

Python Quick Reference Card

Loosely adopted from John Zelle's "Python Programming: An Introduction to Computer Science", Franklin, Beedle, 1994.

Python Reserved Words

and	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	yield
def	finally	in	print	

print Statement

```
print <expr>
print <expr>, <expr>, ..., <expr>
print <expr>, <expr>, ..., <expr>
```

Assignment Statement

```
<variable> = <expr>
<variable1>, <variable2>, ..., <variableN>
    = <expr1>, <expr2>, ..., <exprN>
```

Input (numeric)

```
<variable> = input(<prompt>)
<variable1>, <variable2>, ..., <variableN>
    = input(<prompt>)
```

Input (string)

```
<variable> = raw_input(<prompt>)
```

Definite Loop (iteration)

```
for <variable> in <sequence>:
    <body>
```

while Loop

```
while <condition>:
    <body>
```

break Statement

```
while True:
    ...
    if <condition>: break
    ...
```

Module Import

```
import <module_name>
from <module_name> import
    <name1>, <name2>, ...
from <module_name> import *
```

Numeric Operators

Operator	Operation
+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation
%	integer remainder (modulus)
abs()	absolute value

Math Library Functions

Name	Returns
pi	an approximation of pi
e	an approximation of e
sin(x)	sine of x radians
cos(x)	cosine of x radians
tan(x)	tangent of x radians
asin(x)	inverse of sine x radians
acos(x)	inverse of cosine x radians
atan(x)	inverse of tangent x radians
log(x)	natural log of x
degrees(x)	converts x radians to degrees
radians(x)	converts x degrees to radians
log10(x)	base 10 log of x
exp(x)	exponential of x
ceil(x)	smallest integer \geq x
floor(x)	largest integer \leq x
sqrt(x)	square root of x

Common Built-in Functions

Name	Returns
range(n)	list of ints from 0 to n-1
range(m,n)	list of ints from m to n-1
range(m,n,p)	list of ints from m to n-1 counting by p
type(x)	Python data type of x
int(x)	the value of x converted to an int; x numeric or string
float(x)	the value of x converted to a float; x numeric or string
round(x)	nearest whole value of x

Sequence Ops (strings and lists)

Operator	Returns
<seq>+<seq>	concatenation of sequences
<seq>*n	sequence concatenated with itself n times
<seq>[n]	item at n (indexing): n \geq 0, 0-based from left n < 0, 1-based from right
len(<seq>)	number of items in sequence
<seq>[m:n]	slice from m to n-1; m, n default to 0, len(<seq>)

String Library Functions

Name	Meaning
capitalize(s)	Copy of s with only the 1 st character uppercase
capwords(s)	Copy of s with first character of each word uppercase
center(s,w)	Center s in a field of width w
count(s,sub)	Count the number of times sub occurs in s
find(s,sub)	Find first position where sub occurs in s
join(list)	Concatenate list of strings in one string
ljust(s,w)	Left-justify s in a field of width w
lower(s)	Copy of s in lowercase
lstrip(s)	Copy of s with leading whitespace removed
replace(s,old,new)	Replace all occurrences of substring old in s with substring new
rfind(s,sub)	Like find, but returns the rightmost position
rjust(s,sub)	Right-justify s in a field of width w
rstrip(s)	Copy of s with trailing whitespace removed
split(s,c)	Split s into a list of substrings delimited by optional character c (defaults is space)
upper(s)	Copy of s in uppercase

Type Conversion Functions

Name	Returns
float(<expr>)	expr as a floating point value
int(<expr>)	expr as an integer
long(<expr>)	expr as a long integer value
str(<expr>)	a string representation of expr
eval(<string>)	the numeric value of expr, evaluated as an expression
ord(c)	ASCII code for character c
chr(i)	character corresponding to integer ASCII code i

Boolean Expressions

Literals: True, False
Operators: and, or, not

Preventing Execution on Import

```
if __name__ == "__main__":
    main()
```

String Formatting

Expression syntax

<template-string> % (<value>, ...)

Specifier syntax

%<width>.<precision><type-char>

- width and precision are optional
- 0 width => use whatever space needed
- width with leading 0 => pad as needed with 0 (space is default)
- negative width means left-justify (right-justify is default)
- type-char: i (int), f (float), s (string)

File Processing

Opening and Closing Files

```
<filevar> = open(<name>,<mode>) -or-
<filevar> = file(<name>,<mode>)
```

- mode is 'r'ead, 'w'rite, or 'a'ppend
- add + for updating

```
<filevar>.close()
```

Reading a File

```
<filevar>.read()
<filevar>.readline()
<filevar>.readlines()
for <variable> in <filevar>:
    <body>
```

Writing a File

```
<filevar>.write(<string>)
```

Object Constructor

```
<class-name>(<param>, ...)
```

Object Method Call

```
<object>.<method_name>(<param>, ...)
```

Function Definition

```
def <name>(<formal-param>, ...)
```

Function Call

```
<name>(<formal-param>, ...)
```

Return Statement

```
return <expr>, ...
```

Simple Conditions

```
<expr> <relational-operator> <expr>
```

Relational Operators

Symbol	Meaning
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Note: These operators return a bool value (True or False)

if Statement

```
if <condition>:
    <body>
```

```
if <condition>:
    <body1>
else:
    <body2>
```

```
if <condition-1>:
    <body-1>
elif <condition-2>:
    <body-2>
...
else: #optional in elif
    <default-body>
```

Exception Handling

```
try:
    <statements>
except <ExceptionType>, <variable>:
    <handler1>
except <ExceptionType>, <variable>:
    <handler2>
...
except:
    <default handler>
```

Note: <variable> is optional; if present, is assigned the actual exception object that was raised.

Preventing Execution on Import

```
if __name__ == "__main__":
    main()
```

Random Library

random() returns a uniformly distributed pseudorandom value in the range [0,1)
 randrange(<params>) returns a uniformly distributed pseudorandom value from range(<params>)

Class Definition

```
class <class-name>:
    <method-definition>
    ...
```

- method-definition is a function with a special 1st parameter, self, that refers to the object to which the method is being applied
- The constructor method is __init__

docStrings

A string at the beginning of a module, class, function, or method can be used for documentation. docStrings are carried along at runtime and are used for interactive help and the PyDoc utility.

List Methods

Method	Meaning
<list>.append(x)	add x to end of list
<list>.sort()	sort the list
<list>.reverse()	reverse the list
<list>.index(x)	return index of first occurrence of x
<list>.insert(i,x)	insert x into list at i
<list>.count(x)	return count of number of x's in the list
<list>.remove(x)	delete 1st occurrence of x in the list
<list>.pop(i)	delete the ith element and return its value

ASCII Table (decimal)

32	SP	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	DEL

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