Cheatography

Python Cheat Sheet

by filoucool via cheatography.com/157372/cs/33337/

Comparison Operators

x< y	Less	x <= y	Less or eq
x > y	Greater	x >= y	Greater or eq
x == y	Equal	x != y	Not equal

Boolean Operators

not x	x and y	x or y

Arithmetic Operators

x + y	add	x - y	subtract
x * y	multiply	x / y	divide
x % y	modulus	x ** y	x^y

Assignment shortcuts: x op= y
Example: x += 1 increments x

Exception Handling

```
try:
      sta tements
except [ exception type [ asvar
] ]:
      sta tements
finally:
      sta tements
```

String / List / Tuple Operations

len(*s*)	length of s
s[i]	ith item in s (0-based)
s[start : end]	slice of *s* from *start* (included) to *end* (excluded)
x in *s*	**True** if *x* is contained in *s*
x not in s	**True** if *x* is not contained in *s*
s + *t*	the concatenation of *s* with *t*
s * *n*	*n* copies of *s* concatenated
sorted(*s*)	a sorted copy of *s*
s.index(*item*)	position in *s* of *item*
s.lower()	lowercase copy of *s*

String / List / Tuple Operations (cont)

s.replace(*old, new*)	copy of *s* with old replaced with *new*
s.split(*delim*)	list of substrings delimited by *delim*
s.strip()	copy of *s* with whitespace trimmed
s.upper()	uppercase copy of *s*

See also
http://docs.python.org/library/stdtypes.html#stringmethods

Environment

sys.argv	List of command line arguments (argv[0] is executable)
os.environ	Dictionary of environment variables
os.curdir	String with path of current directory

import sys; print(sys.argv) or
from sys import argv; print(argv)

Python List Methods

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

Python sys.argv

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

Function Definitions

```
def name(arg1, arg2, ...):
   statements
      return expr
```

Conversion Functions

int(*expr*)	Converts *expr* to integer
float(*expr*)	Converts *expr* to float
str(*expr*)	Converts *expr* to string
chr(*num*)	ASCII char *num*

Data Types

Integer	-256, 15
Float	-253.23, 1.253e-10
String	"Hello", 'Goodbye', """Multiline"""
Boolean	True, False
List	[value, ...]
Tuple	(value, ...) [1]
Dictionary	{ key: value, ... }
Set	{ value, value, ... } [2]

1. Parentheses usually optional
2. Create an empty set with set()

Mutating List Operations

del *lst[i]*	Deletes *i*th item from lst
lst.append(*e*)	Appends *e* to *lst*
lst.insert(*i, e*)	Inserts *e* before *i*th item in *lst*
lst.sort()	Sorts *lst*

See also
https://docs.python.org/3/library/stdtypes.html#typesseq-mutable

Python File Methods

close()	readlines(*size*)
flush()	seek(*offset*)
fileno()	tell()
isatty()	truncate(*size*)

By **filoucool**
cheatography.com/filoucool/

Not published yet.
Last updated 25th July, 2022.

Cheatography

Python Cheat Sheet

by filoucool via cheatography.com/157372/cs/33337/

Python File Methods (cont)

next()	write(*string*)
read(*size*)	writelines(*list*)
readline(*size*)	

Python Indexes and Slices

len(a)	6
a[0]	h
a[5]	n
a[-1]	n
a[-2]	m
a[1:]	[i,j,k,m,n]
a[:5]	[h,i,j,k,m]
a[:-2]	[h,i,j,k]
a[1:3]	[i,j]
a[1:-1]	[i,j,k,m]
b=a[:]	Shallow copy of a

Indexes and Slices of a=[h,i,j,k,m,n]

Statements

If Statement

```
if expression:
   statements
elif expression:
   statements
else:
   statements
```

While Loop

```
while expression:
   statements
```

For Loop

```
for var in collection:
   statements
```

Counting For Loop

```
for i in range(start, end [,
step]):
   statements
(start is included; end is not)
```

Python 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[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.

Python Datetime Methods

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

Python Class Special Methods

__new__(cls)	__lt__(self, other)
__init__(self, args)	__le__(self, other)
__del__(self)	__gt__(self, other)
__repr__(self)	__ge__(self, other)
__str__(self)	__eq__(self, other)
__cmp__(self, other)	__ne__(self, other)
__index__(self)	__nonzero__(self)
__hash__(self)	
__getattribute__(self, name)	
__setattr__(self, name, attr)	
__delattr__(self, name)	
__getattr__(self, name)	
__call__(self, args, kwargs)	

Python Time Methods

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