

## Introduction to Classes.

### Classes

In short, a class is a code template used to generate objects. You declare a class with the class keyword and an arbitrary class name. Class names can be any combination of numbers and letters, although they must not begin with a number. The code associated with a class must be enclosed within braces.

```
class Dog {  
    //class body  
    //class body  
    //class body  
}
```

If a class is a template for generating objects, it follows that an object is data that has been structured according to the template defined in a class. An object is said to be an instance of its class. It is of the type defined by the class.

```
$dog1 = new Dog();  
$dog2 = new Dog();
```

The *new* operator is used with a class name as its only operand and generates an *instance* of that class; in our example, it generates a Dog object.

### Properties

Classes can define special variables called properties. A property, also known as a member variable, holds data that can vary from object to object. You just have to use the public, protected, private keywords.

```
class Dog {  
    public $name = "default name";  
    public $breed = "best breed";  
    public $gender = "male";  
    public $price = 0;  
}
```

You can access property variables on any object-by-object basis using the characters '->' in conjunction with an object variable and property name.

```
$dog1 = new Dog();  
print $dog1->name;  
// default name
```

### Methods

Methods are what we use to have our objects perform tasks. Methods are special functions declared within a class. A method declaration resembles a function declaration. The big difference being, a method has to be declared within a class.

```
public function myMethod( $argument, $another) {  
    // stuff  
}
```

Methods can accept a number of qualifiers, including a visibility keyword. They can also be public, protected , or private.

```
class Dog {
    public $firstName = "default name";
    public $lastName = "default last name";
    public $gender = "male";
    public $price = 0;

    function getName() {
        return "{$this->firstName}" .
            "{$this->lastName}";
    }
}

$dog1 = new Dog();
$dog1->firstName = "BooBoo";
$dog1->lastName = "Iwao";

print "The dog's name is {$dog1->getName()}.";

//The dog's name is BooBoo Iwao.
```

The `$this` is a mechanism by which a class can refer to an object instance. If this is confusing, think of `$this` like this.

`$this->firstName`

*is the same as*

the `$firstName` property of the current instance.