**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!).

## What about methods?

The sister object above contains a bunch of properties about my sister, but doesn't really say what my sister does. For instance, let's say my sister likes to paint. You might have a paintPicture() method that 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"*