



- ✓ 1. Intro to Objects
- ✓ 2. Objects in Code
- ✓ 3. Quiz: Umbrella (7-1)
- ✓ 4. Objects
- ✓ 5. Object Literals
- 6. Naming Conventions
- 7. Summary of Objects
- 8. Quiz: Menu Items (7-2)
- 9. Quiz: Bank Accounts 1 (7-3)
- 10. Quiz: Bank Accounts 2 (7-4)
- 11. Quiz: Facebook Friends (7-5)
- 12. Quiz: Donuts Revisited (7-6)
- 13. Lesson 7 Summary

## Object-literal notation

```
var sister = {  
  name: "Sarah",  
  age: 23,  
  parents: [ "alice", "andy" ],  
  siblings: ["julia"],  
  favoriteColor: "purple",  
  pets: true  
};
```

The syntax you see above is called **object-literal notation**. There are some important things you need to remember when you're structuring an object literal:

- The "key" (representing a **property** or **method** name) and its "value" are separated from each other by a **colon**
- The **key: value pairs** are separated from each other by **commas**
- The entire object is wrapped inside curly braces **{ }**.

And, kind of like how you can look up a word in the dictionary to find its definition, the **key** in a **key:value** pair allows you to look up a piece of information about an object. Here's are a couple examples of how you can retrieve information about my sister's parents using the object you created.

```
// two equivalent ways to use the key to return its value  
sister["parents"] // returns [ "alice", "andy" ]  
sister.parents // also returns [ "alice", "andy" ]
```

Using **sister["parents"]** is called **bracket notation** (because of the brackets!) and using **sister.parents** is called **dot notation** (because of the dot!).



Dictionary - Caleb Roenigk - Creative Commons

- ✔ 1. Intro to Objects
- ✔ 2. Objects in Code
- ✔ 3. Quiz: Umbrella (7-1)
- ✔ 4. Objects
- ✔ 5. Object Literals
- 6. Naming Conventions
- 7. Summary of Objects
- 8. Quiz: Menu Items (7-2)
- 9. Quiz: Bank Accounts 1 (7-3)
- 10. Quiz: Bank Accounts 2 (7-4)
- 11. Quiz: Facebook Friends (7-5)
- 12. Quiz: Donuts Revisited (7-6)
- 13. Lesson 7 Summary

returns "Sarah paints a picture!" whenever you call it. The syntax for this is pretty much exactly the same as how you defined the properties of the object. The only difference is, the `value` in the `key:value` pair will be a function.

```
var sister = {
  name: "Sarah",
  age: 23,
  parents: [ "alice", "andy" ],
  siblings: ["julia"],
  favoriteColor: "purple",
  pets: true,
  paintPicture: function() { return "Sarah paints!"; }
};

sister.paintPicture();
```

Returns: "Sarah paints!"

and you can access the name of my sister by accessing the `name` property:

```
sister.name
```

Returns: "Sarah"