Python 2.7 Quick Reference Sheet

ver 2.01 – 110105 (sjd)

Interactive Help in Python Shell

help()	Invoke interactive help
help(<i>m</i>)	Display help for module <i>m</i>
help(f)	Display help for function f
dir(m)	Display names in module m

Small Operator Precedence Table

func_name(args,)	Function call
x[index : index]	Slicing
x[index]	Indexing
x.attribute	Attribute reference
**	Exponentiation
*, /,%	Multiply, divide, mod
+, -	Add, subtract
>, <, <=, >=, !=, ==	Comparison
in, not in	Membership tests
not, and, or	Boolean operators
	NOT, AND, OR

Module Import

import module_name
from module_name import name, ...
from module_name import *

Common Data Types

Common D	Common Data Types	
Туре	Description Literal Ex	
int	32-bit Integer	3, -4
long	Integer > 32 bits	101L
float	Floating point number	3.0, -6.55
complex	Complex number	1.2J
bool	Boolean	True, False
str	Character sequence	"Python"
tuple	Immutable sequence	(2, 4, 7)
list	Mutable sequence	[2, x, 3.1]
dict	Mapping	{ x:2, y:5 }

Common Synta	x Structures
Assignment Sta	tement
var = exp	
Console Input/	•
<pre>var = input(</pre>	[prompt])
<i>var</i> = raw_in	put([<i>prompt</i>])
print <i>exp</i> [,]	•
Selection	
if (boolean_e	exp):
stmt	
[elif (boolear	n_exp):
stmt]	
[else:	
stmt]	
Repetition	
while (<i>boole</i>)	an_exp):
stmt	
Traversal	
	versable_object:
stmt	
Function Defini	
-	_name(parmameters):
stmt	
Function Call	,
function_nai	me(arguments)
Class Definition	1
class <i>Class_n</i>	ame [(super_class)]:
[class vari	iables]
def <i>metho</i>	<pre>d_name(self, parameters):</pre>
stmt	
Object Instantia	ation
obj_ref = Cla	ss_name(arguments)
Method Invoca	tion
obj_ref.meth	nod_name(arguments)
Exception Hand	lling
try:	
stmt	
	ption_type] [, var]:
stmt	

Common Built-in Functions

Function	Returns
abs(x)	Absolute value of <i>x</i>
dict()	Empty dictionary, eg: d = dict()
float(x)	int or string x as float
id(<i>obj</i>)	memory addr of obj
int (x)	float or string x as int
len(s)	Number of items in sequence s
list()	Empty list, eg: m = list()
max(s)	Maximum value of items in s
min(s)	Minimum value of items in s
open(f)	Open filename f for input
ord(<i>c</i>)	ASCII code of <i>c</i>
pow(<i>x,y</i>)	x ** y
range(x)	A list of x ints 0 to x - 1
round(x,n)	float x rounded to n places
str(<i>obj</i>)	str representation of obj
sum(s)	Sum of numeric sequence s
tuple(items)	tuple of items
type(<i>obj</i>)	Data type of <i>obj</i>

Common Math Module Functions

Function	Returns (all float)	
ceil(x)	Smallest whole $nbr >= x$	
cos(x)	Cosine of x radians	
degrees(x)	x radians in degrees	
radians(x)	x degrees in radians	
exp(<i>x</i>)	e ** x	
floor(x)	Largest whole nbr <= x	
hypot(x, y)	sqrt(x * x + y * y)	
log(x [, base])	Log of x to base or natural log if	
	base not given	
pow(<i>x, y</i>)	x ** y	
sin(x)	Sine of x radians	
sqrt(x)	Positive square root of x	
tan(<i>x</i>)	Tangent of x radians	
pi	Math constant pi to 15 sig figs	
е	Math constant e to 15 sig figs	

Common String Methods

S.method()	Returns (str unless noted)	
capitalize	S with first char uppercase	
center(w)	S centered in str w chars wide	
count(sub)	int nbr of non-overlapping	
	occurrences of sub in S	
find(sub)	int index of first occurrence of	
	sub in S or -1 if not found	
isdigit()	bool True if S is all digit chars,	
	False otherwise	
islower()	bool True if S is all lower/upper	
isupper()	case chars, False otherwise	
join(seq)	All items in seq concatenated	
	into a str, delimited by S	
lower()	Lower/upper case copy of S	
upper()		
Istrip()	Copy of S with leading/ trailing	
rstrip()	whitespace removed, or both	
split([sep])	List of tokens in S, delimited by	
	sep; if sep not given, delimiter	
	is any whitespace	

Formatting Numbers as Strings

Syntax: "format_spec" % numeric_exp format_spec syntax: % width.precision type

- width (optional): align in number of colums specified; negative to left-align, precede with 0 to zero-fill
- precision (optional): show specified digits of precision for floats; 6 is default
- type (required): d (decimal int), f (float), s (string), e (float exponential notation)
- Examples for x = 123, y = 456.789

"%6d" % x -> . . . 123

"%06d" % x -> 000123

"%8.2f % y -> . . 456.79

"8.2e" % y -> 4.57e+02

"-8s" % "Hello" -> Hello . . .

Common List Methods

L.method()	Result/Returns
append(<i>obj</i>)	Append <i>obj</i> to end of <i>L</i>
count(<i>obj</i>)	Returns int nbr of occurrences of
	<i>obj</i> in <i>L</i>
index(<i>obj</i>)	Returns index of first occurrence
	of <i>obj</i> in <i>L</i> ; raises ValueError if
	<i>obj</i> not in <i>L</i>
pop([index])	Returns item at specified <i>index</i>
	or item at end of L if <i>index</i> not
	given; raises IndexError if L is
	empty or <i>index</i> is out of range
remove(<i>obj</i>)	Removes first occurrence of <i>obj</i>
	from <i>L</i> ; raises ValueError if <i>obj</i> is
	not in L
reverse()	Reverses <i>L</i> in place
sort()	Sorts <i>L</i> in place

Common Tuple Methods

T.method()	Returns
count(<i>obj</i>)	Returns nbr of occurrences of
	<i>obj</i> in <i>T</i>
index(<i>obj</i>)	Returns index of first occurrence
	of <i>obj</i> in <i>T</i> ; raises ValueError if
	obj is not in T

Common Dictionary Methods

D.method()	Result/Returns	
clear()	Remove all items from D	
get(<i>k</i> [, <i>val</i>])	Return $D[k]$ if k in D , else val	
has_key(k)	Return True if <i>k</i> in <i>D</i> , else False	
items()	Return list of key-value pairs in	
	D; each list item is 2-item tuple	
keys()	Return list of D's keys	
pop(<i>k</i> , [<i>val</i>])	Remove key k, return mapped	
	value or <i>val</i> if <i>k</i> not in <i>D</i>	
values()	Return list of D's values	

Common File Methods

F.method()	Result/Returns
read([<i>n</i>])	Return str of next <i>n</i> chars from <i>F</i> ,
	or up to EOF if <i>n</i> not given
readline([n])	Return str up to next newline, or
	at most <i>n</i> chars if specified
readlines()	Return list of all lines in F, where
	each item is a line
write(s)	Write str s to F
writelines(L)	Write all str in seq L to F
close()	Closes the file

Other Syntax

Hold window for user keystroke to close:	
raw_input("Press <enter> to quit.")</enter>	
Prevent execution on import:	
ifname == "main":	
main()	

Displayable ASCII Characters

32	SP	48	0	64	@	80	Р	96	`	112	р
33	!	49	1	65	Α	81	σ	97	а	113	q
34	u	50	2	66	В	82	R	98	b	114	r
35	#	51	3	67	U	83	S	99	C	115	S
36	\$	52	4	68	О	84	Т	100	d	116	t
37	%	53	5	69	Ε	85	כ	101	e	117	u
38	&	54	6	70	F	86	>	102	f	118	٧
39	1	55	7	71	O	87	W	103	g	119	W
40	(56	8	72	Ι	88	Χ	104	h	120	Х
41)	57	9	73	_	89	Υ	105	i	121	У
42	*	58	:	74	7	90	Z	105	j	122	Z
43	+	59	;	75	Κ	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	ı	124	
45	1	61	Ш	77	М	93]	109	m	125	}
46		62	>	78	N	94	۸	110	n	126	~
47	/	63	?	79	0	95		111	0	127	DEL

 $' \ 0' = 0, \ ' \ t' = 9, \ ' \ n' = 10$