CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Now, we'll build a function that creates Book objects and adds them to our Box

myBox

```
height: 6   length: 10   width: 12   volume:  720   weight:  24
material: "cardboard"    destination1: "Orlando"    destination2: "Miami"
"# of stops": 2          "# of Books": 0
```

We'll create a property that tracks the number of books, set initially to zero. Our function will use this value to dynamically assign property names.

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Now, we'll build a function that creates Book objects and adds them to our Box

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 0
```

Each time we create and add a Book object, we'll increase the number of books in our Box.

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

We'll concatenate the current book # with "book" to get our property name.

Our Book's properties will come from the function parameters we've passed in.

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Now, we'll build a function that creates Book objects and adds them to our Box

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 0
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 0
```

```
addBook(myBox, "Great Expectations", "Charles Dickens");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 1
```

New book, new number of books.

```
addBook(myBox, "Great Expectations", "Charles Dickens");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 1
book1: { title: "Great Expectations", author: "Charles Dickens"}
```

The correct book number in the property name has been dynamically assigned.

```
addBook(myBox, "Great Expectations", "Charles Dickens");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 1
book1: { title: "Great Expectations", author: "Charles Dickens"}
```

```
addBook(myBox, "Great Expectations", "Charles Dickens");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 1
book1: { title: "Great Expectations", author: "Charles Dickens"}
```

```
addBook(myBox, "The Remains of the Day", "Kazuo Ishiguro");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 2
 book1: { title: "Great Expectations", author: "Charles Dickens"}
```

```
addBook(myBox, "The Remains of the Day", "Kazuo Ishiguro");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 2
   book1: { title: "Great Expectations", author: "Charles Dickens"}
   book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
```

```
addBook(myBox, "The Remains of the Day", "Kazuo Ishiguro");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 2
   book1: { title: "Great Expectations", author: "Charles Dickens"}
   book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
```

```
addBook(myBox,"Peter Pan", "J. M. Barrie");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 3
   book1: { title: "Great Expectations", author: "Charles Dickens"}
   book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
```

```
addBook(myBox,"Peter Pan", "J. M. Barrie");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 3
  book1: { title: "Great Expectations", author: "Charles Dickens"}
  book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
  book3: { title: "Peter Pan", author: "J. M. Barrie"}
```

```
addBook(myBox,"Peter Pan", "J. M. Barrie");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 3
  book1: { title: "Great Expectations", author: "Charles Dickens"}
  book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
  book3: { title: "Peter Pan", author: "J. M. Barrie"}
```

```
addBook(myBox,"On the Road", "Jack Kerouac");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 4
  book1: { title: "Great Expectations", author: "Charles Dickens"}
  book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
  book3: { title: "Peter Pan", author: "J. M. Barrie"}
```

```
addBook(myBox,"On the Road", "Jack Kerouac");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

CHANGING OUR CONTENTS TO INDIVIDUAL OBJECTS

Let's add some books!

myBox

```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 4
   book1: { title: "Great Expectations", author: "Charles Dickens"}
   book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
   book3: { title: "Peter Pan", author: "J. M. Barrie"}
   book4: { title: "On the Road", author: "Jack Kerouac"}
```

```
addBook(myBox,"On the Road", "Jack Kerouac");
```

```
function addBook (box, name, writer){
    box["# of Books"]++;
    box["book" + box["# of Books"]] = {title: name, author: writer};
}
```

REFERENCING OBJECTS WITHIN OBJECTS

Use the dot extension or subsequent bracket notation to get to deeper properties

myBox

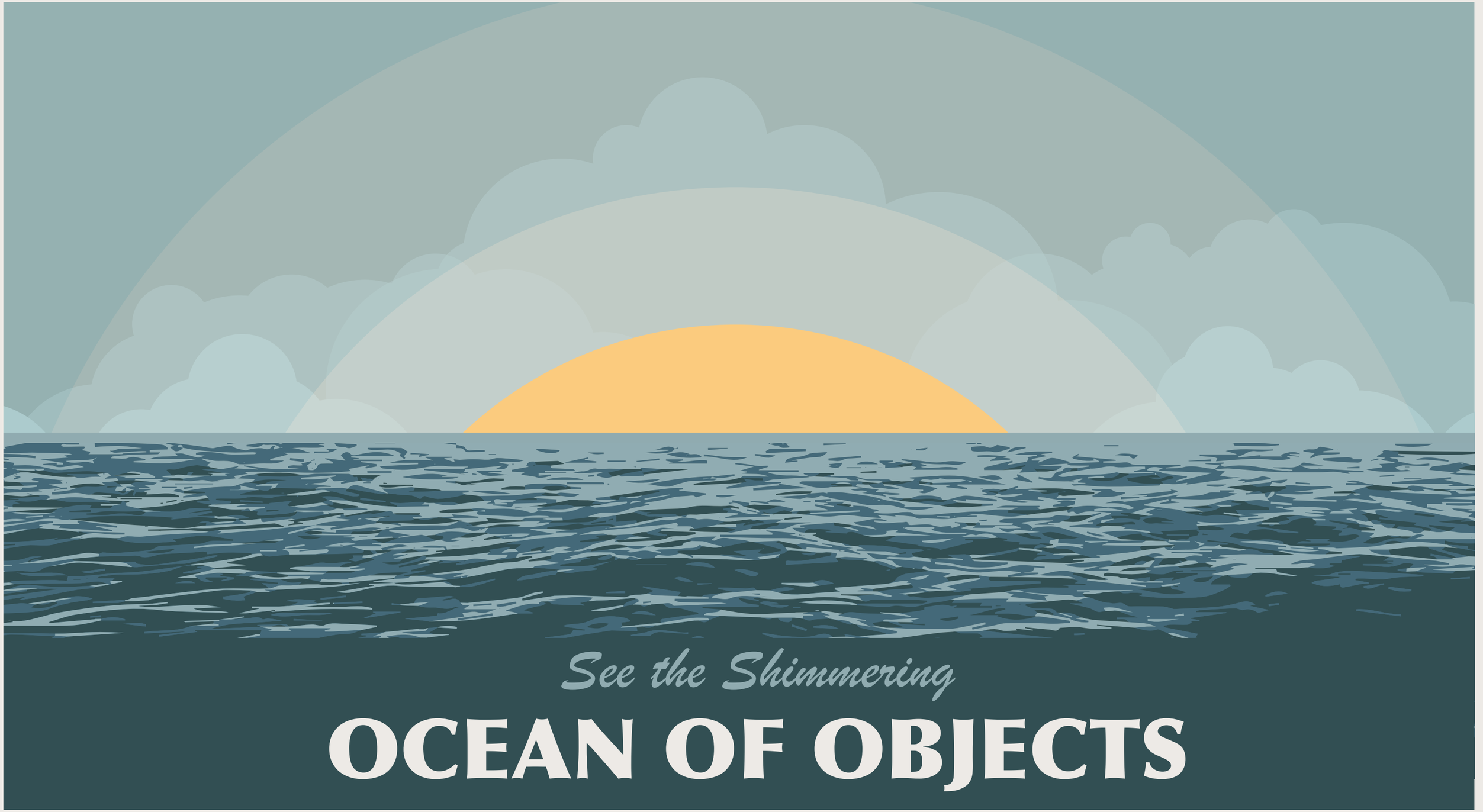
```
height: 6   length: 10   width: 12   volume: 720   weight: 24
material: "cardboard"   destination1: "Orlando"   destination2: "Miami"
"# of stops": 2          "# of Books": 4
 book1: { title: "Great Expectations", author: "Charles Dickens"}
 book2: { title: "The Remains of the Day", author: "Kazuo Ishiguro"}
 book3: { title: "Peter Pan", author: "J. M. Barrie"}
 book4: { title: "On the Road", author: "Jack Kerouac"}
```

```
console.log( myBox.book1.title );
```

→ Great Expectations

```
console.log( myBox["book4"]["author"] );
```

→ Jack Kerouac


See the Shimmering
OCEAN OF OBJECTS

AN AQUARIUM OBJECT FILLED WITH OTHER OBJECTS

Let's first build some add/remove functionality for creatures and environment toys

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false }
};
```

```
function addCritter( container, name, type, species, length ){
    container[name] = {type: type, species: species, length: length};
}
```

```
function addToy( container, name, type, material, moves ){
    container[name] = {type: type, material: material, moves: moves};
}
```

Wouldn't it be nice if these functions belonged only to the aquarium instead of an entire program? Let's try adding one.

AN AQUARIUM OBJECT FILLED WITH OTHER OBJECTS

Let's first build some add/remove functionality for creatures and environment toys

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false }
};
```

```
function addCritter( container, name, type, species, length ){
    container[name] = {type: type, species: species, length: length};
}
```

PROPERTIES CAN ALSO BE FUNCTIONS

An Object's function properties are often called its "methods"

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){

      }

};
```

We add a new property to our aquarium that takes the name of our addCritter function. Then we build an anonymous function.

```
function addCritter( container, name, type, species, length ){
    container[name] = {type: type, species: species, length: length};
}
```

PROPERTIES CAN ALSO BE FUNCTIONS

An Object's function properties are often called its "methods"

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){

      }

};
```

Our container parameter now disappears, since we are making the function BELONG TO that very container.

```
function addCritter( container, name, type, species, length ){
    container[name] = {type: type, species: species, length: length};
}
```

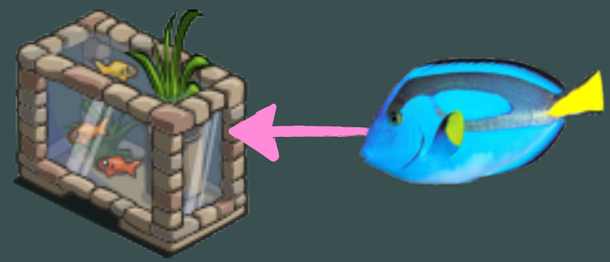
THE VERY USEFUL "THIS" KEYWORD

"This" always refers to the owner Object of the function in which the "this" is used.

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      }
};
```

When called with this, addCritter says: Hey, aquarium! Make a new property called name and assign to it a new Object with these properties!

```
function addCritter( container, name, type, species, length ){
    container[name] = {type: type, species: species, length: length};
}
```



WOOHOO, A PROPERTY THAT HOLDS A FUNCTION!

Our addCritter function is now available as a property on the aquarium Object

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      }

};
```

Let's add a creature!

WOOHOO, A PROPERTY THAT HOLDS A FUNCTION!

Our addCritter function is now available as a property on the aquarium Object

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      }
};
```

```
aquarium.addCritter("Bubbles", "fish", "yellow tang", 5.6);
```

We call the function just like referencing any other property in aquarium, but we also pass it a set of appropriate parameters.

WOOHOO, A PROPERTY THAT HOLDS A FUNCTION!

Our addCritter function is now available as a property on the aquarium Object

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.addCritter("Bubbles", "fish", "yellow tang", 5.6);
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.addCritter("Bubbles", "fish", "yellow tang", 5.6);
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {


};
```

All we will need to delete any property, whether creature or toy, is its name.

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];


};
```

A temp variable will help us hold on to the Object that we remove. This way we'll still have access to it outside the aquarium.

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
         this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];

};
```

Next we remove the property from the Owner object, in this case, the aquarium .

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];
    return temp;
};
```

Finally we return the temp variable, so that we can still have a reference to the removed Object.

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Marlin: { type: "fish", species: "clownfish", length: 4.1 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];
    return temp;
};
```

```
var fishOutOfWater = aquarium.takeOut("Marlin");
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },

      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];
    return temp;
};
```

```
var fishOutOfWater = aquarium.takeOut("Marlin");
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },

      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];
    return temp;
};
```

```
var fishOutOfWater = aquarium.takeOut("Marlin");
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

Uh oh! Notice that we lost Marlin's name! Let's fix that problem with some property trickery.

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];
    return temp;
};
```

```
var fishOutOfWater = aquarium.takeOut("Marlin");
console.log( fishOutOfWater );
```

→ `Object {type: "fish", species: "clownfish", length: 4.1}`

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    var temp = this[name];
    delete this[name];
    return temp;
};
```

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    this[name].name = name;
    var temp = this[name];
    delete this[name];
    return temp;
};
```

The first name in this line of code finds the desired Object in the aquarium using the parameter as a property name.

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    this[name].name = name;
    var temp = this[name];
    delete this[name];
    return temp;
};
```

Coming after a dot, the second name creates a new property IN the Object we want to remove! Notice that this is NOT the same as the function's parameter!

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

```
aquarium.takeOut = function ( name ) {
    this[name].name = name;
    var temp = this[name];
    delete this[name];
    return temp;
};
```

The third name assigns the old property name to the newly created name property in the removed Object. Sneaky!

HMM...OUR TANK'S A LITTLE FULL

Let's build another method that removes any object from our aquarium

```
var aquarium = {
      Nemo: { type: "fish", species: "clownfish", length: 3.7 },
      Dory: { type: "fish", species: "blue tang", length: 6.2 },
      Peach: { type: "echinoderm", species: "starfish", length: 5.3 },
      "Coral Castle": { type: "environment", material: "coquina", moves: false },
      "Dragon Statue": { type: "environment", material: "plastic", moves: false },
      addCritter: function ( name, type, species, length ){
          this[name] = {type: type, species: species, length: length};
      },
      Bubbles: { type: "fish", species: "yellow tang", length: 5.6 }
};
```

Woohoo, no identity crisis for Marlin!

```
aquarium.takeOut = function ( name ) {
    this[name].name = name;
    var temp = this[name];
    delete this[name];
    return temp;
};
```

```
var fishOutOfWater = aquarium.takeOut("Marlin");
console.log( fishOutOfWater );
```

→ `Object {type: "fish", species: "clownfish", length: 4.1, name: "Marlin"}`