

“Special Methods” for Python Classes

Each of these is called automatically
when certain situations occur.

```
__init__(self, <optional additional args >)
```

This is called when an object of the class is constructed. If the class name is C we make a new object of the class with

```
C( <value for each optional arg> )
```

```
__str__(self)
```

This should return a string. It is called automatically whenever the system wants a string representation for an object of the class.

If `x` is an object of the class, this is returned by

```
str(x)
```

And it is what is printed by

```
print(x)
```

```
__add__(self, x)
```

```
__sub__(self, x)
```

```
__mul__(self, x)
```

```
__div__(self, x)
```

These allow the arithmetic operators + - * / to be used with objects of a class. Each should return a new object. In the expression a op b self is a and x is b.

```
__lt__(self, x)
```

```
__le__(self, x)
```

```
__gt__(self, x)
```

```
__ge__(self, x)
```

```
__eq__(self, x)
```

```
__ne__(self, x)
```

These allow you to use the comparison operators < <= > >= == and != with objects of the class. Each should return True or False. To sort a list of objects you should have __lt__(self, x) defined for the class.