

Specific Language Removals in 3.0

In addition to extensions, a number of language tools have been removed in 3.0 in an effort to clean up its design. Table P-2 summarizes the changes that impact this book, covered in various chapters of this edition. Many of the removals listed in Table P-2 have direct replacements, some of which are also available in 2.6 to support future migration to 3.0

Removed | Replacement

reload(M) | imp.reload(M) (or exec)

*apply(f, ps, ks) | f(*ps, **ks)*

`X` | repr(X)

X <> Y | X != Y

long | int

9999L | 9999

D.has_key(K) | K in D (or D.get(key) != None)

raw_input | input

old input | eval(input())

xrange | range

file | open (and io module classes)

X.next | X.__next__, called by next(X)

X.__getslice__ | X.__getitem__ passed a slice object

`X.__setslice__` | `X.__setitem__` passed a slice object
`reduce` | `functools.reduce` (or loop code)
`execfile(filename)` | `exec(open(filename).read())`
`exec open(filename)` | `exec(open(filename).read())`
`0777` | `0o777`
`print x, y` | `print(x, y)`
`print >> F, x, y` | `print(x, y, file=F)`
`print x, y,` | `print(x, y, end='')`
`u'ccc'` | `'ccc'`
`'bbb'` for byte strings | `b'bbb'`
`raise E, V` | `raise E(V)`
`except E, X:` | `except E as X:`
`def f((a, b)):` | `def f(x): (a, b) = x`
`file.xreadlines` | `for line in file:` (or `X=iter(file)`)
`D.keys()`, etc. as lists | `list(D.keys())` (dictionary views)
`map()`, `range()`, etc. as lists | `list(map())`, `list(range())` (built-ins)
`map(None, ...)` | `zip` (or manual code to pad results)
`X=D.keys(); X.sort()` | `sorted(D)` (or `list(D.keys())`)
`cmp(x, y)` | `(x > y) - (x < y)`
`X.__cmp__(y)` | `--lt--`, `--gt--`, `--eq--`, etc
`X.__nonzero__` | `X.__bool__`
`X.__hex__`, `X.__oct__` | `X.__index__`
Sort comparison functions | Use `key=transform` or `reverse=True`