1.1 Python 3.x Magic Methods
c__ini__() Initialize the object

1.2 Conversion

c__str__()	str(c)
c__rep__()	repr(c)
c__forma__(spec)	"{0:spec}".format(c)
c__complex__()	complex(c)
c__int__()	to int
c.__float__()	to float
c.__bool__()	to bool
c.__bytes__()	like __str__ but for byte arrays

1.3 Iteration

c.__iter__()	iter(c)
c.__next__()	next(c)
c.__reversed__()	reverse iterator

1.4 Attribute access

c.__getattr__(a)	c.a, fallback
c.__getattribute__(a)	c.a, unconditional
c.__setattr__(a, v)	c.a = v
c.__delattr__(a)	del c.a
c.__dir__()	list attributes (for dir())
c.__dict__{}	access class dict directly

1.5 Calling

c.__call__(...)	c(...)

1.6 Immutable container

c.__contains__(e)	e in c
c.__len__()	Length (for len())
c.__getitem__(x)	c[x]

1.7 Mutable container

c.__setitem__(x, v)	c[x] = v
c.__delitem__(x)	del c[x]
c.__missing__(x)	c[x] if c not in x

1.8 Emulating numbers

c.__add__(x)	c + x
c.__sub__(x)	c - x
c.__mul__(x)	c * x
c.__mod__(x)	c % x
c.__floordiv__(x)	c // x
c.__divmod__(x)	divmod(c, x)
c.__pow__(x)	c ** x
c.__rshift__(x)	c >> x
c.__lshift__(x)	c << x
c.__and__(x)	c & x
c.__xor__(x)	c ^ x
c.__or__(x)	c	x
c.__radd__(x)	x + c
etc.	
c.__iadd__(x)	c += x
etc.	
c.__neg__(x)	-c
c.__pos__()	+c
c.__abs__()	abs(c)
c.__invert__()	~c

c.__round__(n)	round(c, n)
c.__floor__()	used by math.floor()
c.__ceil__()	used by math.ceil()
c.__trunc__()	used by math.trunc()
c.__index__()	lst[c]
c.__oct__()	oct(c)
c.__hex__()	hex(c)
c.__matmul__()	matrix multiplication with @

1.9 Comparisons

c._lt_(o)	c < o
c.__gt__(o)	c > o
c.__le__(o)	c <= o
c.__ge__(o)	c >= o
c.__eq__(o)	c == o
c.__ne__(o)	c != o

1.10 Context handlers

c.__enter__(o)	with c: (entry)
c.__exit__(o, exp_type, exp_val, trace)	with c: (exit)

1.11 Low-level stuff
Read the documentation!

c.__new__(cls, …)	object construction
c.__del__()	clean up object
c.__slots__()	limit attributes
c.__hash__()	compute hash value
c.__sizeof__()	sys.getsizeof(c)
c.__metaclass__()	the class of the class
c__prepare__()	used for metaclasses
c.__instancecheck__(x)	isinstance(x, c)
c.__subclasscheck__(x)	issubclass(x, c)

1.12 Pickling

c.__getstate__()	pickle.dump(pkl_file, self)
c.__setstate__()	data = pickle.load(pkl_file)

There few more magic methods for advanced pickling

1.13 Copying

c.__copy__()	copy.copy() for c
c.__deepcopy__(memodict={})	copy.deepcopy() for c

1.14 Descriptor Objects
Class with one or more methods like this:

c.__get__(instance, owner)	when value is retrieved
c.__set__(instance, value)	when value is changed
c.__delete__(self, instance)	when value is deleted

owner is the owner class itself

1.15 Built-ins

__all__[]	list of default module exports
__doc__	doc strings
__name__	module name or "__main__"
__qualname__	full name of the class
__module__	module name
__defaults__{}	default argument values
__kwdefaults__{}	defaults for keyword-only parameters
__code__	code object of compiled function body
__globals__{}	function's global variables
__version__	module version in string format