

String concatenation can be done in several ways in PHP. In many languages, building strings with properties require continuously opening and closing a string (using "").

```
print "Dog_weight is " . $this->dog_weight . ". Dog breed is " . $this->dog_breed . "Dog
color is " . $this->dog_color;
```

The print code line above is valid in PHP. However, as you can see, you have a lot of quotes and many periods. It can be difficult to get everything matched up correctly. The period is a string concatenation character in PHP and would be required, if you choose to use this technique (you would have to use a similar madness in many languages). However, PHP is much friendlier than this.

This pointer—The \$this pointer is used to gain access to properties contained in an object. this indicates that the code wants to retrieve the value contained in a property that exists in the particular object (instance of the class). Soon we will be creating an instance of the class named \$lab. When the code that will exist in the \$lab instance is executed, the \$this pointer will tell the operating system that it wants the value in the property (such as dog_weight) that exists in the \$lab instance only. Note that the format of the statement includes a \$ sign for the \$this pointer but not for the variable (\$this->dog_weight).

You might ask, why do we need the \$this pointer? The simple answer is that you can create a property that exists for every instance of the class (called a static property). If this type of property changed, it would change for all the instances of the class. Our private properties only change for the particular instance (\$lab) of the class in which it was referenced.

```
print "Dog weight is $this->dog_weight. Dog breed is $this->dog_breed. Dog color is
$this->dog_color.";
```

PHP allows you to place properties within strings (quotes). This allows you to use fewer periods and quotes (and maybe reduce pulling too many hairs from your head).

For examples of the \$this pointer, visit:

<http://php.net/manual/en/language.oop5.basic.php>

For videos of the \$this pointer, visit:

<https://www.thenewboston.com/videos.php?cat=11&video=17177>

Now that you have the code needed to produce an output, you need to add a method in the class to execute the print line. All “actions” that take place in a class must be included in a method. Methods are created in a similar style as classes (except they are actually contained within the classes). Methods are declared using the keyword `function` followed by a method name and `()`. It is common practice for method names to be lowercase, although PHP will accept uppercase characters. The `_` can also be included at the beginning or within the method name. All code with a method is contained within `{}`.

Example 3-4. Basic class structure with properties and a method in `dog.php` file

```
<?php
class Dog
{
    private $dog_weight = 0;
    private $dog_breed = "no breed";
    private $dog_color = "no color";
    private $dog_name = "no name";
```