

Python for Scientists

Part 8-1 – The standard library

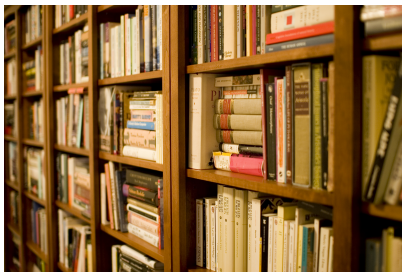
↪ *Cyril Desjoux* ↪

June, 2016

Updated : November 7, 2018



The standard library



Photography by Stewart Butterfield

Interacting with the `os`

Classical use

```

» import os
» os.listdir(os.curdir)           # List current directory content
   [myscript1.py, myscript2.py]

```

- `os` module provide an extensive library of tools to interact with the OS
- Some function are presented hereafter

Main functions of the `os` module

Method	Description
<code>os.uname()</code>	Return information identifying the current OS
<code>os.listdir(path)</code>	Return a list of the entries in the directory given by path
<code>os.mkdir(path)</code>	Create a directory
<code>os.makedirs(path)</code>	Recursively create a directory
<code>os.rmdir(path)</code>	Remove a directory
<code>os.removedirs(path)</code>	Recursively remove directories
<code>os.rename(src, dest)</code>	Rename a directory or a file
<code>os.remove(path)</code>	Remove a file
<code>os.system()</code>	Execute an external command
<code>os.getcurdir</code> (<code>/getcwd</code>)	Returns the current (<code>/working</code>) directory
<code>os.chdir()</code>	Change working directory

The regular expressions

Classical syntax

```
» import re
» re.match("^ello.+", "Hello World")
  <_sre.SRE_Match object; span=(0, 11), match=Hello World>
» re.match("'^ello.*w.*'", "Hello World")

» lst = ["ta01.txt", "tb02.txt", "tc03.txt", "td04.txt", "c0.cfg"]
» [re.sub("^t[a-z]0", "a00", i) for i in lst]
  [a001.txt, a002.txt, a003.txt, a004.txt, c0.cfg]
```

- Python re standard package provides support for regular expression
- Regular expressions are used in almost all languages
- Regular expression are powerful tools to manipulate strings

The regular expressions

Main metacharacters and classes

Metachar	Description
?	Matches the preceding element 0 or 1 time
+	Matches the preceding element 1 or more times
*	Matches the preceding element 0, 1, or more times
.	Matches any single character
^	Matches the starting position, or opposite of
\$	Matches the ending position
[]	Matches a single character that is contained within the brackets
()	Marked subexpression (also known as <i>block</i> or <i>capturing group</i>)
	Matches either the expression before or the expression after the operator
{m, n}	Matches the preceding element at least m and not more than n times
\d	Matches any decimal digit : equivalent to the class [0-9]
\D	Matches any non digit character : equivalent to the class [^0-9]
\s	Matches any whitespace character
\S	Matches any non-whitespace character
\w	Matches any alphanumeric character : equivalent to [a-zA-Z0-9_]
\W	Matches any non-alphanumeric character ; equivalent to [^a-zA-Z0-9_]
\	Escape character