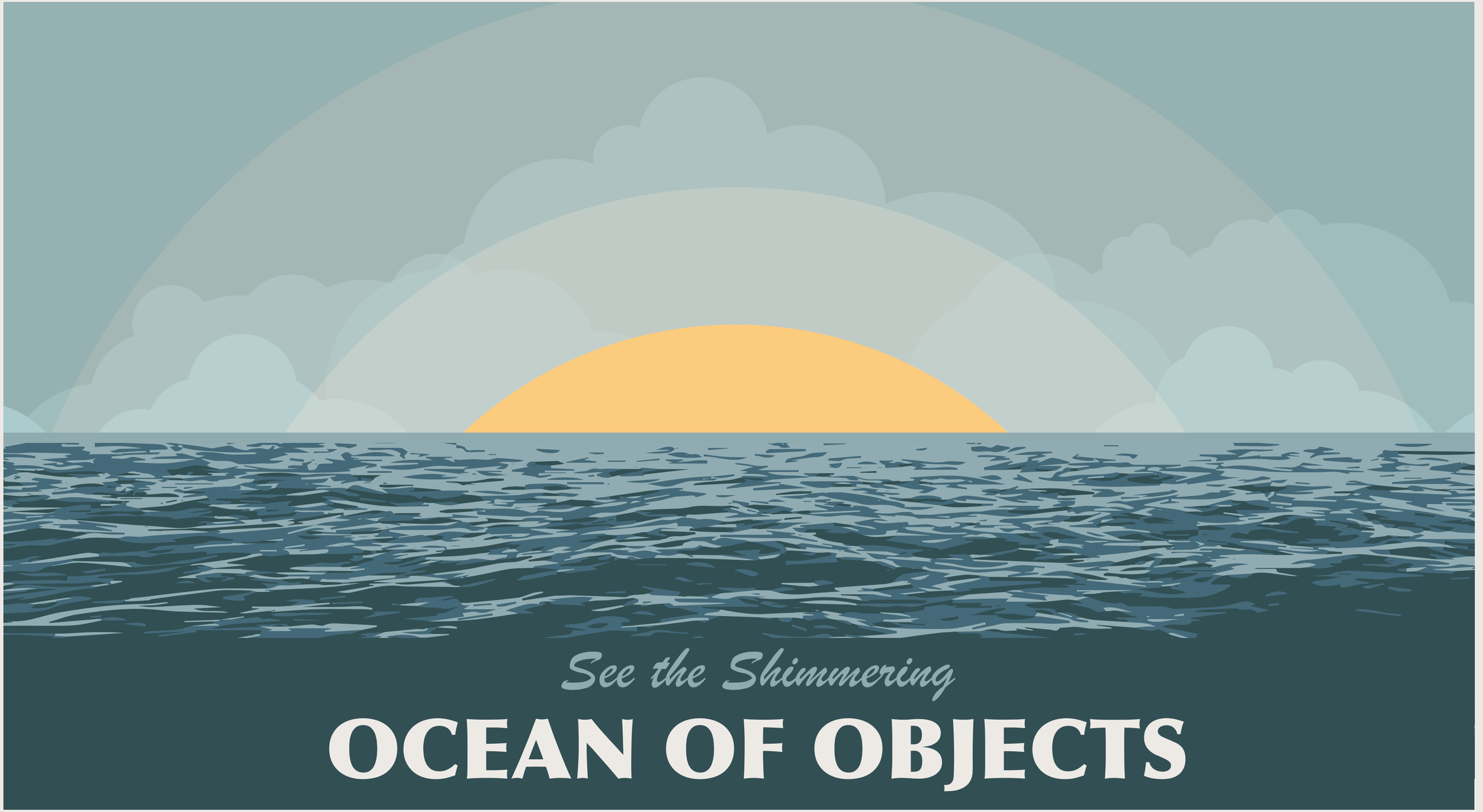

See the Shimmering
OCEAN OF OBJECTS

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};
}
```

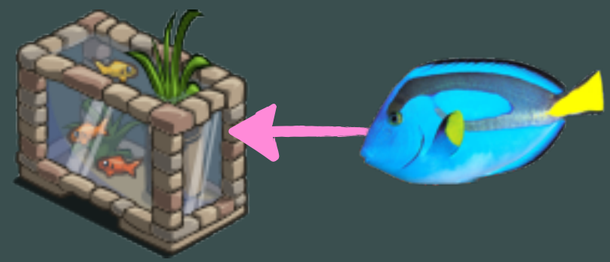
THE VERY USEFUL "THIS" KEYWORD

"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"}`

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;
};
```

```
var toy = aquarium.takeOut("Dragon Statue");
```

Our removal method works for toy Objects too!

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 },


      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;
};
```

```
var toy = aquarium.takeOut("Dragon Statue");
```

Our removal method works for toy Objects too!

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 },


      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;
};
```

```
var toy = aquarium.takeOut("Dragon Statue");
```

Our removal method works for toy Objects too!

HMM...OUR TANK'S A LITTLE FULL

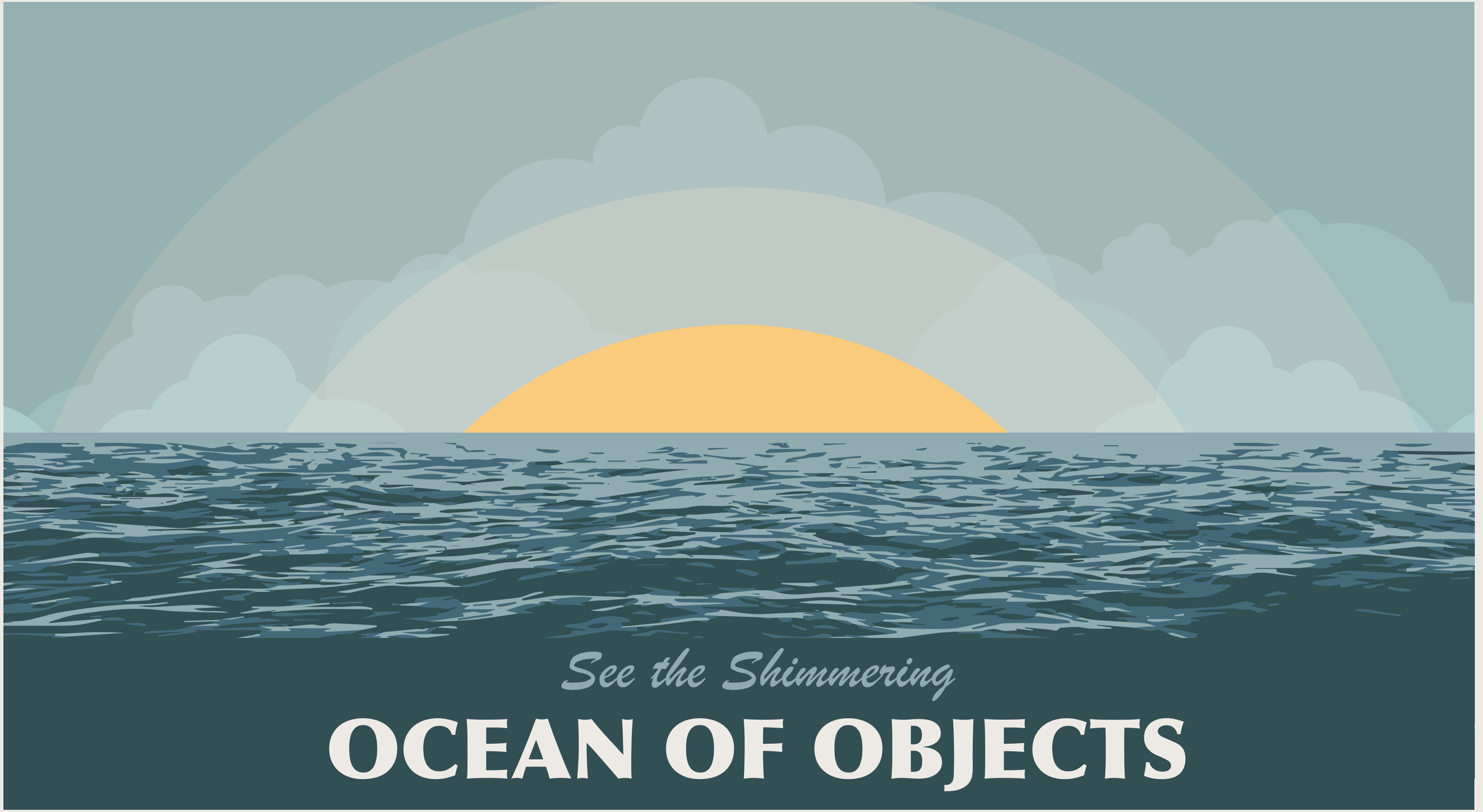
Let's build another method that removes any object from our aquarium

```
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 },
      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;
};
```

```
var toy = aquarium.takeOut("Dragon Statue");
console.log( toy );
```

→ `Object {type: "environment", material: "coquina", moves: false, name: "Dragon Statue"}`


See the Shimmering
OCEAN OF OBJECTS