

JavaScript - Objects

Dr. Charles Severance

www.wa4e.com

<http://www.wa4e.com/code/javascript-objects>
<http://www.wa4e.com/code/javascript-objects.zip>



Definitions



- **Class** - a template - Dog
- **Method or Message** - A defined capability of a class - bark()
- **Attribute** - A defined data item in a class - color
- **Object or Instance** - A particular instance of a class - Lassie

Image CC-BY 2.0: <https://www.flickr.com/photos/dinnerseries/23570475099>

Terminology: Class



Defines the abstract characteristics of a thing (object), including the thing's characteristics (its **attributes, fields, or properties**) and the thing's behaviors (the things it can do, or **methods, operations, or features**). One might say that a **class** is a **blueprint** or factory that describes the nature of something. For example, the **class** Dog would consist of traits shared by all dogs, such as breed and fur color (characteristics), and the ability to bark and sit (behaviors).

Terminology: Instance



One can have an **instance** of a class or a particular object. The **instance** is the actual object created at runtime. In programmer jargon, the Lassie object is an **instance** of the Dog class. The set of values of the attributes of a particular **object** is called its **state**. The **object** consists of state and the behavior that's defined in the object's class.

Object and Instance are often used interchangeably.

Terminology: Method



An object's abilities. In language, **methods** are verbs. Lassie, being a Dog, has the ability to bark. So `bark()` is one of Lassie's methods. She may have other **methods** as well, for example `sit()` or `eat()` or `walk()` or `save_timmy()`. Within the program, using a **method** usually affects only one particular object; all Dogs can bark, but you need only one particular dog to do the barking

Method and Message are often used interchangeably.

Objects in JavaScript

- The OO pattern in JavaScript is a little different.
- The function is indeed a store and reuse pattern.
- **The function keyword returns a value which is the function itself - it makes a function!**



First-Class Functions

In computer science, a programming language is said to have **first-class functions** if it treats functions as first-class citizens.

Specifically, this means the language supports passing functions as arguments to other functions, returning them as the values from other functions, and assigning them to variables or storing them in data structures.

http://en.wikipedia.org/wiki/First-class_function

Building Objects



```
function PartyAnimal() {  
    this.x = 0;  
    this.party = function () {  
        this.x = this.x + 1;  
        console.log("So far "+this.x);  
    }  
}
```

```
an = new PartyAnimal();
```

```
an.party();  
an.party();  
an.party();
```

This is the template for
making PartyAnimal objects.

Each PartyAnimal
object has a bit of data.

Each PartyAnimal object
has a bit of code.

Create a PartyAnimal
object

Tell the object to run the
party() code.

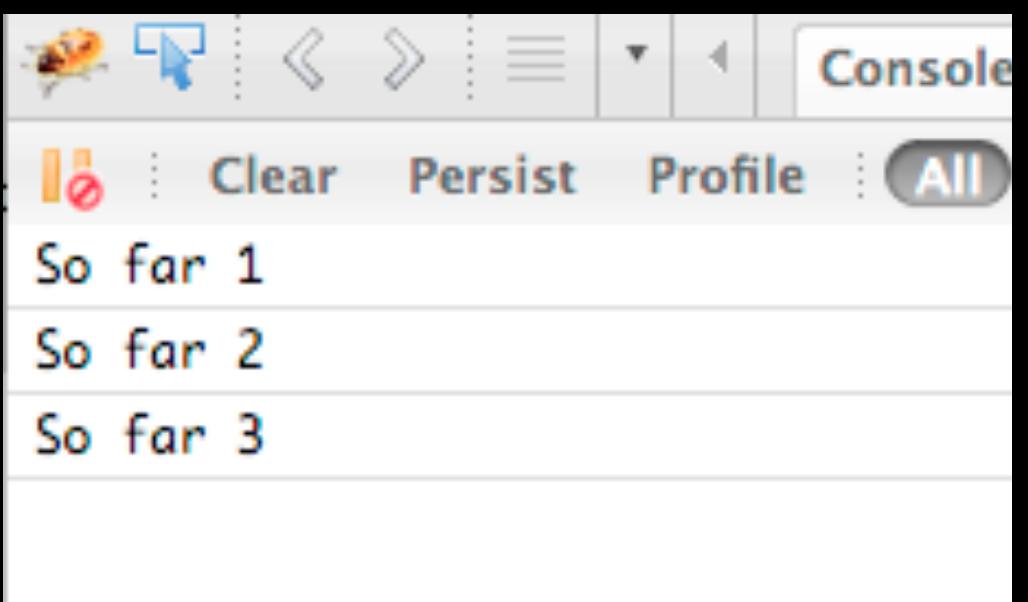
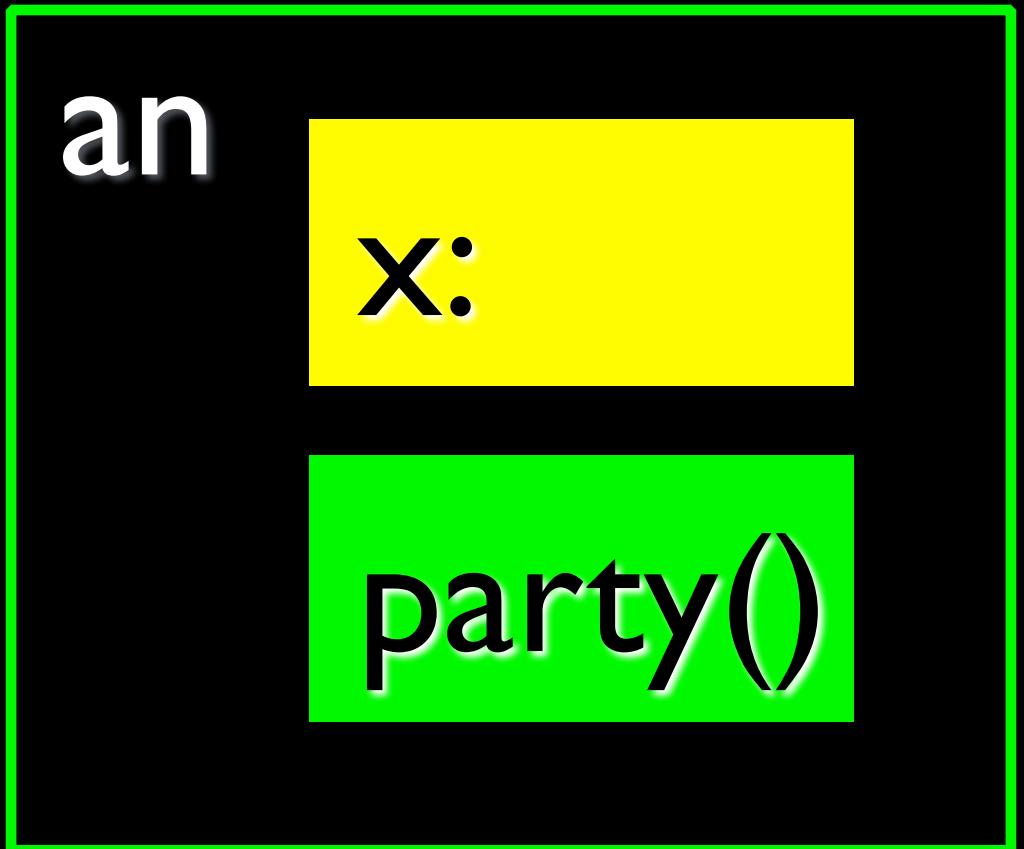
js01.htm

```
function PartyAnimal() {  
    this.x = 0;  
    this.party = function () {  
        this.x = this.x + 1;  
        console.log("So far "+this.x);  
    }  
}
```

```
an = new PartyAnimal();
```

```
an.party();  
an.party();  
an.party();
```

js-01.htm





Object Life Cycle

[http://en.wikipedia.org/wiki/Constructor_\(computer_science\)](http://en.wikipedia.org/wiki/Constructor_(computer_science))

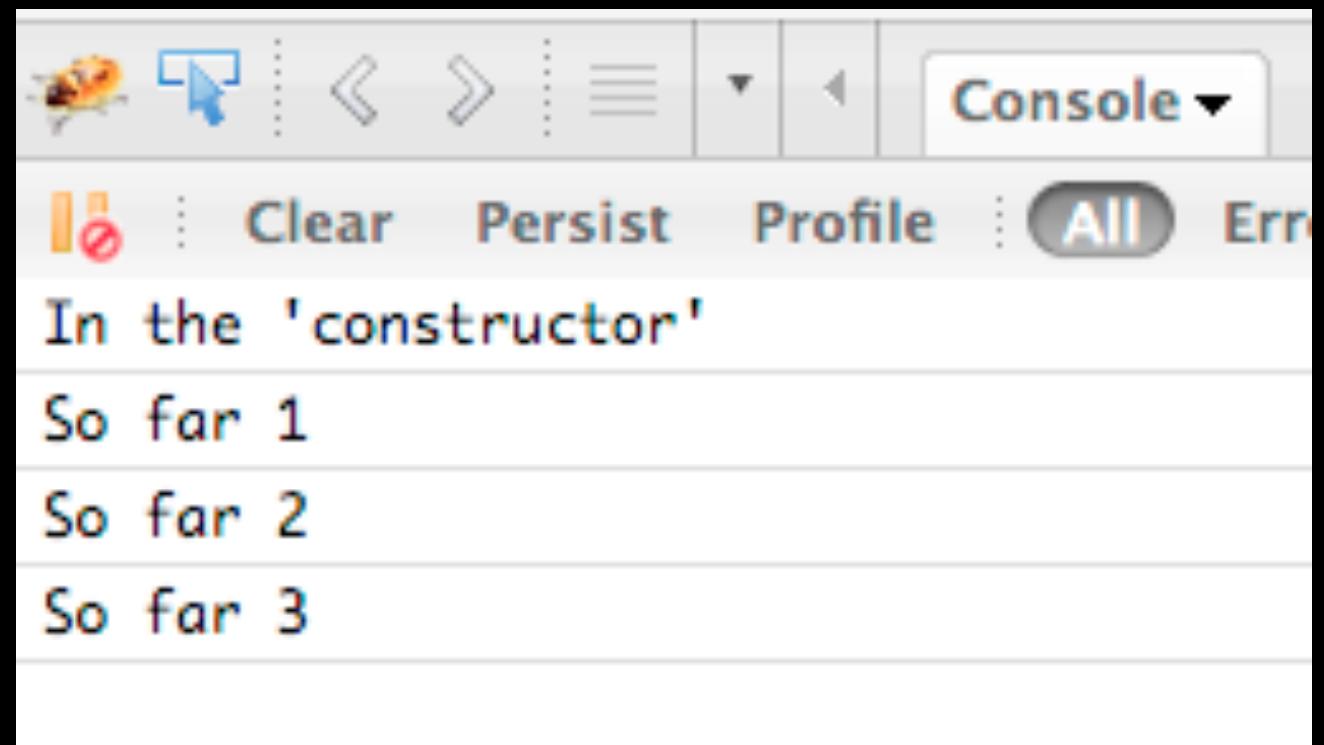
Object Life Cycle

- Objects are created, used, and discarded
- Constructors are implicit in JavaScript - natural
 - A **constructor** in a class is a special block of statements called when an object is created
- Destructors are not provided by JavaScript

[http://en.wikipedia.org/wiki/Constructor_\(computer_science\)](http://en.wikipedia.org/wiki/Constructor_(computer_science))

js03.htm

```
function PartyAnimal() {  
    this.x = 0;  
    console.log("In the 'constructor'");  
    this.party = function () {  
        this.x = this.x + 1;  
        console.log("So far "+this.x);  
    }  
}  
  
an = new PartyAnimal();  
  
an.party();  
an.party();  
an.party();
```



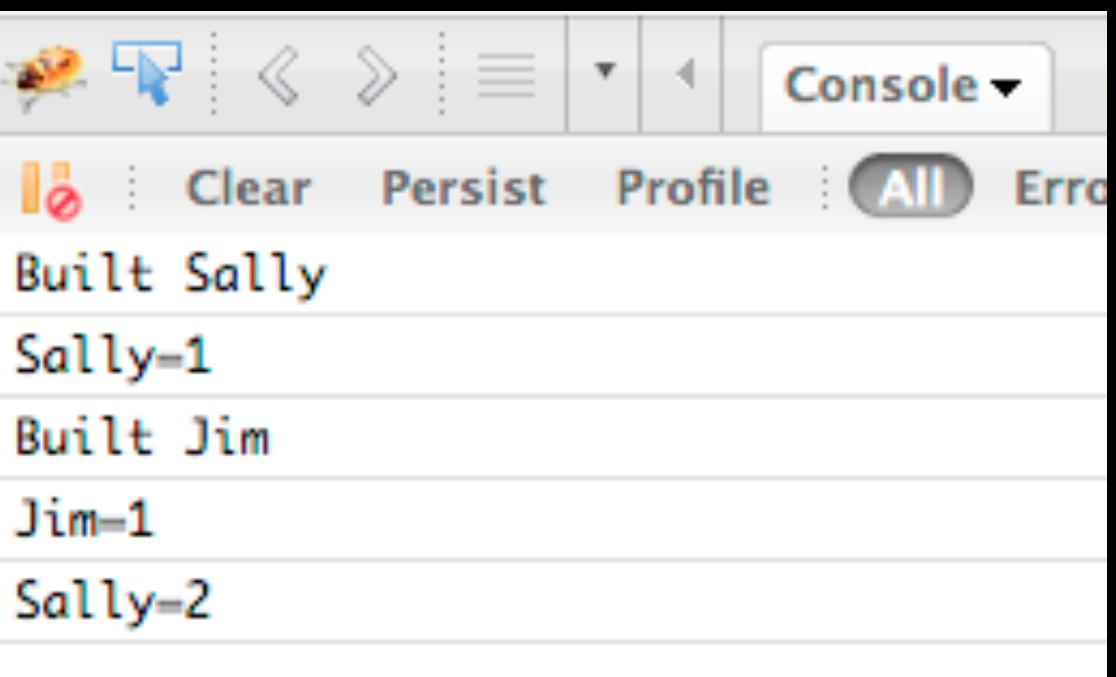
Many Instances

- We can create **lots of objects** - the class is the template for the object.
- We can store each **distinct object** in its own variable.
- We call this having multiple **instances** of the same class.
- Each **instance** has its own copy of the **instance variables**.

```
function PartyAnimal(nam) {  
    this.x = 0;  
    this.name = nam;  
    console.log("Built "+nam);  
    this.party = function () {  
        this.x = this.x + 1;  
        console.log(nam+"="+this.x);  
    }  
  
}  
  
s = new PartyAnimal("Sally");  
s.party();  
  
j = new PartyAnimal("Jim");  
j.party();  
s.party();
```

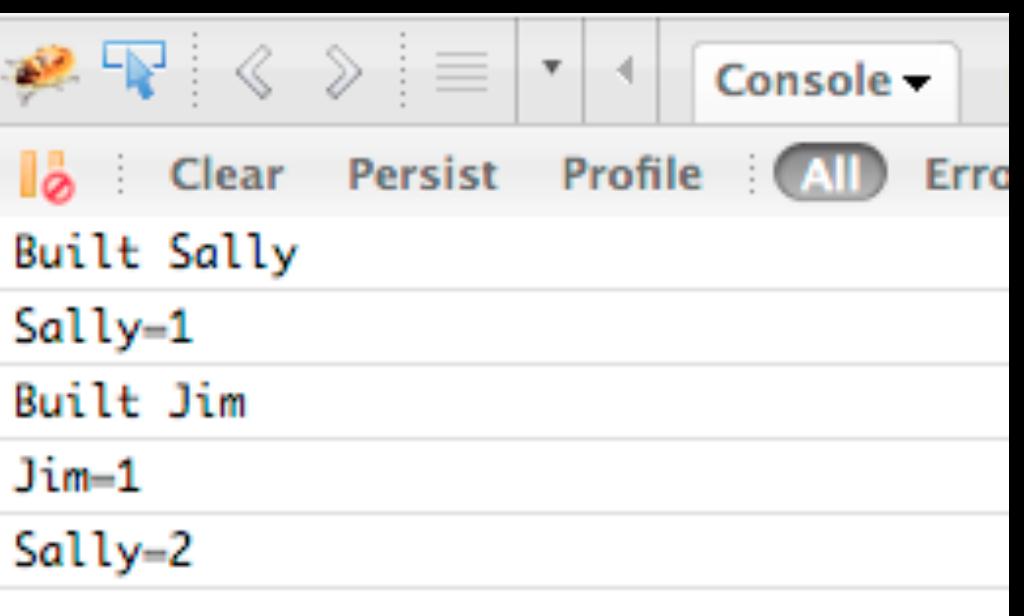
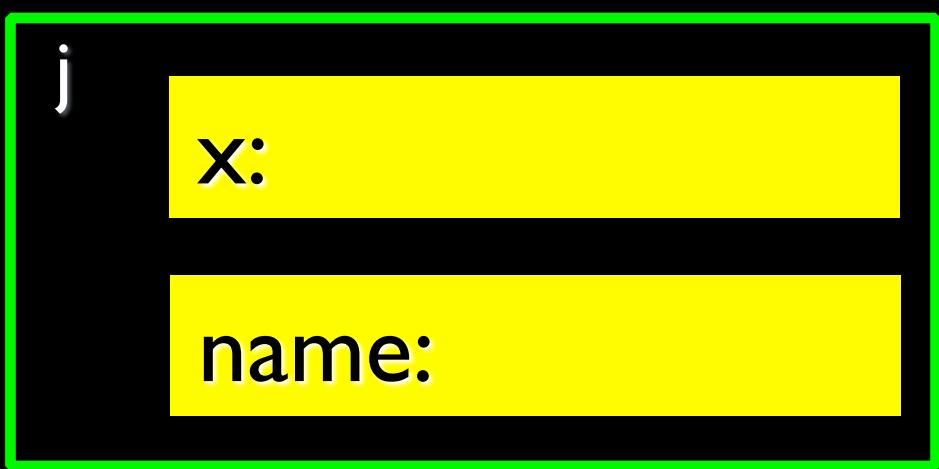
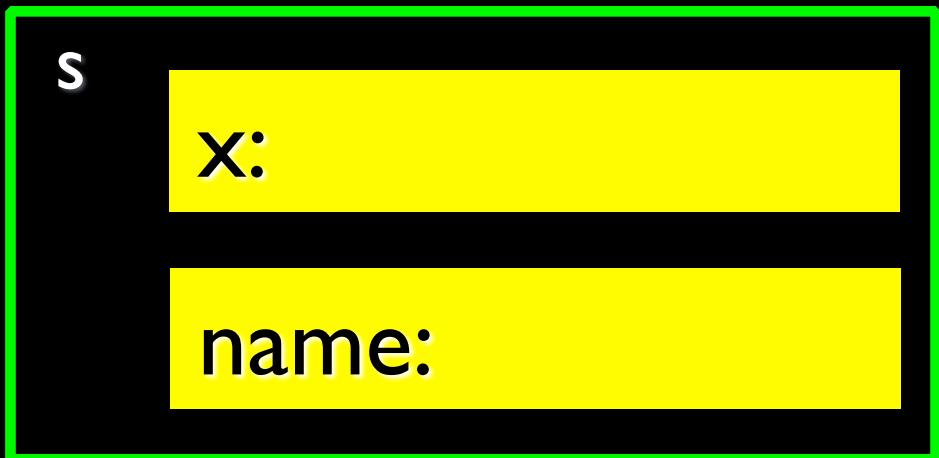
js04.htm

Constructors can have additional **parameters**. These can be used to set up **instance variables** for the particular instance of the class (i.e., for the particular object).



```
function PartyAnimal(nam) {  
    this.x = 0;  
    this.name = nam;  
    console.log("Built "+nam);  
    this.party = function () {  
        this.x = this.x + 1;  
        console.log(nam+"="+this.x);  
    }  
  
s = new PartyAnimal("Sally");  
s.party();  
  
j = new PartyAnimal("Jim");  
j.party();  
s.party();
```

js04.htm



Definitions



- **Class** - a template – Dog
- **Method or Message** - A defined capability of a class - bark()
- **Constructor** - A method which is called when the instance / object is created
- **Object or Instance** - A particular instance of a class - Lassie

Summary

- The key for this lecture is to understand how to read OO documentation for JavaScript and how to use objects.
- Building brand new complex objects is more advanced.
- It is important to remember that JavaScript uses objects as its “Associative Array”.

Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) as part of www.wa4e.com and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here

Additional Source Information

- “Snowman Cookie Cutter” by Didriks is licensed under CC BY
<https://www.flickr.com/photos/dinnerseries/23570475099>
- Photo from the television program *Lassie*. Lassie watches as Jeff (Tommy Rettig) works on his bike is Public Domain
https://en.wikipedia.org/wiki/Lassie#/media/File:Lassie_and_Tommy_Rettig_1956.JPG