

hidden property (on the HTML form) with the same name (`index`). Since `index` is a property and not a `div` tag, the JavaScript value `property` must be used to set the value of `index`.

`obj` is the JSON object (with all the dogs) that was created when the AJAX call occurred. Information can be retrieved from a JSON object in a very similar way as PHP associative arrays. The `obj` object is similar to the multidimensional `dogs` array shown in previous chapters. The top array is the `dogs` “array”. In the `dogs` array are the “arrays” for each individual dog. These do not have an associated key name, but have a numerical index. Each dog array contains the individual elements (`dog_name`, `dog_color`, `dog_breed`, and `dog_weight`). The `index` property, set in the previous code line, contains the dog index selected by the user. It will be used to pull the selected dog information from `obj` to populate the form objects.

Since you know the exact location of the data to be retrieved (in `index`), loops are not needed.

`obj.dogs[index].dog_name` uses the JSON object name (`obj`), the top array name (`dogs`), the number of the dog array needed (`index`), and the name of the field required (`dog_name`) to access the required information. Again, the format is similar to pulling information from a PHP associative array. The `dog_name` and `dog_weight` values use this dot notation format to pull the information from the `obj` JSON object and place it in the appropriate text box on the HTML form.

```
dog_color = obj.dogs[index].dog_color;
if(dog_color == "Brown")
    {
        colorbuttons[0].checked = true;
    } else if (dog_color == "Black")
    {
        colorbuttons[1].checked = true;
    } else if (dog_color == "Yellow")
    {
        colorbuttons[2].checked = true;
    } else if (dog_color == "White")
    {
        colorbuttons[3].checked = true;
    }
```

Setting the color requires a little more work. The `dog_color` is pulled from the `obj` object (`dog_color = obj.dogs[index].dog_color;`) and placed into a property (`dog_color`). An `if` statement is then used to determine what color exists in this property. (Yes, JavaScript has a case statement that you could have also used). The `if` statement sets the `checked` property of the correct radio button to `true`, causing the radio button to be selected. Notice that the default value (`'mixed'`) is not included in the `if` statement. If the dog is `'mixed'`, or somehow the color did not have a value in the object, there is no reason to change from the default value (`'mixed'`).

```
dog_breed = obj.dogs[index].dog_breed;
document.getElementById('dog_breed').value = dog_breed;
document.getElementById('update').style.display = "inline";
document.getElementById('delete').style.display = "inline";
document.getElementById('insert').style.display = "none";
}
```