Assuming that the XML file is formatted as shown here, the output includes:

```
Array ( [dog] => Array ( [0] => SimpleXMLElement Object ( [dog_name] => Woff [dog_weight] => 12
[dog_color] => Yellow [dog_breed] => Lab ) [1] => SimpleXMLElement Object ( [dog_name] => Sam
[dog_weight] => 10 [dog_color] => Brown [dog_breed] => Lab ) ) )
```

A combination of multidimensional arrays and SimpleXML objects have been created. This does not provide useful data that can easily be manipulated. However, you can use JSON methods to trick PHP into creating a multidimensional *associate array*.

```
class dog_data
{
        function __construct()
        {
                $xmlfile = file_get_contents(get_dog_application("datastorage"));
                $xmlstring = simplexml:load_string($xmlfile);
                $json = json_encode($xmlstring);
                print_r($json);
        }
}
{"dog":[{"dog_name":"Woff","dog_weight":"12","dog_color":"Yellow","dog_breed":"Lab"},{"dog_name"
:"Sam","dog_weight":"10","dog_color":"Brown","dog_breed":"Lab"}]}
```

Using the PHP `json_encode` method changes the data into well-structured JSON data. You could use one of the several PHP techniques to manipulate JSON data or, with one additional statement (`json_decode`), you can create a well-structured multidimensional associate array.

```
class dog_data
{
        function __construct()
        {
                $xmlfile = file_get_contents(get_dog_application("datastorage"));
                $xmlstring = simplexml:load_string($xmlfile);
                $json = json_encode($xmlstring);
                $dogs_array = json_decode($json,TRUE);
                print_r($dogs_array);
        }
}
Array ( [dog] =>
Array (
[0] => Array ( [dog_name] => Woff [dog_weight] => 12 [dog_color] => Yellow [dog_breed] => Lab )
[1] => Array ( [dog_name] => Sam [dog_weight] => 10 [dog_color] => Brown [dog_breed] => Lab ) ) )
```

As you can see, there is no longer a mixture of arrays and SimpleXML objects. An associate array has been created that uses keywords instead of numerical values for subscripts (indexes). In the previous example, an array called `"dog"` has been created with two rows (each row is represented by an array). In each row, the columns (cells) are referenced by a column name (`dog_name`, `dog_weight`, `dog_color`, and `dog_breed`) instead of indexes (0, 1, 2, 3). These rows and columns can be manipulated using some of the techniques you have seen in previous chapters.

www.it-ebooks.info