

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



### 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

