Learn Python the Hard Way - Symbol Review Cheat Sheet

by dicewitch (dicewitch) via cheatography.com/38332/cs/11977/

Keywords

Keyword	Description	Example
`and`	Logical and	`True and False == False`
`as`	Part of the `with-as` statement	`with X as Y: pass`
`assert`	Assert that something is true.	`assert False, "Error!"`
`break`	Stop this loop right now.	`while True: break`
`class`	Define a class.	`class Person (ob ject)`
`continue`	Don't process more of the loop, do it again.	`while True: continue`
`def`	Define a function.	`def X(): pass`
`del`	Delete from dictionary.	`del X[Y]`
`elif`	Else if	`if: X; elif: Y; else: J`
`else`	Else condition	`if: X; elif: Y; else: J`
`except`	If an exception happens, do this.`	`except ValueE rror, e: print e`
`exec`	Run a string as Python	`exec 'print " hel lo"'`
`finally`	Exceptions or not, finally do this no matter what.	`finally: pass`

Keywords (cont)

`for`	Loop over a collection of things.	`for X in Y: pass`
`from`	Importing specific parts of a module.	`from x import y`
`global`	Declares a global variable	`global X`
`if`	If condition	`if: X; elif: Y; else: J`
`import`	Import a module into this one to use.	`import os`
`in`	Part of `for-loops`. Also a test of X in Y.	`for X in Y: pass` also `1 in [1] == True`
`is`	Like `==` to test equality	1 is 1 == True
`lambda`	Create a short anonymous function	`s = lambda y: y`
`not`	Logical not	`not True == False`
`or`	Logical or	`True or False == True`
`pass`	This block is empty.	`def empty(): pass`
`print`	Print this string.	`print 'this string'`
`raise`	Raise an exception when things go wrong.	`raise ValueE rro r("N o")`

Keywords (cont)

`return`	Exit the function with a return value.	`def X(): return`
`try`	Try this block, and if exception, go to `except`.	`try: pass`
`while`	While loop	`while X: pass`
`with`	With an expression as a variable do.	`with X as Y: pas`
`yield`	Pause here and return to caller.	`def x(): yield Y`

Data Types

Type	Description	Example
`True`	True boolean value	`True or False =`
`False`	False boolean value.	`False and True`
`None`	Represents "nothing" or "no value".	x = None
`strings`	Stores textual info	`x = " hel lo"`
`numbers`	Stores integers	`i = 100`
`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 == '"'

String Formats (cont)

%r	Repr format (debugging format).	" %r" % int == " <type 'int'> "
%s	String format.	"%s there" % 'hi' == 'hi there'
%%	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