\_python\_version\_tuple():
93. '''''Returns the Python version as tuple (major, minor, patchlevel) of strings'''
94. **return** platform.python\_version\_tuple()
96. **def** show\_os\_all\_info():
97. '''''打印os的全部信息'''
98. **print**('获取操作系统名称及版本号 : [{}]'.format(get\_platform()))
99. **print**('获取操作系统版本号 : [{}]'.format(get\_version()))
100. **print**('获取操作系统的位数 : [{}]'.format(get\_architecture()))
101. **print**('计算机类型 : [{}]'.format(get\_machine()))
102. **print**('计算机的网络名称 : [{}]'.format(get\_node()))
103. **print**('计算机处理器信息 : [{}]'.format(get\_processor()))
104. **print**('获取操作系统类型 : [{}]'.format(get\_system()))
105. **print**('汇总信息 : [{}]'.format(get\_uname()))
107. **def** show\_os\_info():
108. '''''只打印os的信息，没有解释部分'''
109. **print**(get\_platform())
110. **print**(get\_version())
111. **print**(get\_architecture())
112. **print**(get\_machine())
113. **print**(get\_node())
114. **print**(get\_processor())
115. **print**(get\_system())
116. **print**(get\_uname())
118. **def** show\_python\_all\_info():
119. '''''打印python的全部信息'''
120. **print**('The Python build number and date as strings : [{}]'.format(get\_python\_build()))
121. **print**('Returns a string identifying the compiler used for compiling Python : [{}]'.format(get\_python\_compiler()))
122. **print**('Returns a string identifying the Python implementation SCM branch : [{}]'.format(get\_python\_branch()))
123. **print**('Returns a string identifying the Python implementation : [{}]'.format(get\_python\_implementation()))
124. **print**('The version of Python ： [{}]'.format(get\_python\_version()))
125. **print**('Python implementation SCM revision : [{}]'.format(get\_python\_revision()))
126. **print**('Python version as tuple : [{}]'.format(get\_python\_version\_tuple()))
128. **def** show\_python\_info():
129. '''''只打印python的信息，没有解释部分'''
130. **print**(get\_python\_build())
131. **print**(get\_python\_compiler())
132. **print**(get\_python\_branch())
133. **print**(get\_python\_implementation())
134. **print**(get\_python\_version())
135. **print**(get\_python\_revision())
136. **print**(get\_python\_version\_tuple())
138. **def** test():
139. **print**('操作系统信息:')
140. **if** SHOW\_LOG:
141. show\_os\_all\_info()
142. **else**:
143. show\_os\_info()
144. **print**('#' \* 50)
145. **print**('计算机中的python信息：')
146. **if** SHOW\_LOG:
147. show\_python\_all\_info()
148. **else**:
149. show\_python\_info()
151. **def** init():
152. **global** SHOW\_LOG
153. SHOW\_LOG = True
155. **def** main():
156. init()
157. test()
159. **if** \_\_name\_\_ == '\_\_main\_\_':
160. main()