THE // FLATIRON SCHOOL

OBJECTS

- What are they?
- Defining new objects
- Setting values for a properties
- Retrieving values from a property
- Object literal notation
- Creating object methods

Objects

Objects

JavaScript is designed on a simple object-based paradigm. An object is a collection of properties, and a property is association between a name and a value. A value of property can be a function, which is then known as the object's method. In addition to objects that are predefined in the browser, you can define your own objects.

Objects in JavaScript, just as many other programming languages, can be compared to objects in real life. The concept of objects in JavaScript can be understood with real life, tangible objects.

For example…

Objects

```
//Let’s say we want to
create a new empty object
called dog.

var dog = new Object();
```

Objects

```
//Now we can fill the Object with properties made up of a
name and a value.

dog.name = "Fido";

dog.age = 3;


console.log(dog.name);
//prints Fido

console.log(dog.age);
//prints 3
```

Objects

```
//What if we would like to set the age of the dog to
something else?

dog.age = 4;

console.log(dog.age);
//now it prints 4 instead of 3

//How would I remove a property?
delete dog.age; //I like having age though,
so let’s keep it.
```

Objects

```
//Properties can hold numbers, strings, boolean, arrays, as
well as other objects, and functions.
```

```
dog.name = "Fido";
dog.age = 3;
dog.has_fleas = true;
dog.likes = ["fetching", "eating"];
dog.owner = new Object();
dog.owner.first_name = "Sam";
dog.give_bath = function() { this.has_fleas = false; };
```

Objects

```
//Hold up a sec there... Did you say functions? That’s right.
Attaching a function to a property of an object is also referred to
as an object method.

dog.has_fleas = true;

//creating a method of the dog object
dog.have_bath = function() {
    this.has_fleas = false;
};

console.log(dog.has_fleas); //prints true
dog.have_bath(); //run have_bath method
console.log(dog.has_fleas); //prints false
```

Objects

That is a lot of writing and I’m lazy…

Might there be a simpler syntax I can use to create new objects, set properties, and store values that is clean and easy to understand?

Object Literal Notation

Object Literal

```
var dog = {
    name: "Fido",
    age: 3,
    has_fleas: true,
    likes: ["fetching", "eating"],
    dislikes: ["vacuum", "baths"],
    owner: {
        first_name: "Sam",
        last_name: "Redford"
    },
    give_bath: function() {
        this.has_fleas = false;
    }
};
```

I see, so object literal notation uses var n = { } to create the object and then each value is assigned to the property name using : instead of equal symbol. Additionally each property is separated by a comma.

Not so fast! Did you notice that there is never a terminating comma “,” for the last property of an object?

Let’s look again…

Object Literal

```
var dog = {
    name: "Fido",
    age: 3,
    has_fleas: true,
    likes: ["fetching", "eating"],
    dislikes: ["vacuum", "baths"],
    owner: {
        first_name: "Sam",
        last_name: "Redford" //no comma!
    },
    give_bath: function() {
        this.has_fleas = false;
    } //no comma!
};
```

Object Literal

Can I still update values by using the dot operator?

```
dog.age = 4;
```

Yes you can.

Objects

Before you mentioned that JavaScript is itself an object based programming language…

What did you mean by that?

Objects

```
//all var are technically properties of the window object.
var foo = 'bar';
//same as
window.foo = 'bar';

function hello() { alert('Hi there!'); }
//same as
var hello = function() { alert('Hi there!'); }
//same as
window.hello = function() { alert('Hi there!'); }

//so your saying that the creating a function named hello is actually
creating a new method for the window object? Yes that’s right!
```

Objects

Hold the phone! Then does that mean that all the jQuery methods like fadeTo() and addClass were created by attaching functions to the jQuery object? Additionally jQuery itself is a method of the window object?

Yes totally! Whoa dude…

If you don't believe me just take a look at the latest copy of the jQuery production file: http://code.jquery.com/jquery-1.8.2.js

or on a page that has jQuery loaded type in the console
typeof window.$

Objects

I understand now that all variables and functions are actually attached to the window object. That’s cool and all but I’m still not totally clear what the window object itself is…

According to Mozilla Developers Network:
“The window object represents the window itself. The document property of a window points to the DOM document loaded in that window…”

“In a tabbed browser, such as Firefox, each tab contains its own window object.”

The window is kind of like the upper most “thing” with everything else in your program operating inside of it.