

### Class - Syntax

```
class Dog(object):
    living = True
    age = 0
    def __init__(self, name, sex):
        self.name = name
        self.sex = sex
    def bark(self):
        print(f"{self.name} barked!")

luke = Dog('Luke', 'Male')
print(f"Name: {luke.name}")
print(f"Sex: {luke.sex}")
print(f"Living: {luke.living}")
print(f"Age: {luke.age}")
luke.bark()
```

Name: Luke  
Sex: Male  
Living: True  
Age: 0  
Luke barked!

### Class - Magic Methods

<code>__init__(self, other)</code>	Overrides object creation method
<code>__repr__(self)</code>	Overrides object string representation
<code>__add__(self, other)</code>	Overrides + operator
<code>__sub__(self, other)</code>	Overrides - operator
<code>__mul__(self, other)</code>	Overrides * operator
<code>__floordiv__(self, other)</code>	Overrides // operator
<code>__truediv__(self, other)</code>	Overrides / operator
<code>__mod__(self, other)</code>	Overrides % operator
<code>__pow__(self, other[, modulo])</code>	Overrides ** operator
<code>__lt__(self, other)</code>	Overrides < comparison operator
<code>__le__(self, other)</code>	Overrides <= comparison operator
<code>__eq__(self, other)</code>	Overrides == comparison operator
<code>__ne__(self, other)</code>	Overrides != comparison operator
<code>__ge__(self, other)</code>	Overrides >= comparison operator
<code>__gt__(self, other)</code>	Overrides > comparison operator
<code>__call__(self[, args...])</code>	Overrides () operator

### Class - Magic Methods (cont)

<code>__int__(self)</code>	Overrides <b>int()</b> method
<code>__float__(self)</code>	Overrides <b>float()</b> method
<code>__str__(self)</code>	Overrides <b>str()</b> method
<code>__abs__(self)</code>	Overrides <b>abs()</b> method
<code>__len__(self)</code>	Overrides <b>len()</b> method
<code>__contains__(self, item)</code>	Overrides <b>in</b> keyword behavior

### OOP - Inheritance

```
class Vehicle(object):
    def honk(self):
        print('Honk honk')

class Car(Vehicle):
    def accelerate(self):
        print('Vroom')

honda = Car()
honda.honk()
honda.accelerate()
```

Honk honk  
Vroom