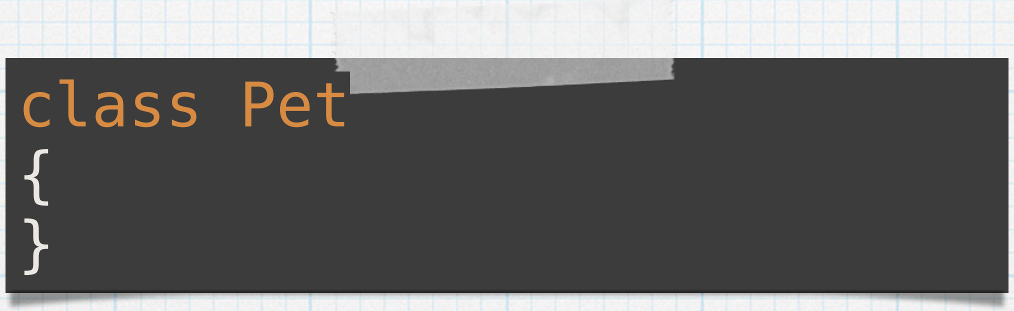
Object Oriented Programming

Classes and Objects

Class

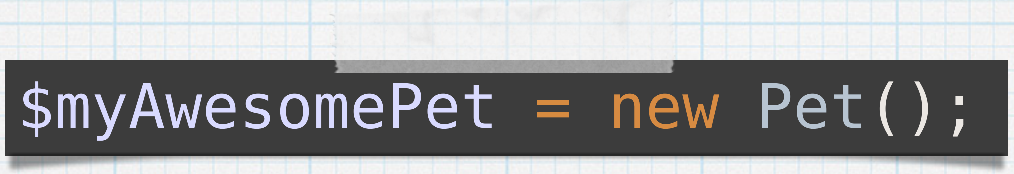
- The class is a recipe or a blueprint



```
class Pet
{
}
```

Object

- An object is a tangible thing

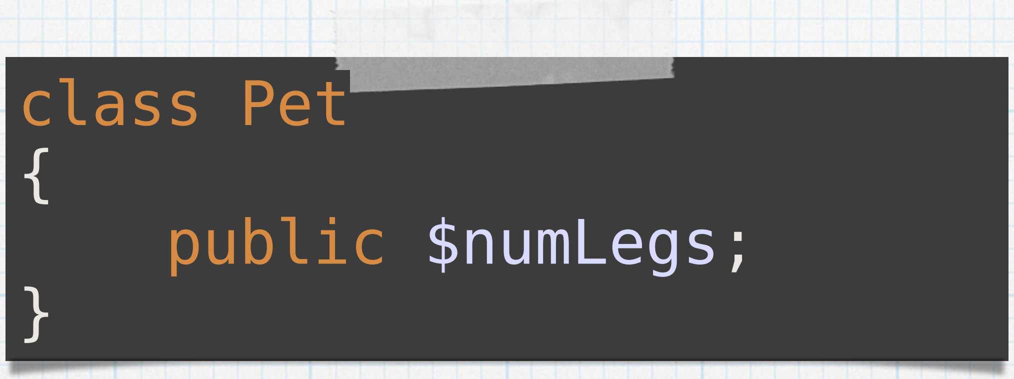


```
$myAwesomePet = new Pet();
```

- $myAwesomePet is an object, an "instance" of the Pet class

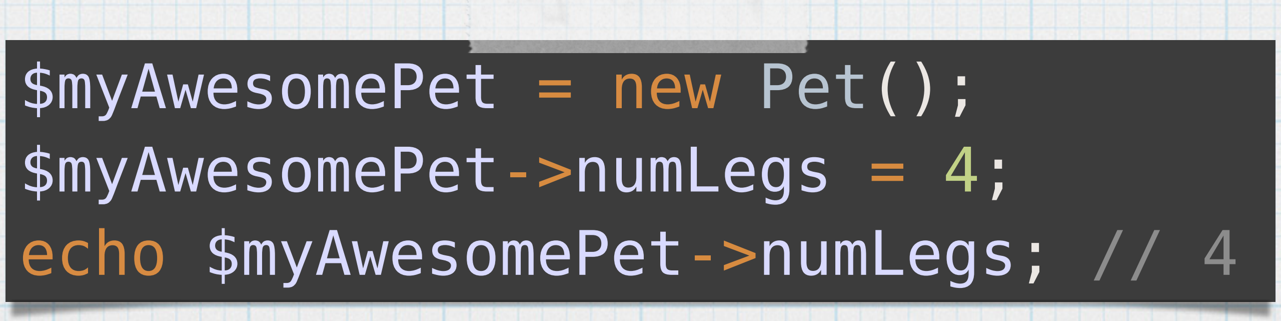
Object Properties

- Properties can be defined in the class



```
class Pet
{
    public $numLegs;
}
```

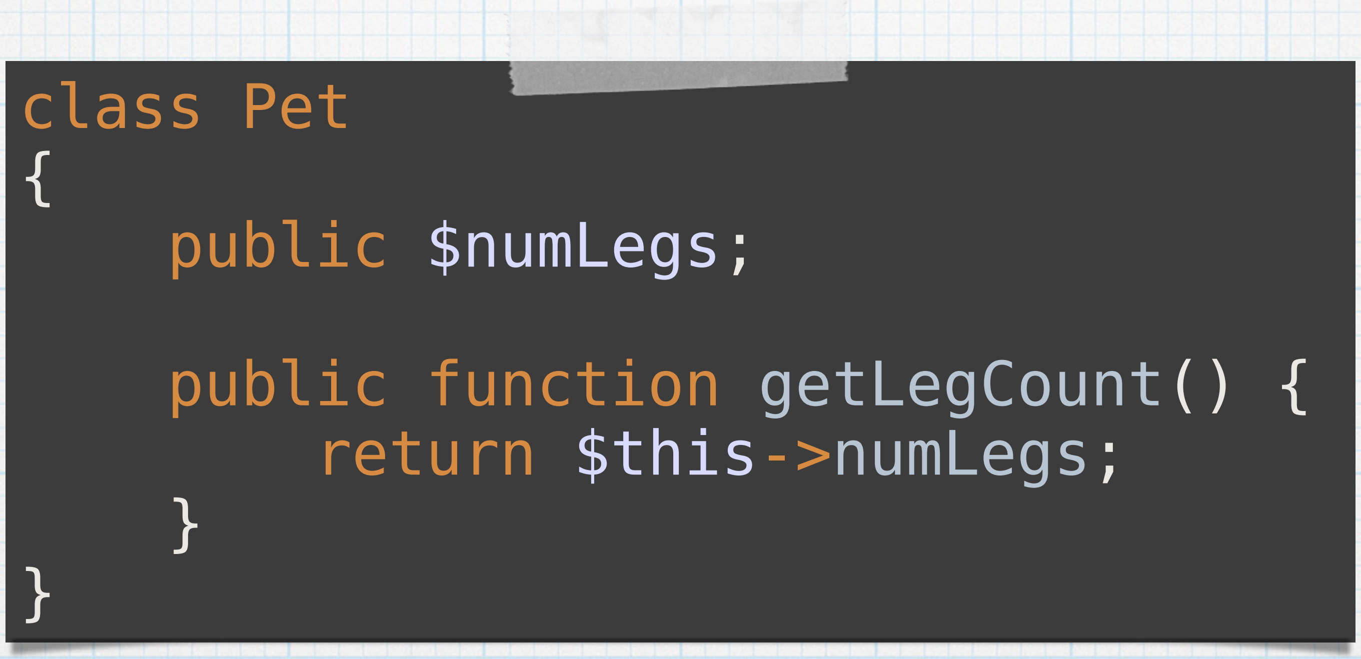
- And set on an object



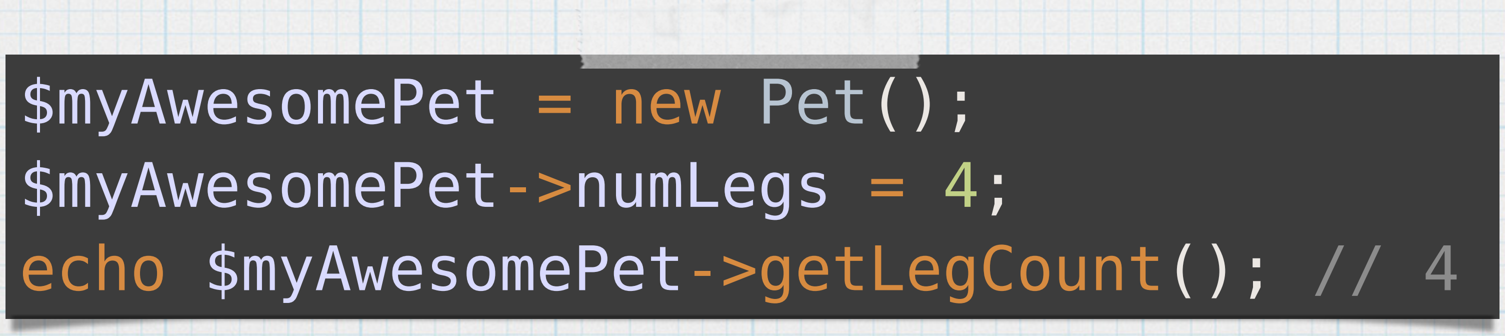
```
$myAwesomePet = new Pet();
$myAwesomePet->numLegs = 4;
echo $myAwesomePet->numLegs; // 4
```

Object Methods

- These are functions with a fancy name



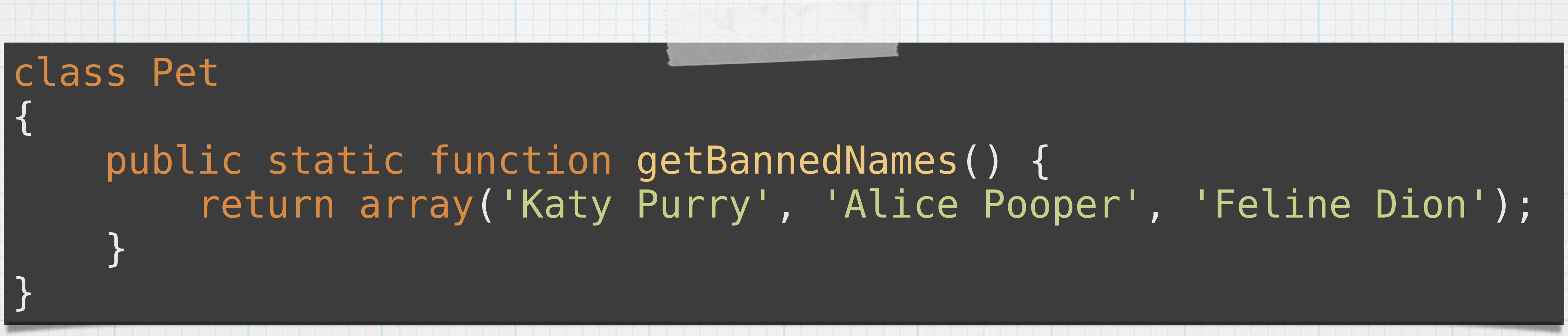
```
class Pet
{
    public $numLegs;

    public function getLegCount() {
        return $this->numLegs;
    }
}
```



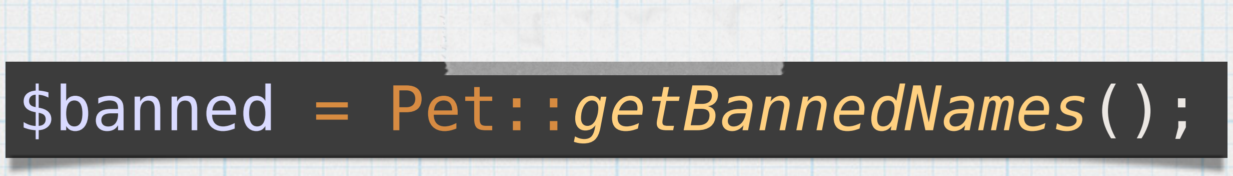
```
$myAwesomePet = new Pet();
$myAwesomePet->numLegs = 4;
echo $myAwesomePet->getLegCount(); // 4
```

"Static" Methods

- We can call static methods without instantiating a class!



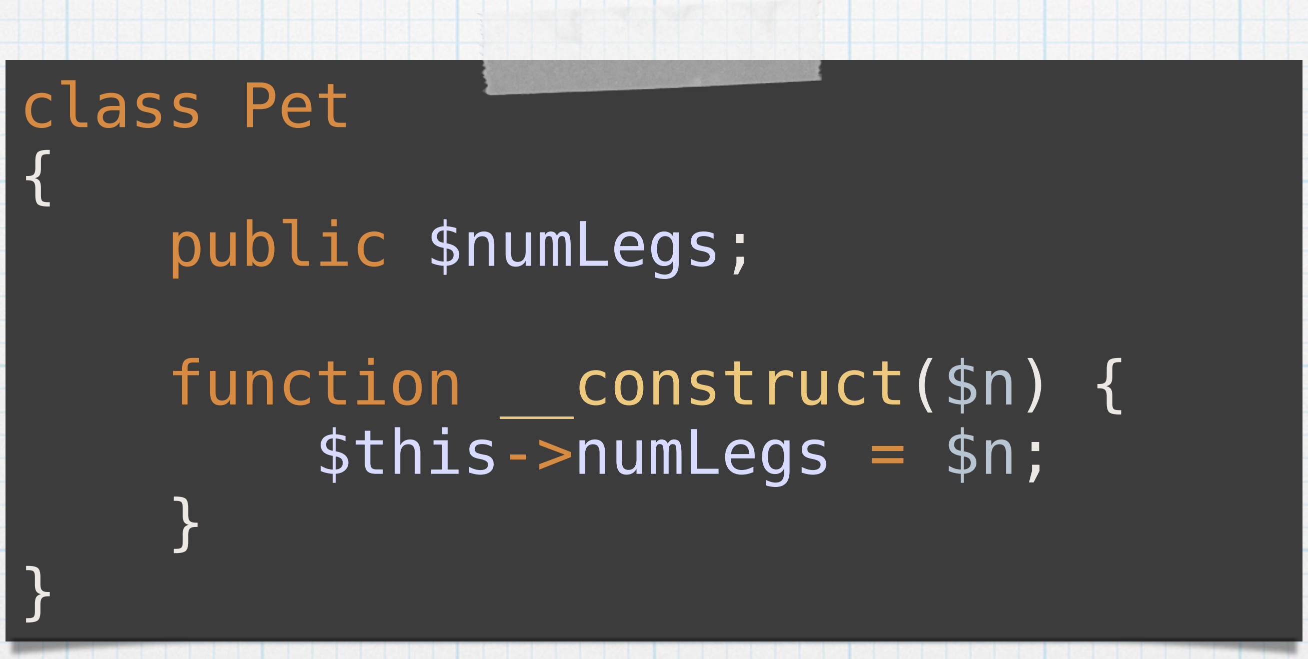
```
class Pet
{
    public static function getBannedNames() {
        return array('Katy Purry', 'Alice Pooper', 'Feline Dion');
    }
}
```



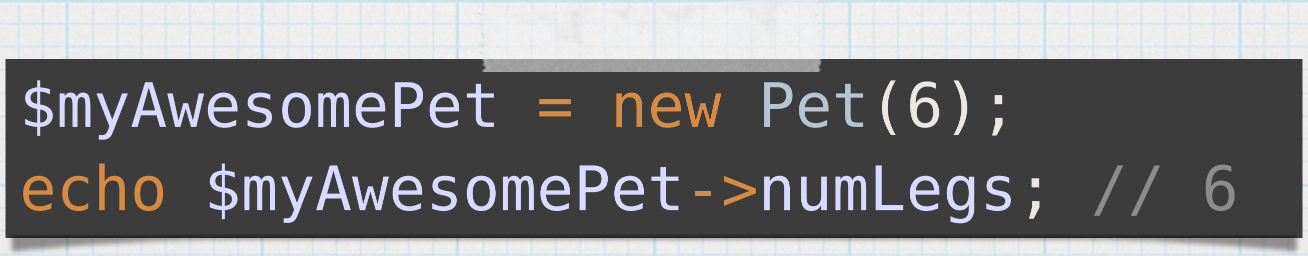
```
$banned = Pet::getBannedNames();
```

Object Constructor

- Useful for initializing stuff when an object is created



```
class Pet
{
    public $numLegs;

    function __construct($n) {
        $this->numLegs = $n;
    }
}
```

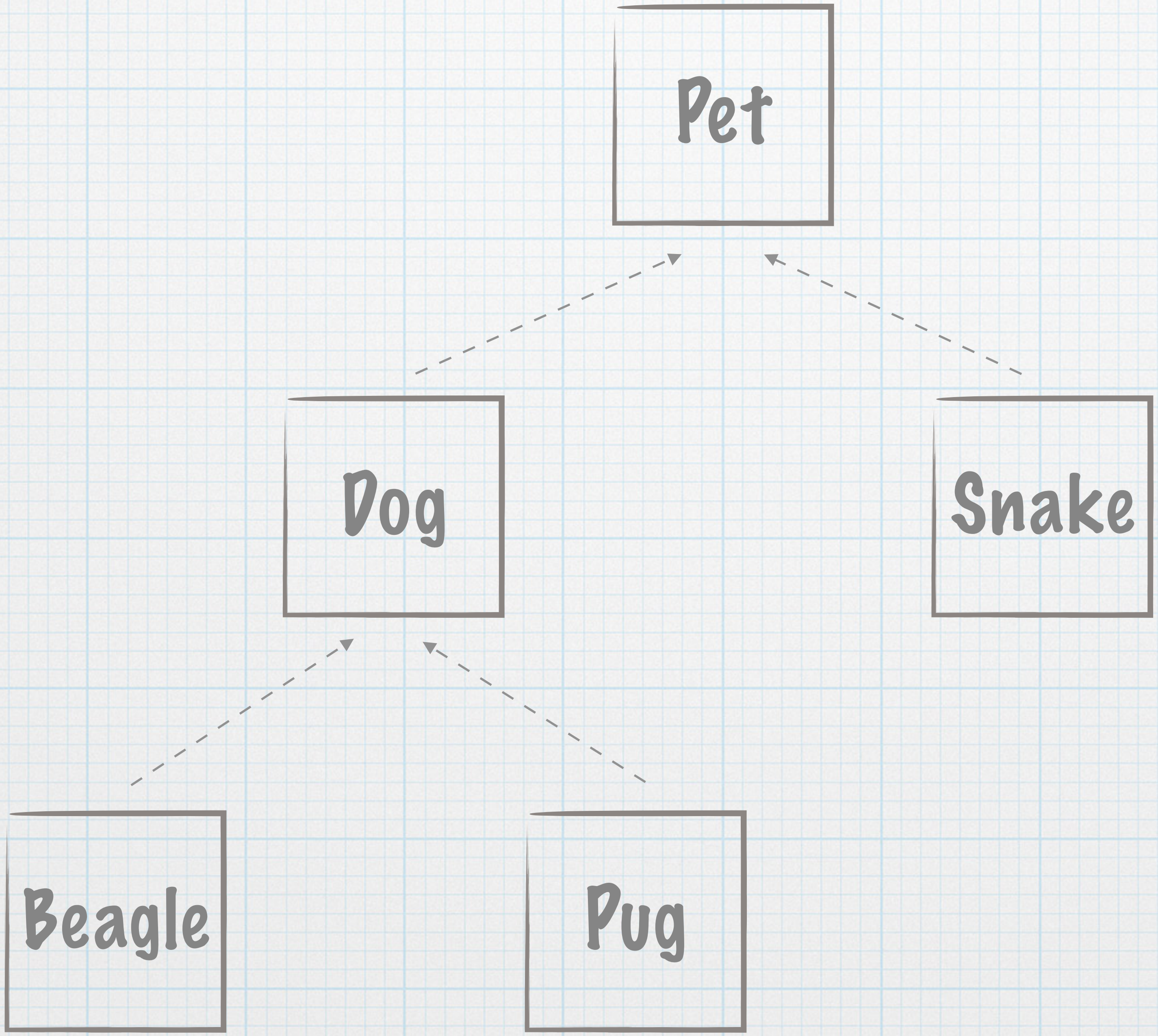


```
$myAwesomePet = new Pet(6);
echo $myAwesomePet->numLegs; // 6
```

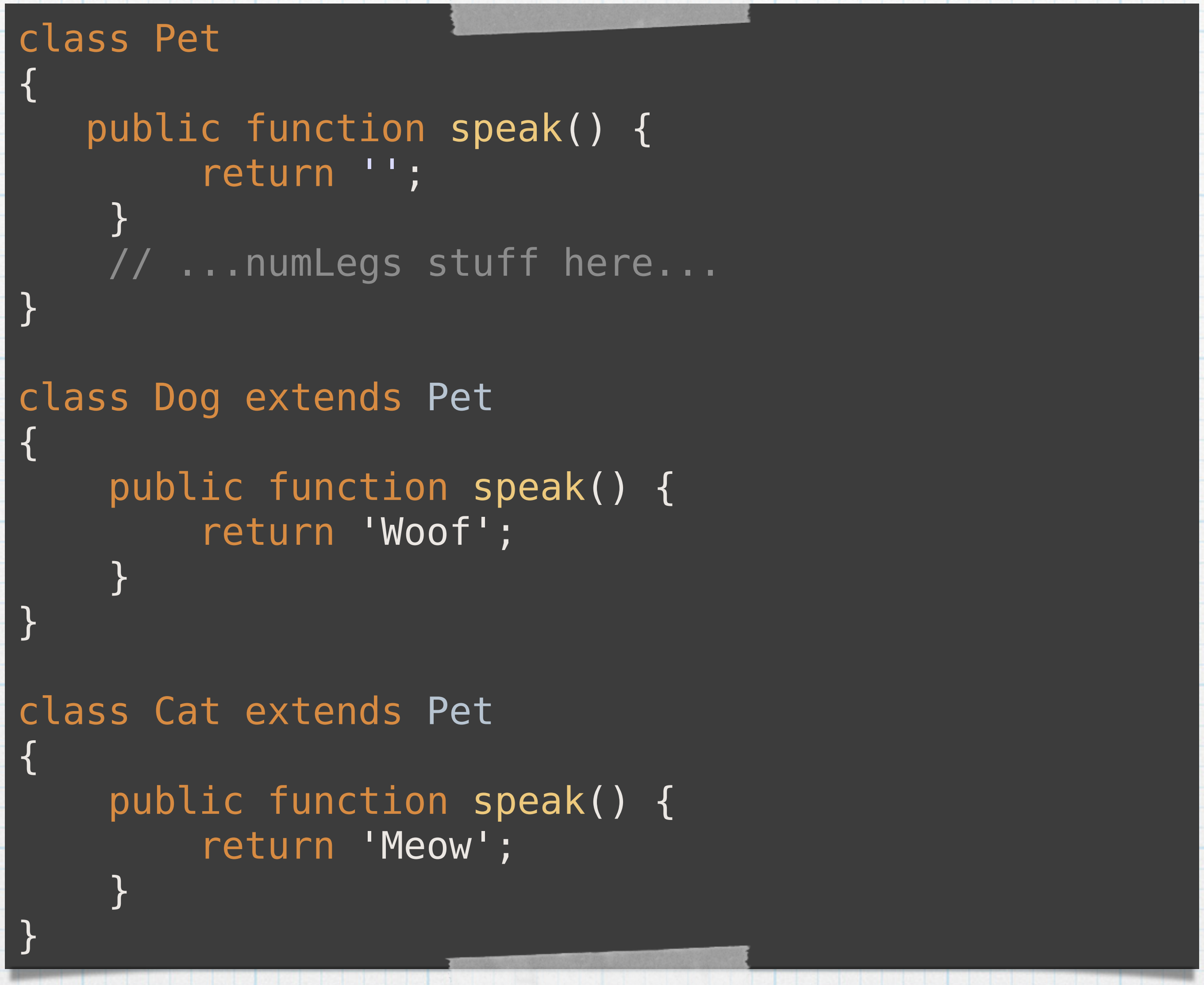
Inheritance

- A class can "inherit" from a parent class
- The child class then has all the properties and methods of the parent class
- The child class can still have its own properties and methods
- This saves us from duplicating code
- A class can only have one parent (in PHP)
- A class can have many children

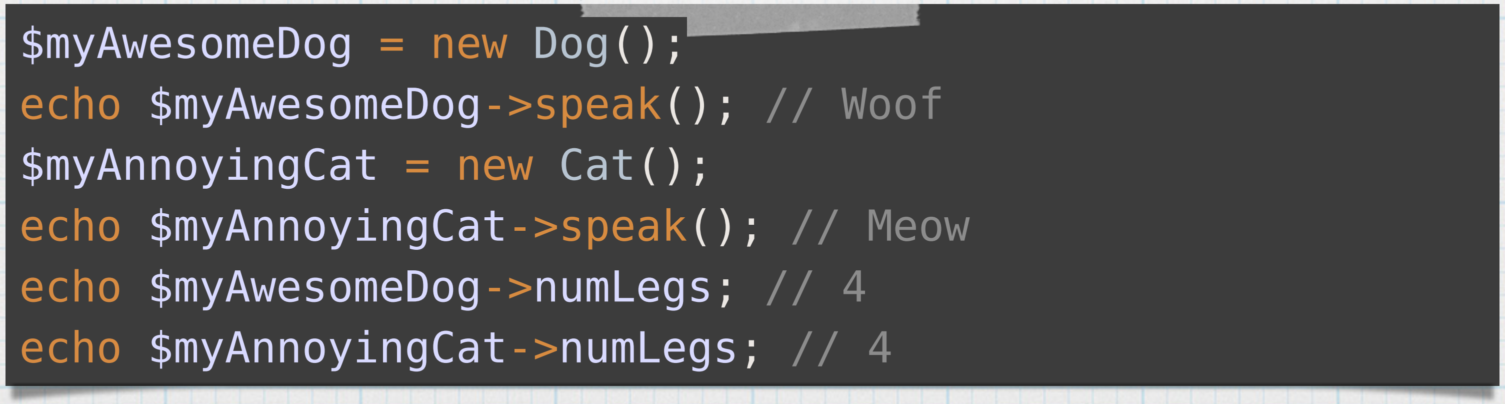
Inheritance



Inheritance



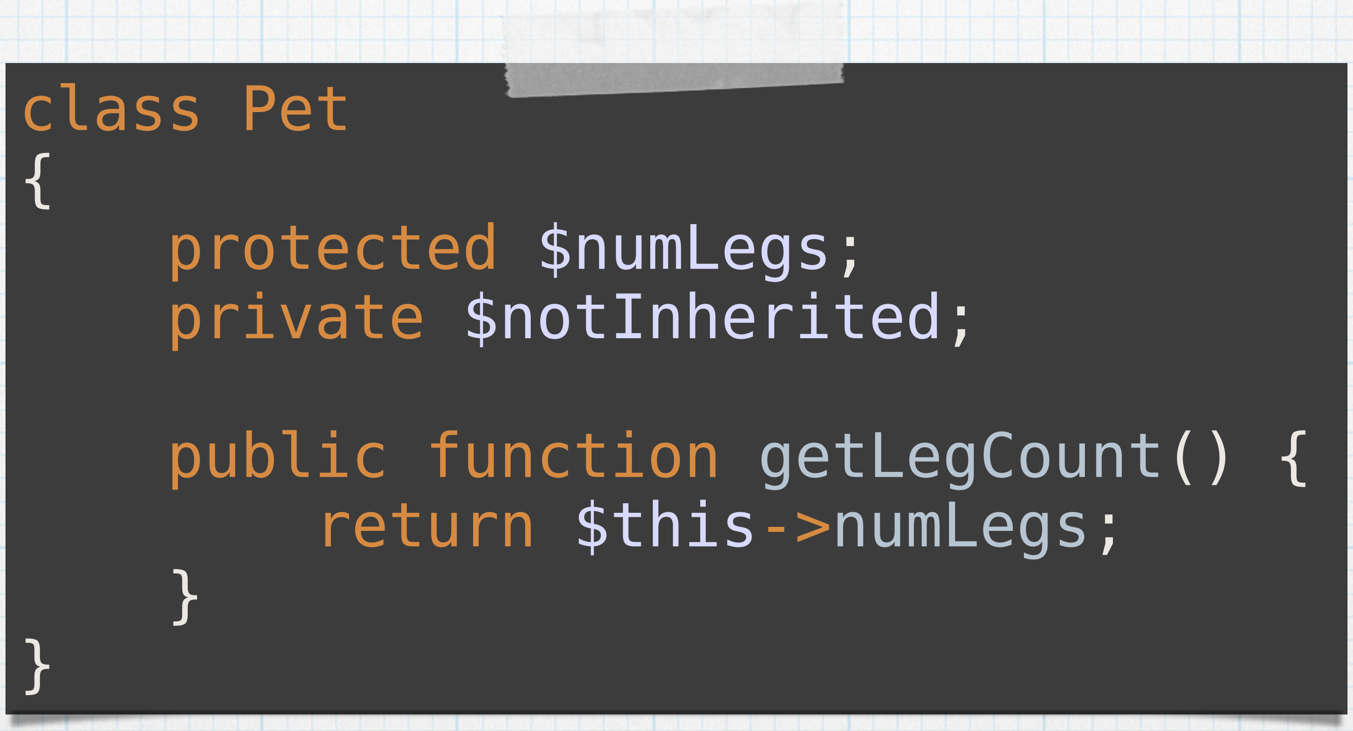
```
class Pet
{
   public function speak() {
        return '';
    }
    // ...numLegs stuff here...
}

class Dog extends Pet
{
    public function speak() {
        return 'Woof';
    }
}

class Cat extends Pet
{
    public function speak() {
        return 'Meow';
    }
}
```



```
$myAwesomeDog = new Dog();
echo $myAwesomeDog->speak(); // Woof
$myAnnoyingCat = new Cat();
echo $myAnnoyingCat->speak(); // Meow
echo $myAwesomeDog->numLegs; // 4
echo $myAnnoyingCat->numLegs; // 4
```

Access Modifiers

- public, private and protected



```
class Pet
{
    protected $numLegs;
    private $notInherited;

    public function getLegCount() {
        return $this->numLegs;
    }
}
```