Home > Articles > Programming > Python

Operators and Expressions in Python

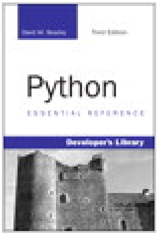
By David M. Beazley

Mar 31, 2006

Contents Print Share This

< Back **Page 7** of 11 Next >

This chapter is from the book



Python Essential Reference, 3rd Edition

Learn More Buy

Type Conversion

Sometimes it's necessary to perform conversions between the built-in types. To convert between types you simply use the type name as a function. In addition, several built-in functions are supplied to perform special kinds of conversions. All of these functions return a new object representing the converted value.

Function	Description
`int(x [,base])`	Converts x to an integer. base specifies the base if x is a string.
`long(x [,base] )`	Converts x to a long integer. base specifies the base if x is a string.
`float(x)`	Converts x to a floating-point number.
`complex(real [,imag])`	Creates a complex number.
`str(x)`	Converts object x to a string representation.
`repr(x)`	Converts object x to an expression string.
`eval(str)`	Evaluates a string and returns an object.
`tuple(s)`	Converts s to a tuple.

Related Resources

Store Articles Blogs

Data Science with Python and R LiveLessons (Anaconda Video Series)

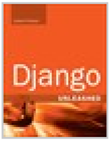
By Ian Stokes-Rees

Online Video $199.99

Python Programming Language LiveLessons

By David Beazley

Online Video $239.99



Django Unleashed

By Andrew Pinkham

Book $39.99

See All Related Store Items

`list(s)`	Converts s to a list.
`set(s)`	Converts s to a set.
`dict(d)`	Creates a dictionary. d must be a sequence of `(key,value)` tuples.
`frozenset(s)`	Converts s to a frozen set.
`chr(x)`	Converts an integer to a character.
`unichr(x)`	Converts an integer to a Unicode character.
`ord(x)`	Converts a single character to its integer value.
`hex(x)`	Converts an integer to a hexadecimal string.
`oct(x)`	Converts an integer to an octal string.

You also can write the `repr(x)` function using backquotes as `´x´`. Note that the `str()` and `repr()` functions may return different results. `repr()` typically creates an expression string that can be evaluated with `eval()` to re-create the object. On the other hand, `str()` produces a concise or nicely formatted representation of the object (and is used by the `print` statement). The `ord()` function returns the integer ordinal value for a standard or Unicode character. The `chr()` and `unichr()` functions convert integers back into standard or Unicode characters, respectively.

To convert strings back into numbers and other objects, use the `int()`, `long()`, and `float()` functions. The `eval()` function can also convert a string containing a valid expression to an object. Here's an example:

```
a = int("34")             # a = 34
b = long("0xfe76214", 16) # b = 266822164L (0xfe76214L)
b = float("3.1415926")    # b = 3.1415926
c = eval("3, 5, 6")       # c = (3,5,6)
```

In functions that create containers (`list()`, `tuple()`, `set()`, and so on), the argument may be any object that supports iteration that is used to generate all the items used to populate the object that's being created.

✚ Share This Save To Your Account

< Back **Page 7** of 11 Next >