

Scientific Programming in Python
Notes
7 October 2009

Operator overloading

Arithmetic

x + y	<code>__add__(self, y)</code>
x - y	<code>__sub__(self, y)</code>
x * y	<code>__mul__(self, y)</code>
x / y	<code>__div__(self, y)</code>
x % y	<code>__mod__(self, y)</code>
x // y	<code>__floordiv__(self, y)</code>
x ** y	<code>__pow__(self, y)</code>
-x	<code>__neg__(self)</code>
+x	<code>__pos__(self)</code>

Comparison

x == y	<code>__eq__(self, y)</code>
x != y	<code>__ne__(self, y)</code>
x > y	<code>__gt__(self, y)</code>
x < y	<code>__lt__(self, y)</code>
x >= y	<code>__ge__(self, y)</code>
x <= y	<code>__le__(self, y)</code>
value in x	<code>__contains__(self, value)</code>

Assignment

x += y	<code>__iadd__(self, y)</code>
x -= y	<code>__isub__(self, y)</code>
x *= y	<code>__imul__(self, y)</code>
x /= y	<code>__idiv__(self, y)</code>
x %= y	<code>__imod__(self, y)</code>
x //= y	<code>__ifloordiv__(self, y)</code>
x **= y	<code>__ipow__(self, y)</code>

Functions

len(x)	<code>__len__(self)</code>
abs(x)	<code>__abs__(self)</code>
int(x)	<code>__int__(self)</code>
float(x)	<code>__float__(self)</code>
complex(x)	<code>__complex__(self)</code>
str(x)	<code>__str__(self)</code>

Indexing

x[i]	<code>__getitem__(self, i)</code>
x[i:j]	<code>__getslice__(self, i, j)</code>
x[i] = value	<code>__setitem__(self, i, value)</code>
x[i:j] = value	<code>__setslice__(self, i, j, value)</code>
del x[i]	<code>__delitem__(self, i)</code>
del x[i:j]	<code>__delslice__(self, i, j)</code>