Python Cheat Sheet

by Runnko via cheatography.com/24696/cs/6190/

List Methods

- append(item)
- count(item)
- extend(list)
- index(item)
- insert(position, item)
- pop(position)
- remove(item)
- reverse()
- sort()

List Slices and Indexes

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]

os lib Variables

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

os lib Variables (cont)

sep	Path separator

Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"

Operations on Dicts

- 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

Datetime Methods

- today()
- now(timezoneinfo)
- utcnow()
- fromtimestamp(timestamp)
- utcfromtimestamp(timestamp)
- fromordinal(ordinal)
- combine(date, time)
- strptime(date, format)

Time Methods

- replace()
- isoformat()
- __str__()
- strftime(format)
- utcoffset()
- dst()
- tzname()

Class Special Methods

- __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)

Opetaions on Sets

|	union
&	intersection
- ^	différence/symetric diff
< <= > >=	inclusion relations
s.update(s2)	s.add(key)
s.copy()	s.discard(key)
s.pop()	s.clear()

String Methods

- capitalize()*
- center(width)
- countr(sub, start, end)
- decode()
- encode()

String Methods (cont)

- 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)

Methods marked * are locale dependant for 8-bit strings.

By **Runnko**
cheatography.com/runnko/

Published 26th November, 2015.
Last updated 26th November, 2015.

File Methods

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

Date Formating

%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[4])
%U	Week number[1] (00 to 53)
%w	Weekday[2] (0 to 6)
%W	Week number[3] (00 to 53)
%x	Date
%X	Time
%y	Year without century (00 to 99)
%Y	Year (2008)
%Z	Time zone (GMT)
%%	A literal "%" character (%)

[1] Sunday as start of week. All days in a new year preceding the first Sunday are considered to be in week 0.

[2] 0 is Sunday, 6 is Saturday.

[3] Monday as start of week. All days in a new year preceding the first Monday are considered to be in week 0.

[4] 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
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