# JavaScript Prototypes and Inheritance
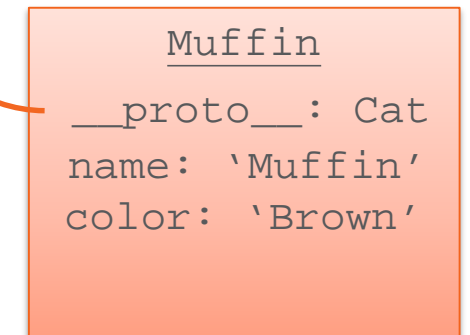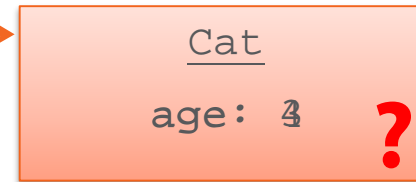
Jim Cooper

@jimthecoop

# Prototypes Behind the Scenes

```
function Cat(name, color) {
    this.name = name
    this.color = color
}

Cat.prototype.age = 3

var fluffy = new Cat('Fluffy', 'White')
var muffin = new Cat('Muffin', 'Brown')

Console.log(fluffy.age) // 3

Console.log(fluffy.age) // 5

Cat.prototype.age = 4
fluffy.__proto__.age =

Console.log(fluffy.age)
Console.log(muffin.age)
```

**Cat**

age: 3  ?

**Fluffy**

__proto__: Cat

name: 'Fluffy'
color: 'White'
age: 5  ?

**Muffin**

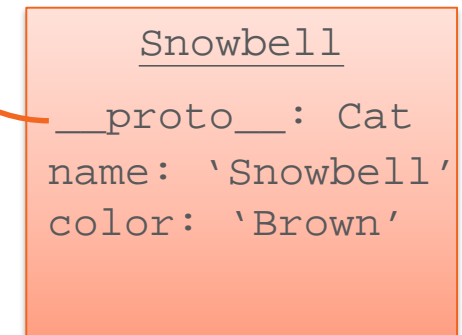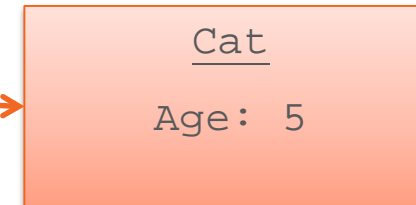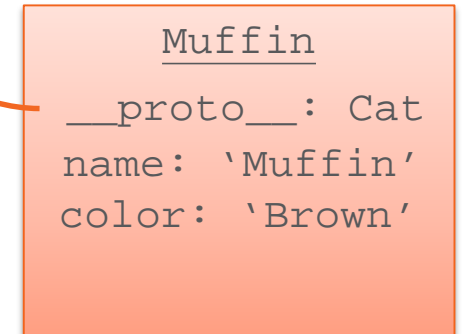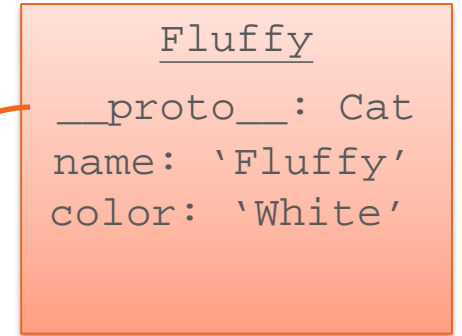__proto__: Cat
name: 'Muffin'
color: 'Brown'

**A function's prototype:**  A function's prototype is the object **instance** that will become the prototype for all objects created using this function as a constructor.

**An object's prototype:**  An object's prototype is the object **instance** from which the object is inherited.

# Changing Prototypes

```
function Cat(name, color) {
    this.name = name
    this.color = color
}
Cat.prototype.age = 4

var fluffy = new Cat('Fluffy', 'White')
var muffin = new Cat('Muffin', 'Brown')

Cat.prototype = { age: 5 }

var snowbell = new Cat('Snowbell', 'White')
```

**Cat**

Age: 4

**Cat**

Age: 5

**Fluffy**

__proto__: Cat
name: 'Fluffy'
color: 'White'

**Muffin**

__proto__: Cat
name: 'Muffin'
color: 'Brown'

**Snowbell**

__proto__: Cat
name: 'Snowbell'
color: 'Brown'

# Summary

| | |
|---|---|
| Prototypal Inheritance | Instance/Prototype Properties |
| Multiple Inheritance | Prototypal Inheritance Structures |