

<b>List Methods</b>	<b>os lib Variables (cont)</b>	<b>Class Special Methods</b>	<b>String Methods (cont)</b>
append(item)	sep Path separator	__new__(cls)	endswith(sub)
count(item)	Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"	__init__(self, args)	expandtabs()
extend(list)		__del__(self)	find(sub, start, end)
index(item)		__repr__(self)	isalnum()*
insert(position, item)	<b>Operations on Dicts</b>	__str__(self)	isalpha()*
pop(position)	d.update(d2)	__cmp__(self, other)	isdigit()*
remove(item)	d.keys()	__index__(self)	islower()*
reverse()	d.values()	__hash__(self)	isspace()*
sort()	d.items()	__getattr__(self, name)	istitle()*
	d.pop(key[,default])	__setattr__(self, name, attr)	isupper()*
<b>List Slices and Indexes</b>	d.popitem()	__lt__(self, other)	join()
len(a) 6	d.get(key[,default])	__le__(self, other)	ljust(width)
a[0] 0	d.setdefault(key[,default])	__gt__(self, other)	lower()*
a[5] 5	d.clear()	__ge__(self, other)	lstrip()
a[-1] 5	del d[key]	__eq__(self, other)	partition(sep)
a[-2] 4	d[key] = value	__ne__(self, other)	replace(old, new)
a[1:] [1,2,3,4,5]	<b>Datetime Methods</b>	__nonzero__(self)	rfind(sub, start, end)
a[:5] [0,1,2,3,4]	today()	__delattr__(self, name)	rindex(sub, start, end)
a[:-2] [0,1,2,3]	now(timezoneinfo)	__call__(self, args, kwargs)	rjust(width)
a[1:3] [1,2]	utcnow()	<b>Operations on Sets</b>	rpartition(sep)
a[1:-1] [1,2,3,4]	fromtimestamp(timestamp)	union	rsplit(sep)
b=a[:] Shallow copy of a	utcfromtimestamp(timestamp)	& intersection	rstrip()
<b>Indexes and Slices of a.</b>	fromordinal(ordinal)	- ^ difference/symmetric diff	split(sep)
a=[0,1,2,3,4,5]	combine(date, time)	< <= > >= inclusion relations	splitlines()
<b>os lib Variables</b>	strptime(date, format)	s.upda- s.add(key)	startswith(sub)
altsep Alternative sep	<b>Time Methods</b>	te(s2)	strip()
curdir Current dir string	replace()	s.copy() s.discard(key)	swapcase()*
defpath Default search path	isoformat()	s.pop() s.clear()	title()*
devnull Path of null device	__str__()	<b>String Methods</b>	translate(table)
extsep Extension separator	strtime(format)	capitalize()*	upper()*
linesep Line separator	utcoffset()	center(width)	zfill(width)
name Name of OS	dst()	countr(sub, start, end)	
pardir Parent dir string	tzname()	decode()	Methods marked * are locale dependant for 8-bit strings.
pathsep Patch separator		encode()	



### File Methods

close()  
flush()  
fileno()  
isatty()  
next()  
read(size)  
readline(size)  
readlines(size)  
seek(offset)  
tell()  
truncate(size)  
write(string)  
writelines(list)

### Date Formatting

%a Abbreviated weekday (Sun)  
%A Weekday (Sunday)  
%b Abbreviated month name (Jan)  
%B Month name (January)  
%c Date and time  
%d Day (leading zeros) (01 to 31)  
%H 24 hour (leading zeros) (00 to 23)  
%I 12 hour (leading zeros) (01 to 12)  
%j Day of year (001 to 366)  
%m Month (01 to 12)  
%M Minute (00 to 59)  
%p AM or PM  
%S Second (00 to 61<sup>4</sup>)  
%U Week number<sup>1</sup> (00 to 53)  
%w Weekday<sup>2</sup> (0 to 6)  
%W Week number<sup>3</sup> (00 to 53)  
%x Date  
%X Time

### Date Formatting (cont)

%y Year without century (00 to 99)  
%Y Year (2008)  
%Z Time zone (GMT)  
%% A literal "%" character (%)

<sup>1</sup> Sunday as start of week. All days in a new year preceding the first Sunday are considered to be in week 0.

<sup>2</sup> 0 is Sunday, 6 is Saturday.

<sup>3</sup> Monday as start of week. All days in a new year preceding the first Monday are considered to be in week 0.

<sup>4</sup> This is not a mistake. Range takes account of leap and double-leap seconds.

### sys lib Variables and sys.args

argv Command line args  
builtin\_module\_names Linked C modules  
byteorder Native byte order  
check\_interval Signal check frequency  
exec\_prefix Root directory  
executable Name of executable  
exitfunc Exit function name  
modules Loaded modules  
path Search path  
platform Current platform  
stdin, stdout, stderr File objects for I/O

### sys lib Variables and sys.args (cont)

version\_info Python version info  
winver Version number  
sys.argv[0] foo.py  
sys.argv[1] bar  
sys.argv[2] -c  
sys.argv[3] qux  
sys.argv[4] --h

sys.argv for the command:  
\$ python foo.py bar -c qux --h

