

Keywords			Keywords (cont)			Keywords (cont)		
Keyword	Description	Example						
and	Logical and	True and False == False	for	Loop over a collection of things.	for X in Y: pass	return	Exit the function with a return value.	def X(): return
as	Part of the with-as statement	with X as Y: pass	from	Importing specific parts of a module.	from x import y`	try	Try this block, and if exception, go to except.	try: pass
assert	Assert that something is true.	`assert False, "Error!"	global	Declares a global variable	global X	while	While loop	while X: pass
break	Stop this loop right now.	while True: break	if	If condition	if: X; elif: Y; else: J	with	With an expression as a variable do.	with X as Y: pas
class	Define a class.	class Person (object)	import	Import a module into this one to use.	import os	yield	Pause here and return to caller.	def x(): yield Y
continue	Don't process more of the loop, do it again.	while True: continue	is	Like == to test equality	1 is 1 == True	Data Types		
def	Define a function.	def X(): pass	lambda	Create a short anonymous function	s = lambda y: y	Type	Description	Example
del	Delete from dictionary.	del X[Y]	not	Logical not	not True == False	True	True boolean value	True or False :
elif	Else if	if: X; elif: Y; else: J	or	Logical or	True or False == True	False	False boolean value.	False and True
else	Else condition	if: X; elif: Y; else: J	pass	This block is empty.	def empty(): pass	None	Represents "nothing" or "no value".	x = None
except	If an exception happens, do this.	except ValueError, e: print e	print	Print this string.	print 'this string'	strings	Stores textual info	x = " hel lo"
exec	Run a string as Python	exec 'print " hel lo"'	raise	Raise an exception when things go wrong.	raise ValueError r("N o")	numbers	Stores integers	i = 100
finally	Exceptions or not, finally do this no matter what.	finally: pass				floats	Stores decimals	i = 10.389
						lists	Stores a list of things	j = [1,2,3,4]

Data Types (cont)

dicts Stores a key=value mapping of things `e = {'x': 1, 'y': 2}`

String Formats

Escape	Description	Example
--------	-------------	---------

%d	Decimal int	" %d" % 45 == '45'
----	-------------	--------------------

%o	Octal number	`%o % 1000 == '1750'
----	--------------	----------------------

%u	Unsigned decimal	" %u" % -1000 == '1000'
----	------------------	-------------------------

%x	Hexadecimal lowercase	" %x" % 1000 == '3e8'
----	-----------------------	-----------------------

%X	Hexadecimal uppercase.	" %X" % 1000 == '3E8'
----	------------------------	-----------------------

%e	Exponential notation, lowercase 'e'.	" %e" % 1000 == '1.000 000 e+03'
----	--------------------------------------	----------------------------------

%E	Exponential notation, uppercase 'E'.	" %E" % 1000 == '1.000 000 E+03'
----	--------------------------------------	----------------------------------

%f	Floating point real number.	" %f" % 10.34 == '10.34 0000'
----	-----------------------------	-------------------------------

%F	Same as %f.	" %F" % 10.34 == '10.34 0000'
----	-------------	-------------------------------

%g	Either %f or %e, whichever is shorter.	" %g" % 10.34 == '10.34'
----	--	--------------------------

%G	Same as %g but uppercase.	" %G" % 10.34 == '10.34'
----	---------------------------	--------------------------

%c	Character format.	" %c" % 34 == '4'
----	-------------------	-------------------

String Formats (cont)

%r Repr format (debugging format). `" %r" % int == "<type 'int'> "`

%s String format. `"%s there" % 'hi' == 'hi there'`

%g A percent sign. `" %g" % 10.34 == '10.34%'`

String Escape Sequences

**** Backslash

' Single-quote

" Double-quote

\a Bell

\b Backspace

\f Formfeed

\n Newline

\r Carriage

\t Tab

\v Vertical tab